SmartOrBAC

Enforcing security in the Internet of Things

Imane BOUIJ-PASQUIER
ENSA UCA
Marrakech, MOROCCO

Anas ABOU EL KALAM
ENSA UCA
Marrakech, MOROCCO

Abdellah AIT OUAHMAN
ENSA UCA
Marrakech, MOROCCO

***Abstract*—The emergence of the Internet of Things (IoT) paradigm, provides a huge scope for more streamlined living through an increase of smart services but this coincides with an increase in security and privacy concerns, therefore access control has been an important factor in the development of IoT.**

This work proposes an authorization access model called SmartOrBAC built around a set of security and performance requirements. This model enhances the existing OrBAC (Organization-based Access Control) model and adapts it to IoT environments. SmartOrBAC separates the problem into different functional layers and then distributes processing costs between constrained devices and less constrained ones and at the same time addresses the collaborative aspect with a specific solution. This paper also presents the application of SmartOrBAC on a real example of IoT and gives a complexity study demonstrating that even though this model is extensive, it does not add additional complexity regarding traditional access control models.

Keywords—internet of things; security; privacy; access control model; authorization process

I. INTRODUCTION

Today we are seeing a change in our perception of Internet towards a global network of "smart objects", which we can call the Internet of Things (IoT). These advances are estimated to accelerate over the next few years [1, 2] in response to reduced hardware costs, internet's technological maturity and the swift development of communication technology. This will lead to a smooth assimilation of these smart objects into the Internet, which will in turn enable mobile and widespread access. Areas that are expected to be directly affected include healthcare [3, 4], supply chain management [5], transport systems [6], agriculture and environmental monitoring [7, 8], life at home and more, as we move towards "smart homes" [9, 10, 11] and the next generation of "smarter cities"[12].

This extension and proliferation of technology will certainly change our life, but will also present security and privacy challenges [13, 14, 15], since unexpected information leaks and illegitimate access to data and physical systems could have a high impact on our lives. Moreover, malicious modifications or denial of service may also cause damage in the context of IoT. Subsequently, the implementation of an access control mechanism that respects both the character of and the constraints on, smart objects in the IoT environment, is imperative. In this paper addresses one of the most relevant security issues - authorization and access control - in the context of distributed, cross-domain systems that consist of resource constrained devices not directly operated by humans. Especially, the problem where a single constrained device is communicating with several other devices from different organizations or domains. Based on OrBAC [16] access control model, our "Smart OrBAC" proposal is specifically designed for IoT environments. It, in fact, takes the main features of IoT into account and facilitates a distributed-centralized approach where authorization decisions are based on local conditions, and in this way offers context-aware access control.

The main contributions of this work can be outlined as follows:

- Exhaustive study and deep analysis of IoT security requirements and needs regarding its specific characteristics.
- Abstraction layers design regarding the specificities of IoT devices.
- SmartOrBAC, our access Control Model for IoT.
- Applying SmartOrBAC to an IoT case study and showing that it does not present additional complexity.

The rest of the paper is organized as follows: Section 2 presents a healthcare case study that allows us to extrapolate the relevant security requirements that an access control mechanism must fulfill, and then, generalizes these requirements for wider use in Section 3. Afterward, Section 4 gives an overview of the literature and discusses the important access control models currently existing in the IoT environment. Then, Section 5 describes the background needed to understand our new work. The SmartOrBAC access control model is then detailed in Section 6. Section 7 presents a complexity study comparing SmartOrBAC with traditional models followed by a brief description of the implementation in Section 8. Finally, Section 9 presents our conclusions.

II. CASE STUDY

Before going into technical details, let us first discuss a representative scenario [17, 18]. A number of security requirements will be derived from this scenario.

Assume that John, a man with a heart condition, has opted for an assisted living service that is provided by a medical center. John uses a device that monitors his heart rate and his position; his home is also equipped with multiple sensors and actuators (temperature sensor, humidity sensor, luminosity sensor...). In the case of a cardiac problem, the heart monitor alerts the emergency services, and informs of John's current location. Moreover, the device uses smart logic to identify its owner "John" and allows him to configure the device's settings, including access control. This mainly prevents situations where

someone else wearing that device acts as the owner and changes the access control and security settings.

In addition, John can add additional people to be notified in case of emergency, such as members of his family. Furthermore, the device saves the collected data, in order to assist his physician with his analysis.

However, John is worried that one of these authorized people may use the device to monitor his location even in the absence of an emergency. Furthermore, he is reluctant to let his health insurance company have access to this stored data, due to the possibility that they may decide he is too big a risk and therefore refuse to insure him.

A doctor, who monitors John's health remotely from the medical center, receives an alarm that John has fainted. An ambulance is instructed to go to assist John. A smart driving application is used by the ambulance to reach John's home as quickly as possible.

The situation requires the interaction of the following key actors:

- **Smart home of John**, actuators and sensors located in the house are used to collect vital information about the patient and sent to the monitoring service in the medical center, which oversees the patient's condition.
- **The medical center,** for monitoring John's health and the environmental conditions in the smart home. It then initiates appropriate action, such as alerting emergency services and sending the smart ambulance.
- The **ambulance** requests information from Traffic Monitoring in the police department in order to find the best route to John's home and save valuable time.
- **The police department for traffic jams monitoring, which** receives data from the distributed platforms sensors in order to infer the traffic status in the city's streets.
- **The smart city**: which includes all the previous stakeholders as a sub-stakeholders and where are various types of sensors, which are connected through Wireless Sensor Network (WSN) platform using various access technologies and/or communication protocols (ZigBee, Bluetooth, Wi-Fi, etc.) sharing their data.

In fact, each actor can be considered as (or belongs to) an *organization* or a domain e.g. "the medical center".

Subsequently, each organization is structured by different *roles* e.g. "doctor in the monitoring service", several *activities*, e.g. "consult", several *views* (groups of objects), e.g." patient's medical history, received sensor's data from monitored patient" and finally, the *context*, e.g. "A medical emergency such as John's faint".

The scenario intends to demonstrate:

- The cross-application nature of smart objects in one IoT service by showing their ability to simultaneously connect multiple application sectors and, more specifically, smart health, smart home, smart living, smart transport, etc…
- John needs to have the option of configuring his preferences related to trusted people or groups who can access his data in case of emergency (e.g. heart rate, location…).
- He must be able to block access to specific persons or groups, if he mistrusts them.
- The security measures must not affect the device's battery lifetime significantly. More precisely, since physically accessing the implanted device is hard or even impossible, the security measures should not affect battery lifetime significantly and not require direct physical interaction.
- Easy and intuitive configuration of the device.

III. General Requirements of the IoT

This section presents the most important IoT requirements derived from the case study, and then, generalized for wider use.

- *Interoperability*: The access control model must be designed for multiple organizations. On the one hand, each organization set up its own policies. On the other hand, it must respect other collaborating organization's policies.
- *Context awareness*: In IoT environments, context is highly important [19, 20]. In fact, services and applications use knowledge from the context surrounding them in order to gain information about their users and the users' environment [21, 22, 23, 24]. Thus authorization decisions are inextricably linked to local contextual data available to the device.
- *Ergonomie*: Due to the high saturation level of smartobjects in everyday life, many non-expert users are pushed to define permissions on their devices. Therefore, an access control mechanism must be simple to use: easily administrated, expressed and modified. In addition, it must enable policy updates without re-provisioning individual devices, and it must be designed so as not to require manual intervention of the user in the access control process.
- *Heterogeneity*: A collaborative environment may combine several technologies [25, 26, 27, 28]. This heterogeneity results in interoperation challenges, such as devices from different producers that provide proprietary features used by several services implemented according to diverse standards and protocols in order to initiate multiple functions [29].
- *Fine grained Access control*: The access control mechanism must be able to apply different permissions for different requesting entities rather than being all-or-nothing. Consequently, there is a need for granularity in authorization decisions.

- *Lightweight solution*: Due to the constrained energy nature of the IoT component, access control may minimize resource usage on the constrained device.
- *Scalability*: is the way to scale while managing increasingly large volumes of users, applications and connected devices. An Access control mechanism should naturally be extensible in size, structure, and number of organizations [30].

IV. Related Work

Zhang and Gong proposed in [31] the UCON model taking into consideration flexibility and heterogeneity in an IoT distributed environment. However, UCON is a conceptual model only, and thus it does not give details on the implementation of the monitoring process. This approach is indeed still not practical.

The CAPBAC model is implemented in a centralized approach in [32] where the proposed framework is based on a central Policy Decision Point (PDP) which handles authorization decisions. Whereas the implementation of capability-based access control in IoT is considered in [33] with an entirely distributed approach without intervention of central entities. The limits of both a purely centralized approach and fully distributed approach will be detailed bellow later on in this paper (see V.B Main architectures for access control in the Internet of Things).

The Capability-based Context-Aware Access Control (CCAAC) [34] is a delegation model based on a federated vision of IoT [35], where a central entity in each domain is in charge of authorizing a delegation request from a delegator, and making the decision about granting it to the delegate. However, this vision does not make use of technologies specifically designed for constrained highly context dependent environments such as IoT. Furthermore, the technical requirements in the constrained environment of the different actors involved in the proposed delegation mechanism are missing from this study.

Seitz et al. present in [36] an authorization framework based on XACML [37]. Evaluating XACML policies is too heavy-weight for constrained devices; therefore most of the authorization process is externalized. In order to convey the authorization decision from the external point to the device, an assertion is encoded in JSON [38] and is sent to the end-device (i.e., sensor or constrained device). The end-device takes responsibility for local conditions verification. However, this study does not give information about the central component involved neither about its management within the organization. Also, this proposal is bound to the use of XACML, which is not specifically designed for use in constrained devices.

V. Towards Centralized-Distributed Access Control for the Internet of Things

The integration of resource constrained devices into the Internet requires specifically designed technology and protocol that respect the nature of these smart objects. Recently, several IETF Working Groups have been focused on the adaptation of existing Internet protocols to IoT scenarios. These rising protocols, such as CoAP [39] and 6LoWPAN [40, 41, 42] aim to enable a seamless integration of the constrained devices into the Internet. It is then necessary to develop security mechanisms to fully take advantage of the huge potential offered by these protocols and technologies.

Prior to the detailed presentation of SmartOrBAC, this section describes briefly some of the core concepts that make up the proposed scheme. First of all, an overview of the OrBAC access control model and its benefits over other commonly accepted models are given. Then an overview of the main approaches and trends to provide access control logic in IoT scenarios is presented based on the architecture taxonomy proposed in [43].

A. Organization-Based Access control model (OrBAC)

The OrBAC model introduces the concept of organization as a structured group of active entities, in which subjects play specific roles. An activity is a group of one or more actions, a view is a group of one or more objects, and a context is a specific situation.

Actually, the Role entity is used to structure the link between the subjects and the organizations .The Empower (*org*, *r*, *s*) relationship (or predicate) means that *org* employs subject *s* in role *r*. In the same way, the objects that satisfy a common property are specified through views, and activities are used to abstract actions.

In security rules, permissions are expressed as Permission (*org*, *r*, *v*, *a*, *c*), obligations and prohibitions are defined similarly. Such an expression is interpreted as: in the context *c*, organization *org* grants role *r* the permission to perform activity *a* on view *v*.

As rules are expressed only through abstract entities, OrBAC is able to specify the security policies of several collaborating and heterogeneous organizations.

In our context, OrBAC presents several benefits:

- *Rules expressiveness*: OrBAC defines permissions, interdictions and obligations.
- *Abstraction of the security policy*: OrBAC has a structured and an abstracted expression of the policy; it also separates the specification from the implementation of the policy.
- *Scalability*: OrBAC has no limitation in size or capacity. It can define an extensible policy. It is then easily applicable to large-scale environments such as IoT.
- *Loose coupling*: each organization is responsible for its assets and entities. Implementation details as well as private information are managed separately by each organization.
- *Evolvability*: a policy in OrBAC is evolvable. It easily handles changes in organizations.
- *User-friendliness*: specifying and updating an OrBAC security policy are rather intuitive.
- *Popularity*: OrBAC has a growing community. Many research studies are being conducted, based on OrBAC.

- *Context-aware*: OrBAC takes the context (e.g. specific situations, time and location constraints) into account.
- *Fine-grained access control*: thanks to the context ad to its abstract and concrete concepts, OrBAC enables security administrators to define, set, specify, implement dynamic security policies and control access to individual data items and attributes.

However, despite the several advantages of OrBAC, it is not completely adapted to IoT. In particular, OrBAC is not able to manage collaboration-related aspects. In fact, as OrBAC security rules have the Permission (*org*, *r*, *v*, *a*, *c*) form, it is not possible to represent rules that involve several independent organizations (e.g. when the ambulance's driver, in *the medical center* organization, requests information from *Traffic Monitoring in the police department* organization in order to find the best route to John's home and save valuable time), or even, autonomous sub-organizations of a particular collaborative system (e.g. when the *police department for traffic jams monitoring* which is a sub-organization of smart city accede to data from the distributed sensors nodes in *the smart city* organization, in order to infer the traffic status in the city's streets). Moreover, it is impossible (for the same reason) to associate permissions to entities belonging to other partner-organizations *(*or to sub-organizations). As a result, if we can assume that OrBAC provides a framework for expressing the security policies of several organizations, it is unfortunately only adapted to centralized structures and does not cover the distribution, collaboration and interoperability needs, and these aspects are very important in the IoT context.

In order to overcome the limitations listed above, on one hand, the OrBAC model will be extended to include collaboration-related and context aware concepts; and on the other hand, a new architecture articulated around four functional layers will be proposed. The resulting framework is called "SmartOrBAC".

B. Main architectures for access control in the Internet of Things

This section gives an overview of the most popular current architecture providing access control in IoT services highlighting their main advantages and drawbacks.

1) Centralized architecture

In order to relieve smart objects from processing a large amount of access control related tasks, these functionalities are externalized to a back-end server or gateway responsible for authorization processing and thus, the end component (e.g sensors and actuators…) have a limited part (see

).

The most pertinent advantage of the centralized approach is that the access control logic is located within a non-constrained entity. It follows that the use of standard security protocols normally used in the traditional Web is not restricted. XACML may for example, be used to express access control policies.

Nonetheless, this approach encounters a major problem. In IoT scenarios such as the healthcare case study seen above, contextual information is of great importance, while in a centralized architecture, authorization evaluation doesn't take into account local contextual information related to the end component. Thereby, this one single vulnerability may compromise sensitive information, and this context insensitive central entity becomes the main weakness of the centralized approach.

2) Distributed approach

In this architecture, the access control process is carried out by the end component. This means that each device must be capable of handling authorization processes and having adequate resources to do so (see Fig. 1). An advantage of this approach is that end-devices act smartly, and are autonomous. A second advantage is that this approach allows real time contextual information to become central to the authorization decision. Furthermore, in this approach, end-to-end security is more easily achieved, as there is no need for an intermediate entity.

However, the need to extend the constrained device with access control logic makes the implementation of this approach unfeasible in resource-constrained devices.

3) Centralized-distributed approach

In this approach, the end-devices partially participate in the access control decisions (see Fig. 1) enabling the authorization evaluation process to take into account contextual information. As seen in the case study, there are environments where access control is not possible without the including information from the end component at the precise time of access request (e.g. location, temperature, humidity, CO2 level, heartbeat rate etc…).

This hybrid (centralized-distributed) approach, as in the centralized approach, allows us to use standard technologies to operate access control and the transmission of contextual information request will then be operated by specific application protocols as the *Constrained Application Protocol* (CoAP).

The most obvious disadvantage of this approach are the delays caused by the transmission of the contextual information from the end component to a central entity when needed. Due to this limitation, the value acquired by the end component may be different at the time of making the authorization decision, and consequently end-to-end security is unattainable.

Each one of these three approaches has advantages and drawbacks that need to be considered while considering them for the design of the access control.

In our proposal, the design of access control is based on the centralized-distributed approach. But unlike other proposals that use this approach, each separate group of components will have a central authorization engine (rather than just having one of these engines centrally performing all the authorization processes).The selection process that determines which entity will act as this engine depend on the contextual properties of the nodes in its group. The aim of this is to make the access control mechanism more time efficient by facilitating a

smoother exchange of information between the end device and the authorization engine.

This vision is made possible by the fact that in a constrained environment, not all the devices are at the same level of constraint. In almost every WSN, less constrained nodes exist, and thus the central authorization server in charge of an area can be implemented on one of them. For more understanding, the next section gives an overview of the different actors involved in the proposed architecture and their properties.

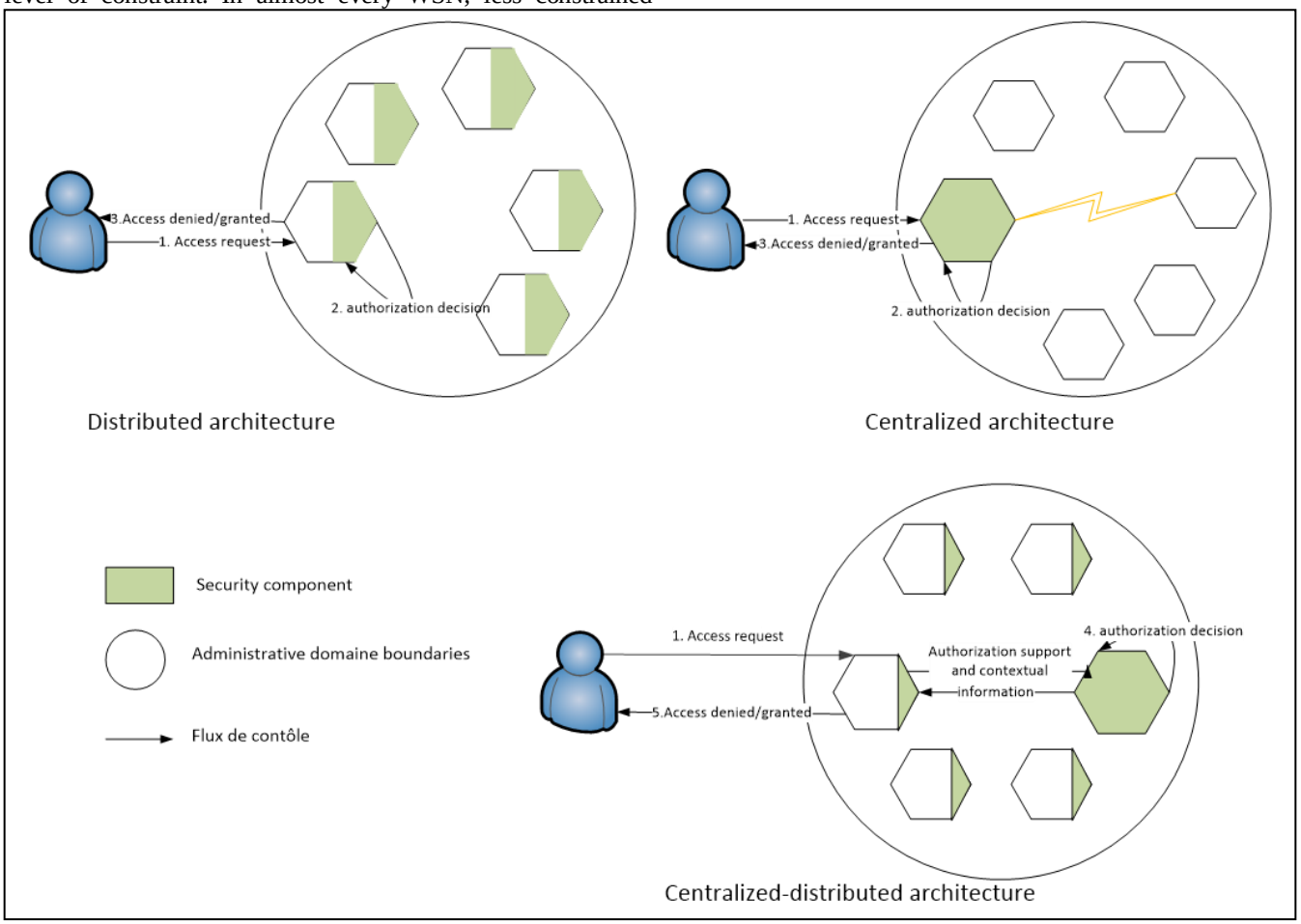

Fig. 1. Main architectures for IoT access control

C. Actors in SmartOrBAC

The main actors are the following [44]:

- **Resource Server (RS)**: An entity which hosts and represents a Resource that might contain sensor or actuator values or other information;
- **Resource Owner (RO)**: The principal that owns the resource and controls its access permissions;
- **Client (C)**: An entity which attempts to access a resource on a Resource Server;
- **Client Owner (CO)**: The principal that owns the Client and controls permissions concerning authorized representations of a Resource.

Consequently, in a basic scenario, *C* wants to access *R* located on *RS*. It follows logically that, *C* and / or *RS* are constrained.

VI. SMARTORBAC

The following paragraph contains description of the key aspects of our proposal. First, an explanation of the most relevant features of the abstraction layers design is given followed by a presentation of the collaborative solution. Then a structured expression of the *context* concept is presented. Finally the proposal is applied on the previous IoT scenario presented above.

A. SmartOrBAC abstraction layers

The SmartOrBAC architecture proposes, among others, a model based on a partitioning of the access control process into functional layers depending on the capabilities offered on each one. This approach is directly inspired by the fact that each device is constrained to a different level; they are in fact not all uniformly constrained. Note that the term “constrained node” is used according to the RFC 7228 [45]. While processing access control related tasks each layer assists the one below when

needed. Note that the authentication process details are out the scope of this study. Only authorization aspects are treated. Four layers are introduced:

1) Constrained layer

One or both of *C* and *RS* are presumed to be located in a constrained node, but despite this, must perform access control related tasks. We thus consider that either of them may be unable to manage complex tasks while processing authorization requests. In addition, nodes do not always have permanent network connectivity. That's why both of *C* and *RS* are considered to be constrained layer actors. In order to address the limitations present in this layer, a less constrained device is associated to each area of constrained devices. This centric entity is defined by the upper layer called less-constrained layer (see Fig. 2).

2) Less constrained layer

To relieve constrained layer actors from conducting computationally intensive tasks, another layer is introduced. Each group of constrained layer actors is bound to a less constrained layer actor that belongs to the same security domain (see Fig. 2).

This link is configured by the entity in charge of the device (see Section VI.A.3) Organization layer). We call this central element the "*Client Authorization Engine*" (CAE), on the client side, and Resource Authorization Engine (RAE) on the resource side.

The Client Authorization Engine (CAE) belongs to the same security domain as *C*. It assists *C* in determining if *RS* is an authorized source for *R* by obtaining authorization information and supporting *C* in handling the authorization process.

The Resource Authorization Engine (RAE) belongs to the same security domain as *R* and *RS*. It assists *RS* in determining the correct permissions of *C* on the requested resource *R*. *RAE* obtains authorization information and supports *RS* in handling the authorization process.

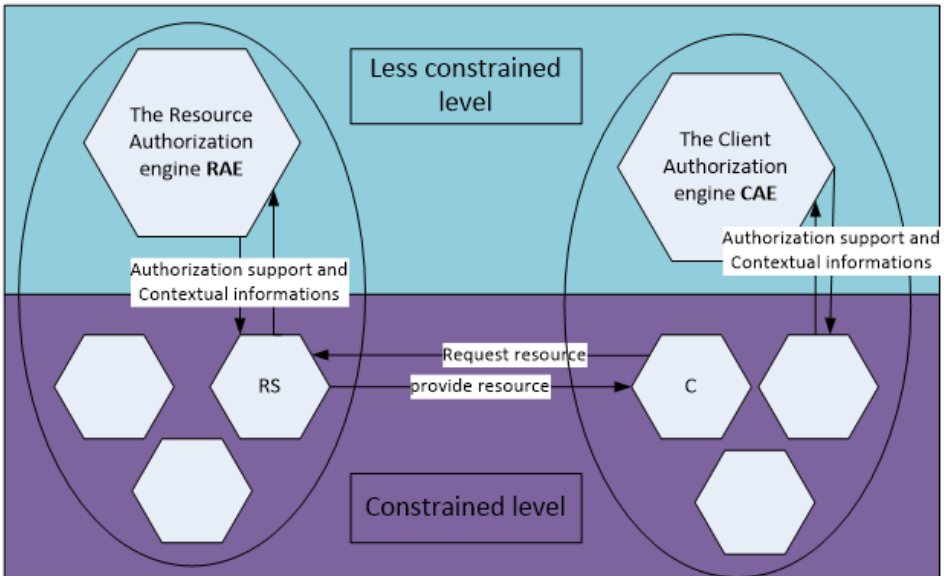

Fig. 2. Constrained and less constrained layers

3) Organization layer

In the real world, *C* and *R* are under the control of some physical entities. These entities are commonly called *ROr* (Resource Organisation) and *COr* (Client Organisation). In order to keep close to reality and to the OrBAC environment, this entity will be represented by Organisations (e.g. the police department, John's home, the medical center). Thus, each organization specifies the security policy for its devices and structures them in security domains.

The client organization *COr* is in charge of the entity proceeding to the resource request and thus, must specify security policies for *C*, including with whom *C* is allowed to communicate. This means that *COr* has to define authorized sources for a resource *R*. *COr* also configures *C* and *CAE* in order to make them belong to the same security domain.

The resource Organization *ROr* belongs to the same security domain as *R* and *RS*. *ROr* is in charge of *R* and *RS* and thus, must specify the authorization policies for *R* and decides with whom *RS* is allowed to communicate. That means that *ROr* has to configure if and how an entity with certain attributes is allowed to access *R*. *ROr* also configures *RS* and *RAE* in order to make them belong to the same security domain.

Subsequently, on the client side, *COr* defines authorized sources for *R*, and on the Resource side, *ROr* configures if and how an entity can access *R*.

In orders to do this, *ROr* and *COr* must have already agreed on the terms of such a service and on how to organize and structure this collaboration. An agreement is passed between the two entities before this interaction takes place (see Collaboration layer: a cross domain access control).

Note that an RS may in some cases be also the *RAE*. This holds in the same way for the *C* and the *CAE*.

4) Collaboration layer: a cross domain access control

As seen in the case study above, cross domain interaction is fundamental in the IoT. Furthermore, this characteristic represents the main difference between the Internet of Things paradigm and a simple sensor network based service that usually only deal with one domain. Note that, throughout this study, we define a domain as a structured independent organization.

Unfortunately, as seen above, the OrBAC access model does not handle the collaborative interaction aspects. To overcome this limitation, SmartOrBAC enhances OrBAC with new collaboration related concepts. This issue is addressed at the collaboration layer, by making a prior agreement between the involved organizations (as shown in Fig. 3) where the access rules to a given resource are jointly defined according to the OrBAC format by organizations that interact.

In order to manage this new agreement, the entity, located in the Organization layer, called *Principal Authorization Manager* "PAM" will be used. From the *RS* point of view, this agreement, which is interpreted in terms of access rules, will be treated just like all the other rules concerning local interactions. The complexity of the external interaction authorization management is hidden from the end constrained device, which keeps the same authorization processing no matter the nature of the client. This abstraction is made possible by the establishment of a fourth layer that manages the cooperation between different organizations.

Basically, SmartOrBAC begins with the publication and negotiation of collaboration rules as well as the corresponding access control rules. First, each organization determines which resources it will offer to external partners, and then references them into the *PAM*. At this point, other organizations can contact it to express their wish to use this specific referenced resource. To do that, the *COr* and the *ROr* negotiate and come to an agreement concerning the use of the resource *R*. Then, they establish a contract and jointly define security rules concerning access to *R*. The *COr*'s and *ROr*'s exchange format and the contract aspect will be discussed in a future paper. In the rest of this section, let us focus on access control rules. These rules are registered -according to an OrBAC format- in the *PAM* of both organizations. Parallel to this, *COr* creates locally a "*virtual resource*" called *R_image* which represents (the remote) *R*, and adds a rule in its OrBAC base to define which entities can invoke *R_image* to use *R* (see Fig. 3 and 4).

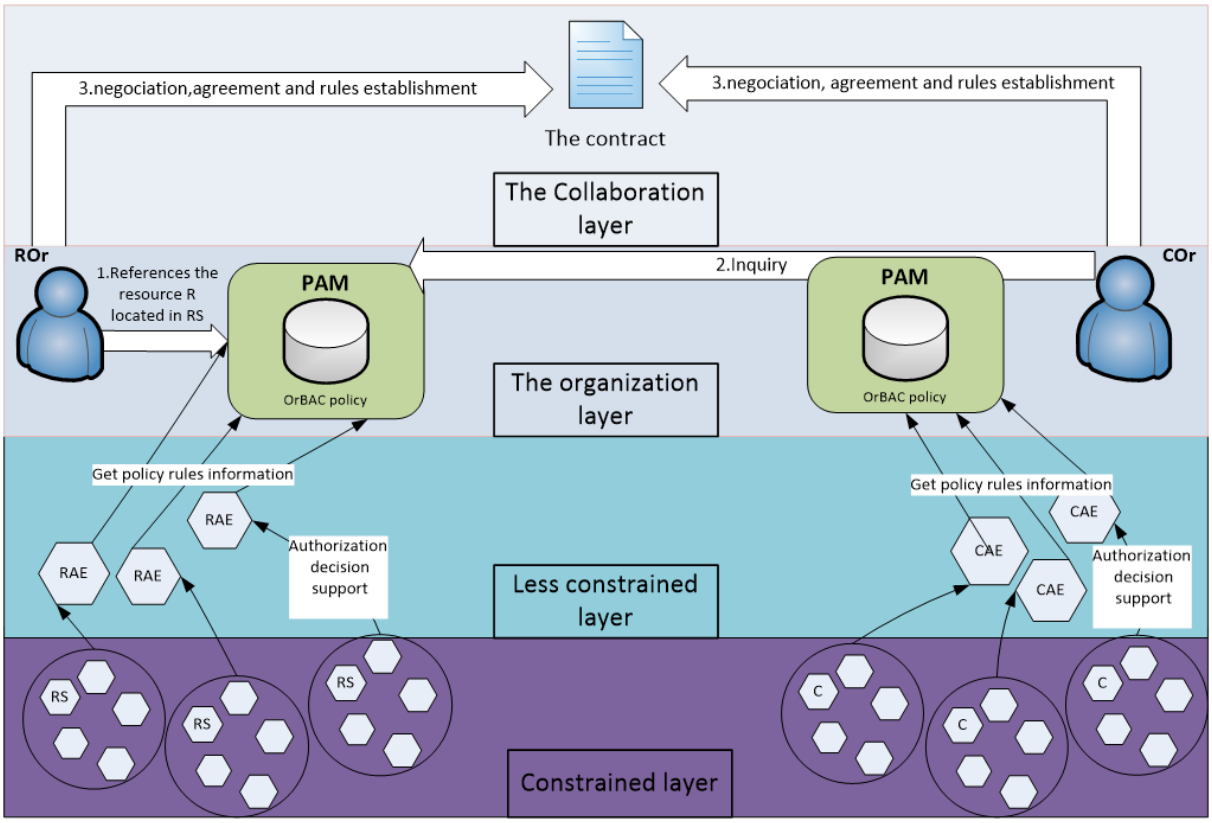

Fig. 3. Management of cross domain requirement in IoT environment

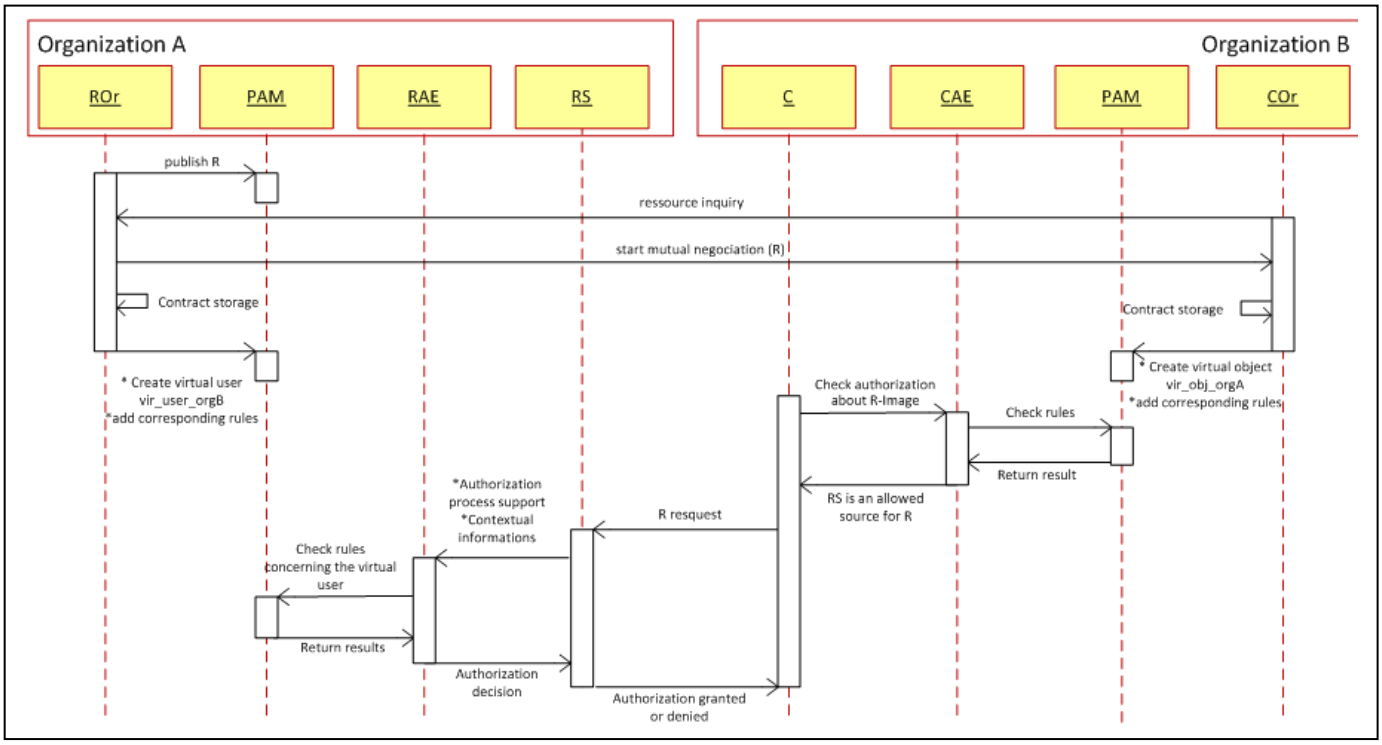

Fig. 4. Sequence diagram shows interactions between entities presented above when a Client from Organization B requests a resource from Organization A

B. Enhancing OrBAC for context awareness

Unlike traditional services where the concept of context is limited to a finite set of use cases, in the IoT environment, the concept is getting wider by taking on an ambient character in order to allow services taking into account the contextual information collected in real time by the different sensors [46]. The Context used in defining the SmartOrBAC rule is a group of contexts (C_{Set}) from several types (C_{Type}). The type of context represents a concrete characteristic such as location, temperature or time, but also security requirement such as trust level or risk level. Subsequently, to introduce the context in the access control decision, a value called (C_{Const}) is given to each C_{Type}. Thus the context definition in SmartOrBAC takes the following format:

$$C_{Type} = \{\text{authLevel, trustLevel, time, location,…}\} \quad (1)$$

$$C_{Set} = \{C_{Type(1)}, C_{Type(2)},\dots , C_{Type(n)}\} \quad (2)$$

$$C_{Const} = \langle C_{Type}\rangle \langle \text{OP}\rangle \langle \text{VALUE}\rangle \quad (3)$$

where OP is a logical operator, i.e. OP $\in \{>,<,\geq,\leq,=, \neq\}$, and VALUE is the estimated level of C_{Type}. Finally, C is expressed as a set of constraints C_{Const} as follows:

$$C = \{C_{Const(1)}, C_{Const(2)},\dots, C_{Const(n)}\} \quad (4)$$

Typically in the previous use case, the emergency context would be defined by a set of constraints related to the patient movement, location and especially to his heartbeat measures.

C. Scenario

In order to illustrate SmartOrBAC, the different concepts detailed above are applied on the previous case study.

First of all, each organization determines which device's resources it will offer to external partners. At this stage, we find in the *PAM* of John's smarthome organization resources such as the heart monitor resource. The medical center organization makes an inquiry to the *PAM*. As soon as the target resource is found, the negotiation phase begins between the *ROr* of the smart home and the *COr* of the medical center. The resulting contract is then transcript in terms of authorization rules regarding the OrBAC format for both of the medical center and smart home of John. More precisely, if the agreement between the two organizations is: "*Assigned doctor from medical center have the permission to remotely actuate the implanted cardioverter defibrillator from the heart monitor device in the heart attack emergency context*", the *ROr* of Smart home should:

- have (or create) a rule that grants the permission to a certain role (e.g. *Doctor*) to actuate the heart monitor: *Permission(smart home, Doctor, vital equipement, Actuating, $C_{heart\_attack\_Emergency}$)*; Note that, from John's smart home's point of view, every user playing the "*Doctor*" role will have this permission;
- create a "virtual user" noted "v_user_doctor" that represents the medical center for its use of the implanted cardioverter defibrillator (see Fig. 5);
- add the following *Empower(smart home, v_user_doctor, Doctor)* association to its rule base. This rule grants the user medical center's doctor the right to play the Doctor role.

In parallel, the COr of the medical center creates locally a "virtual object" heart_monitor_image which represents the (remote) implanted device (the resource made available by

John's Smart Home), and adds a rule in its OrBAC base to define which of the medical center's roles can invoke heart_monitor_image to use the real heart monitor.

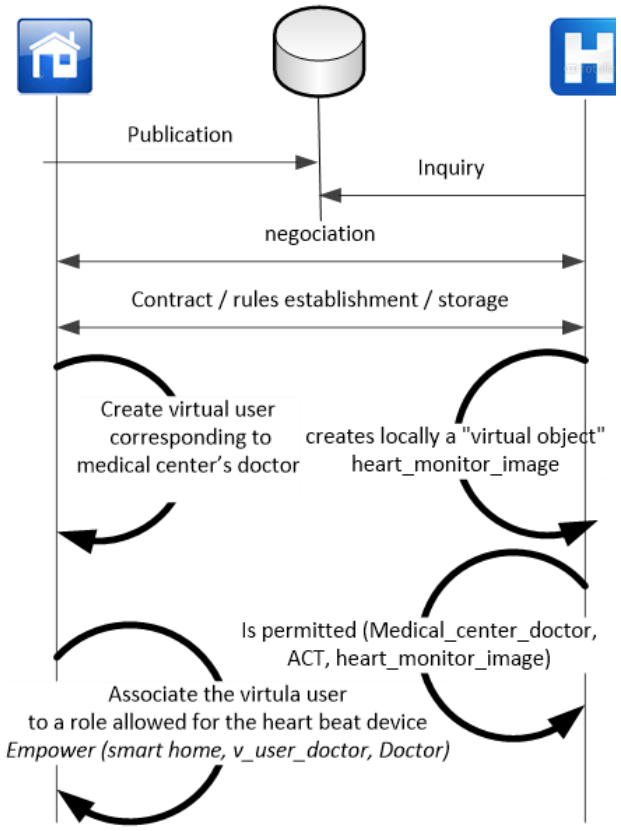

Fig. 5. Virtual user and virtual Object in SmartOrBAC

The derivation of the permission (i.e., instantiation of security rules) mentioned above can be formally expressed as follows:

Permission *(smart home, Doctor, vital equipement, Actuating, Cheart_attack_Emergency)* ∧
Empower *(smart home, v_user_doctor, Doctor)* ∧
Consider *(smart home, ACT, Actuating)* ∧
Use *(smart home, John's heartbeat, vital equipement)* ∧
Hold *(smart home, medical center's doctor, ACT, John's heartbeat, Cheart_attack_Emergency)*

→ ***Is permitted*** *(v_user_doctor, ACT, John's heartbeat)*

Fig. 6. Derivation of permissions in SmartOrBAC

Let's assume that the assisted living dispositive is a set of different devices (sensors and actuators) with different capabilities. We also assume that the specific device *RS* of heart monitoring that the medical center tries to access is located in the constrained layer, such as the client device *C* used by the doctor in the medical center. The link between the *RS* and its corresponding *RAE* located in the less constrained layer has already been configured by the *ROr* of John's smart home. The same applies for the *CAE* and *C* that have been already configured by the *COr* of medical center.

Before the doctor's device *C* in the medical center sends an actuating request to the heart monitoring device *RS*, it asks the corresponding CAE in the medical center for assistance in order to determine if the local image of *RS* (*heart_monitor_image*) is an authorized source.

At this moment, *CAE* starts evaluating the authorization policy rules, using as object the *heart_monitor_image*. Note that at this level, the external nature of the heart monitor device is unknown. Then, if information about policy rules are needed, a request is sent to the *PAM* of the medical center. Once this process is completed, if *RS* is an allowed source, an actuating request is directly sent to the heart monitoring device.

Once the request is received, the authorization decision process begins on the smart home organization side. For that, the device sends an authorization process request, with contextual information, to the corresponding *RAE* in john's smart home. The latter evaluates the authorization decision regarding authorization rules in John smart home's *PAM* - especially those detailed above where the subject is *v_user_doctor* -.The result is sent to *RS* which, in turn, sends an access response to the doctor's device.

VII. Complexity of SmartOrBAC

In this section, the complexity of the capabilities based models (frequently proposed for IoT) is compared to SmartOrBAC on the three following aspects of the access control process:

1) Operations related to the access control decision in an IoT environment.
2) Operations related to the security policy update.
3) The potential risk of errors during the access control policy administration.

Our aim (at this stage) is to demonstrate that even though SmartOrBAC is multifaceted, it is less complex than capabilities based models, it ameliorates the security policy management cost and it reduces the risks of errors.

B. Access control decision

In order to evaluate the complexity of the complete access control management process, we focused on two crucial parameters:

- Quantify the number of decisions required for the definition of the access control policy.
- Rate the complexity of each decision (each management operation).

It is clear that, the more complex the operations needed, the more management resources and processing times are required, and a higher probability of errors is observed which is even more prominent in an IoT environment. Furthermore the operations assigned to the security administrator are naturally more complex and more sensitive than those done by other actors. Therefore in our context, we identify two kinds of operations:

1) Sensitive operations assigned to the security administrator; we note D the cost of this kind of operations.

2) Secondary operations that may be executed by an operator not necessary aware of the security requirements; we note d the cost of such operations (e.g. the assignment of subject to roles in SmartOrBAC in a certain organization).

Next follows a comparison of the administration costs of the capability based models and the SmartOrBAC.

3) Core Capability based models

In the capability based models, the overall quantity of operations is |SUB|.|Op|.|OBJ; where|.| is the number of elements (the cardinal). As the quantity of operations is limited, |Op| = *constant*. Thus the number of operations is equal to |SUB|.|OBJ|.|*constant*|. However, this analysis is correct for traditional services but not for the IoT environment where the contextual information collected in real time by sensors has to be taken into account. As seen in "Enhancing OrBAC for context awareness", the concept of context is no more limited to a finite set of use cases. In order to better represent this reality, the above formula should be correlated to the multiplicity of existent contexts. Thus, the total number of operation would rather be |SUB|.|Op|.|OBJ|.|CONTEXT|.

Assuming that n designates the maximal value of |SUB|, |OBJ| and |CONTEXT|, the number of the operations is $O(n).O(n).\ O(n).O(1) = O(n^3)$. Furthemore, all these operations require the administrator skills, consequently, the cost is equal to $D.O(n^3)$.

4) SmartOrBAC

Prior to the calculation of the administration cost, let us fisrt recall how the access decision is made in SmartOrBAC according to OrBAC:

$\forall org \in$ ORG, $\forall s \in$ SUBJ, $\forall \alpha \in$ ACTION,
$\forall o \in$ OBJ, $\forall r \in$ ROLE,
$\forall a \in$ ACTIV, $\forall v \in$ VIEW, $\forall c \in$ CONT,
Permission (org, r, v, a, c) $\wedge$
Empower (org, s, r) $\wedge$
Consider (org, α, a) $\wedge$
Use (org, o, v) $\wedge$
Hold (org, s, α, o, c) $\wedge$
$\rightarrow$ *Is permitted(s, α, o)*

$$Cost = D.\mathrm{C}(RULE) + d.[\ \mathrm{C}(Empower) + \mathrm{C}(Consider) + \mathrm{C}(Use) + \mathrm{C}(Hold)] \quad (5)$$

while:

$$\mathrm{C}(\mathrm{RULE}) = |\mathrm{Access\_Mode}| + |\mathrm{ORG}| + |\mathrm{ROLE}| + |\mathrm{VIEW}| + |\mathrm{ACTIV}| + |\mathrm{CONT}|$$

$$= |\mathrm{constant}| + |\ \mathrm{constant}\ | + |\ \mathrm{constant}\ | + |\ \mathrm{constant}| + |\ \mathrm{constant}\ |+ \mathrm{O(n)} \approx \mathrm{O(n)} \quad (6)$$

$$\mathrm{C}(Empower) = |\mathrm{ORG}| + |\mathrm{ROLE}| + |\mathrm{SUBJ}|$$

$$= |constant| + |\ constant\ | + O(n) \approx O(n) \quad (7)$$

In the same way,

$$\mathrm{C}(Consider) \approx O(n),\ \mathrm{C}(Use) \approx O(n) \text{ and } \mathrm{C}(Hold) \approx O(n) \quad (8)$$

Hence,

$$Cost \approx D.\ O(n) + \mathrm{d}\ .\ [O(n) + O(n) + O(n) + O(n)\ ]$$

$$\approx D.\ O(n) + \mathrm{d}.O(n) \quad (9)$$

TABLE I. THE COMPLEXITY OF DECISIONS

Capability based models	SmartOrBAC
D.O(n3)	D. O(n) + d.O(n)

In the capability based models, the complexity is a cubic function with a higher factor (D), while in SmartOrBAC it is a linear function with two factors D (major) and d (minor). Subsequently, SmartOrBAC leads to significant reduction of the management complexity.

C. The update of the security rules

In the capability based models, the subject's permission management (adding, updating or deleting) is in the order of $O(n)$. Moreover, all these operations require a major decision. The total cost is thus $D.O(n)$ correlated to contextual parameters it is upgraded to $D.O(n^2)$. Inversely, in SmartOrBAC the management of subjects' permissions corresponds to a change of its roles which involves only $d.O(1)$ operations. Now, if we concentrate our analysis on objects, and in particular, if we want to change the permissions associated to a certain object, the cost in the capability based models is $D.O(n^2)$. In SmartOrBAC, the cost is $d.O(1)$ as we only review the *Use* relationship.

TABLE II. THE COMPLEXITY OF ACCESS CONTROL CHANGES

Capability based models	SmartOrBAC
$D.O(n^2)$	$d.O(1)$

D. Risk of errors

Generally, two indicators are used to evaluate the risks:

- The severity of the threat, estimated in term of its impact.
- The potentiality of the threat (estimated in term of frequency or probability if the cause is accidental, or in term of feasibility if the cause is deliberate).

Basically, the security administrator decisions are more sensitive than the operator or a secretary decision. In the first case, we note the severity by ***S***, while we denote it by ***s*** in the second case.

Besides, the potentiality (e.g. probability) that an administrator makes an error in the security policy definition is lower than the action of the operator or a secretary. Indeed, we note ***p*** the potentiality of the security administrator errors, while ***P*** denotes the potentiality of the operator/secretary errors.

We can thus conclude that:

- The assignments that could be done by a secretary or an operator has a factor of risk (of error) equivalent to s.P

- The factor is equivalent to S.p for Sensitive operations that are done by the security administrator.

Therefore, to calculate the risk of error for the two models (capability based and SmartOrBAC), we only replace *D* in table 1 by *S.p* and *d* by *s.P*. Table 3 summarizes the risk of errors in the three compared models.

TABLE III. THE RISK OF ADMINISTRATION ERRORS

Capability based models	SmartOrBAC
$S.p.O(n^3)$	$S.p.\ O(n) + s.P.O(n)$

The risk of access control management errors is thus reduced in SmartOrBAC.

Consequently, compared to the other models, it appears that not only SmartOrBAC gains simplicity and clarity in the IoT environment (e.g. by taking the context into account in the earlier stages), but it also greatly reduces the cost of administering access control policies as well as making the process less error prone while being clearly context aware.

VIII. IMPLEMENTATION

The transmissions between the different entities included in our Framework (C/RS, C/CAE, RS/RAE) are done via the CoAP [39] protocol (Constrained Application Protocol), which is a specialized web transfer protocol that is intended for use in resource-constrained internet devices. Like HTTP, CoAP is based on the wildly successful REST model: Servers make resources available under a URL, and clients access these resources using methods such as GET, PUT, POST, and DELETE.

Since the XML representation is too verbose for efficient transmission over limited channels, thus JSON-based notation is used for authorization requests and responses. In fact JSON [38] (JavaScript Object Notation) is a lightweight data-interchange format that efficiently reduces the size of the transmitted messages between *C* and *RS* devices and optimizes the processing time.

The device part of our framework (especially *C* and *RS*) was implemented on an example platform: The Arduino Mega 2560 board3. This board features a 16 MHz processor, 256 kB of Flash Memory, 8 kB of SRAM, and 4 kB of EEPROM. The choice of this board is made in order to test our approach on the lowest performance of the end constrained devices.

The board was programmed in JAVA using a custom implementation of the CoAP protocol stack and the assertions were wrapped in JSON format using the standard Java API (javax.json.*).

IX. CONCLUSION

Our SmartOrBAC access model is specifically designed for the IoT environment and it is conceived through an abstraction layer design that makes use of a deep understanding of the IoT paradigm as it is used in the real world. For these smart services, contextual information is a leading element in decision making therefore only a real-time consideration of this information will achieve smartness. For this reason, the "context" notion (originally present in OrBAC) is extended in order to fit the IoT requirements.

Understanding that users belonging to an organization need to dynamically access resources controlled by other organizations the proposed model is extended with specific collaborative mechanisms where the same OrBAC security policy can be used for local as well as external access. In this way, SmartOrBAC improves the management of the security policy and reduces considerably its complexity.

In our future work, the focus willl be laid on making the SmartOrBAC model more effective by going deeper in the study of the negotiation process and the e-contract format. Other possibilities include incorporating a secure authority delegation method based on OrBAC in order to control the link between the end device and the *RAE*/*CAE* in order to make it more dynamic.

Finally, another relevant research line related to this work is the consideration for additional privacy enhancement through techniques such as the use of pseudonyms or anonymous assertions.

REFERENCES

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness" in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, ser. HUC '99*. London, UK: Springer-Verlag, 1999, pp. 304–307.

[2] European Commission, "Internet of things in 2020 road map for the future" *Working Group RFID of the ETP EPOSS, Tech. Rep*., May 2008, http://ec.europa.eu/information society/policy/rfid/ documents/iotprague2009.pdf [Accessed on: 2011-06-12].

[3] Istepanian, R.S.H, Jara, N. Philips and A. Sungoor. "Internet of things for M-health applications (IoMT)". In AMA IEEE Medical Technology Conference on Individualized Healthcare, Washington (2010).

[4] A. Jara, A. Skarmeta and M. Zamora : "An Internet of Things based personal device for diabetes therapy management in ambient assisted living (AAL)". Personal and Ubiquitous Computing, 15(4):431–440 (2011).

[5] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey" Computer Networks, vol.54, no.15, pp. 2787–2805, Oct. 2010.

[6] J. Santa, A. Skarmeta, A. Jara, and M. Zamora: "Telematic platform for integral management of agricultural/perishable goods in terrestrial logistics". In Computers and Electronics in Agriculture, 80:31–40 (2012).

[7] L. W. F. Chaves and C. Decker, "A survey on organic smart labels for the internet-of-things" in Networked Sensing Systems (INSS), 2010 Seventh International Conference on, 2010, pp. 161–164.

[8] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: sensor networks in agricultural production" Pervasive Computing, IEEE, vol. 3, no. 1, pp. 38 – 45, jan.-march 2004.

[9] V. Gungor, T. Kocak and al.: "Smart Grid and Smarthomes: Key Players and Pilot Projects". In Industrial Electronics Magazine, IEEE, 6(4):18–34 (2012).

[10] J.W. Hui and D.E. Culler. IP is dead, long live IP for wireless sensor networks. *In Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 15-28, Raleigh, NC, USA, 2008.

[11] Mathias Kovatsch, Markus Weiss, and Dominique Guinard. Embedding internet technology for home automation. *In Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference* on, pages 1-8, Bilbao, Spain, September 2010. IEEE.

[12] M. Castro, A. Jara, and A. Skarmeta.Smart Lighting Solutions for Smart Cities. In Proc. of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA'13), Barcelona, Spain, pages 1374–1379. IEEE, March 2013.

[13] L. Daigle and O. Kolkman. "The OAuth 2.0 Authorization Framework" Internet Engineering Task Force (IETF), Request For Comments (RFC) 6749, October 2012, http://www.ietf.org/rfc/rfc6749.txt.

[14] Xun Li and Sang Bong Yoo, "Extended Role-Based Security System using Context Information", *In: 2008 Second International Conference on Future Generation Communication and Networking*, pp. 7-12.

[15] Guillemin and P. Friess, "Internet of things strategic research roadmap" The Cluster of European Research Projects, Tech. Rep., September 2009.

[16] L. Daigle and O. Kolkman, "The OAuth 2.0 Authorization Framework," Internet Engineering Task Force (IETF), Request For Comments (RFC) 6749, October 2012, http://www.ietf.org/rfc/rfc6749.txt.

[17] M.Memon, S.Rahr Wagner, C. Pedersen, F. Beevi and F.Overgaard Hansen etc. Ambient Assisted Living Healthcare Frameworks, Platforms, Standards, and Quality Attributes. *Sensors* ***2014****, 14, 4312-4341; doi:10.3390/s140304312.*

[18] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier, "The internet of things for ambient assisted living," *in Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, 2010*, pp. 804–809

[19] Xun Li and Sang Bong Yoo, "Extended Role-Based Security System using Context Information", *In: 2008 Second International Conference on Future Generation Communication and Networking*, pp. 7-12.

[20] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle, "Vision and challenges for realising the internet of things," *European Commission Information Society and Media, March 2010.*

[21] G.Zhang, J.Tian. An Extended Role Based Access Control Model for the Internet of Things. *In 2010 International Conference on Information, Networking and Automation (ICINA).*

[22] A. Bernardos, P. Tarrio, and J. Casar, "A data fusion framework for context-aware mobile services," *in Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on, aug. 2008*, pp. 606 –613.

[23] L. Lin, "Application of the internet of thing in green agricultural products supply chain management," in Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on, vol. 1, 2011, pp. 1022–1025.

[24] D. Martin, C. Lamsfus, and A. Alzua, "Automatic context data life cycle management framework," in Pervasive Computing and Applications (ICPCA), 2010 5th International Conference on, dec. 2010, pp. 330 – 335.

[25] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The Gator Tech Smart House: a programmable pervasive space. *Computer*, 38(3):50-60, 2005.

[26] C.Floerkemeier, M.Lampe. Facilitating RFID development with the accada prototyping platform. *In Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom '07)*, pages 495-500, New York, NY, USA, 2007. IEEE Computer Society.

[27] Christian Floerkemeier, Christof Roduner, and Matthias Lampe. RFID application development with the Accada middleware platform. *IEEE Systems Journal*, 1(2):82-94, December 2007.

[28] Deze Zeng, Song Guo, and Zixue Cheng. The Web of Things: A Survey. *Journal of communications*, vol.-, No.6, September 2011.

[29] F. Ramparany, R. Poortinga, M. Stikic, J. Schmalenstroer, and T. Prante, "An open context information management infrastructure the ist-amigo project," in *Intelligent Environments, 2007. IE 07. 3rd IET International Conference* on, sept. 2007, pp. 398 –403.

[30] Christian Floerkemeier, Christof Roduner, and Matthias Lampe. RFID application development with the Accada middleware platform. *IEEE Systems Journal*, 1(2):82-94, December 2007.

[31] G. Zhang and W. Gong. The Research of Access Control Based on UCON in the Internet of Things. Journalof Software, 6(4):724–731, April 2011.

[32] S. Gusmeroli, S. Piccione, and D. Rotondi. A capability-based security approach to manage access control in the internet of things. Mathematical and Computer Modelling, 58(5-6):1189–1205, Sept 2013.

[33] HERNÁNDEZ-RAMOS, J., JARA, Antonio J., MARIN, Leandro, etal. Distributed capability-based access control for the internet of things. J Internet ServInfSecur, 2013, vol. 3, no 3/4.

[34] B. Bayu, P. N. Mahalle, N. R. Prasad, and R. Prasad. Capability-based access control delegation model on the federated IoT network. In Proc. of the 15th International Symposium on Wirless Personal Multimedia Communications (WPMC'12), Taipei, China, pages 604–608. IEEE, September 2012.

[35] K. Hasebe and M. Mabuchi, "Capability-role-based delegation in workflow systems," in Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on, dec. 2010, pp. 711 – 717.

[36] L. Seitz, G. Selander, and C. Gehrmann. Authorization Framework for the Internet-of-Things. In Proc.of the 14th IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'13), Madrid, Spain, pages 1–6. IEEE, June 2013.

[37] T. Moses. Extensible Access Control Markup Language (XACML) Version 2.0, 2005.

[38] D. Crockford. RFC 4627: The application/json Media Type for Javascript Object Notation (JSON). IETFRFC 4627, July 2006. http://www.ietf.org/rfc/rfc4627.txt.

[39] Z. Shelby, K. Hartke, and C. Bormann. Constrained Application Protocol (CoAP). IETF Internet-draft (work in progress), June 2013.http://tools.ietf.org/html/draft-ietf-core-coap-18.

[40] R. Prasad, My personal Adaptive Global NET (MAGNET), *ser. Signals and Communication Technology Book*. Springer Netherlands, 2010.

[41] J.W. Hui and D.E. Culler. Extending IP to low-power, wireless personal area networks. IEEE Internet Computing, 12(4):37-45, 2008.

[42] N. Kushalnagar, G. Montenegro, and C. Schumacher, Ipv6 Over Low-Power Wireless Personal Area Networks (6lowpans): Overview, Assumptions, Problem Statement, and Goals, RFC4919, August, 2007.

[43] R. Roman, J. Zhou, and J. Lopez. On the Features and Challenges of Security and Privacy in Distributed Internet of Things. Computer Networks, 57(10):2266–2279, July 2013.

[44] IEFT, "Constrained restful environments working group (core wg) - authentication and authorization for constrained environments (ace) mailing list".

[45] C. Bormann, M. Ersue, A. Keranen. Terminology for Constrained-Node Networks. Informational, IETF RFC 7228, May, 2014. http://tools.ietf.org/html/rfc7228.

[46] B. Bayu, P. N. Mahalle, N. R. Prasad, and R. Prasad. Capability-based access control delegation model on the federated IoT network. In Proc. of the 15th International Symposium on Wirless Personal Multimedia Communications (WPMC'12), Taipei, China, pages 604–608. IEEE, September 2012.