Rivera, D. & Cruz-Piris, L. 2019, "Privacy in smart spaces: protecting information in workplaces", in Transforming ergonomics with personalized health and intelligent workplaces, IOS Press, pp. 47-61

Available at http://dx.doi.org/10.3233/978-1-61499-973-7-47

(Article begins on next page)

February 2019

Privacy in Smart Spaces: Protecting information in workplaces

The final publication is available at IOS Press through
http://dx.doi.org/10.3233/978-1-61499-973-7-47

Diego RIVERA [a,1], and Luis CRUZ-PIRIS [a]
[a] *Departamento de Automática, Universidad de Alcalá, Spain*

Abstract. The workplaces organization has evolved during the last decades from individual private offices to open spaces, which offer a higher flexibility degree when there are important changes in firms' necessities. In these open spaces, the optimization of environmental conditions are an essential factor to achieve employees' adequate comfort levels to develop their tasks. The rise of Internet of Things (IoT) technologies has cheapened and extended the processes automation in tasks such as air conditioning, lighting control, etc. New challenges have arisen from this new scenario, where factors such as privacy and access control for personal information are crucial. In this work, we show the application of an access control system which is able to operate with different communications protocols to a use case based on a smart office. We have developed a prototype of the devices which would control the sensors and actuators in the system, and we have carried out a series of experiments to measure the delay added to communication when using the access control techniques. The results demonstrate the validity and feasibility of the system.

Keywords. Ergonomics, Privacy, Internet of Things, Access Control, Smart Spaces

1. Introduction

The workplace is one of the places where people spend more hours each day. In highly developed countries, workplaces are mainly composed of offices. In the last years, there has been a tendency to build open spaces to make more flexible environments, following the workload demand. Furthermore, these open spaces favor the communications between work teams and the usage of agile working methodologies. Despite the important advantages of these spaces, they present new challenges from the user ergonomics point of view, given that the preferences of each employee can collide with their workmates. In the last years, the gradual conversion of open spaces in smart open workplace scenarios due to the implantation of pervasive technologies determines new scenarios to be explored by research.

The rise of Internet of Things (IoT) and the availability of sensors and wearable technology have enabled the use of new information to provide a personalized experience

[1] Corresponding Author: Diego Rivera, Universidad de Alcalá, Spain, Alcalá de Henares, Madrid, Spain; E-mail: diego.rivera@uah.es.

in spaces where they are used [1]. The IoT paradigm has been pointed out as one of the main enabler technological approaches to provide smart spaces services [2]. On the other hand, personalized smart spaces are strongly related with ergonomics and user comfort, as it has been stated by many proposals [3,4], and the possibilities of using pervasive technologies to build smart spaces related with human factors and ergonomics have been stated in works such as [5].

Nowadays, smart spaces are composed of a high number of heterogeneous interacting technologies. Usually, installed sensors and actuators coexist with wearable devices and mobile applications, leading to important challenges in terms of interoperability [6]. Although the possible benefits of such smart spaces are almost as heterogeneous as the technologies composing them, they usually involve providing services composed based on the information gathered through the sensors and devices. These services can be designed, for instance, to monitor or improve health issues [7], optimize one or more parameters of the space (examples of parameters are the energy consumption [8] or the thermal status of a building [9]), or provide new comfort services such as personalized work or study environments [10,11], in which the preferences of the users are taken into account to increase their comfort and satisfaction.

Specifically, workplaces are a recurrent scenario in the literature to illustrate the benefits of pervasive or ubiquitous computing to build smart spaces. For instance, in [12] the authors propose a framework for the construction of user-centered smart offices, and different workplace-related scenarios are used as use cases for the platform proposed in [13]. In [14], there is a study of the conflicts arisen in these environments, while in [15], the authors consider the barriers which have to be addressed when introducing sensor and wearable technologies in the workplace.

Regardless of the specific goal and technology used in smart spaces, one common characteristic of such scenarios is that they usually require a high amount of diverse information to function properly. Much of this information is user-related in one way or another, and in many cases, it can be personal sensitive information [16]. Therefore, privacy and access control are some of the most challenging issues in smart environments [17], as in IoT scenarios in general. Privacy is not only a matter of smart or IoT environment, but it has also been identified as a major concern in workplaces, even in traditional open spaces without pervasive capabilities [18]. Moreover, the social and architectural aspects of the office and the employees also contribute to determining the privacy level on pervasive smart spaces [19].

There are many proposals in the literature regarding different aspects of the security and privacy in IoT environments [20], and controlling efficiently the access to the users' data is one of the issues which has been attracting more attention from researchers [21]. Moreover, privacy has been addressed in smart spaces specifically. In [22], for instance, a privacy policy enforcement system to avoid personal information exposure in context-aware spaces. The relationship between human factors and Internet of Things environments is studied in [23]. That paper concludes that it is necessary to design user-centered mechanisms to correctly assess the privacy and security mechanisms in such scenarios, as it is crucial that the users easily determine who access to their data and with which conditions.

In this work, we address the problem of protecting personal information in a smart workplace scenario which is intended to improve user comfort in offices. To achieve this we use an access control mechanism which is based on the work published in [24]. It proposes the protection of the IoT-specific communication channels by modeling them as resources and then applying OAuth-based techniques (*i.e*. User-Managed Access, UMA [25]) to provide a user-centered and easy-to-use access control mechanism. Here, we define and protect a smart space based on the deployment of a pervasive system in an open office similar to the one shown in Figure 1. This environment would be composed of sets of sensors and actuators, and a decision-making system to assign the best possible work stations to each employee, and ultimately increase the comfort in the space.

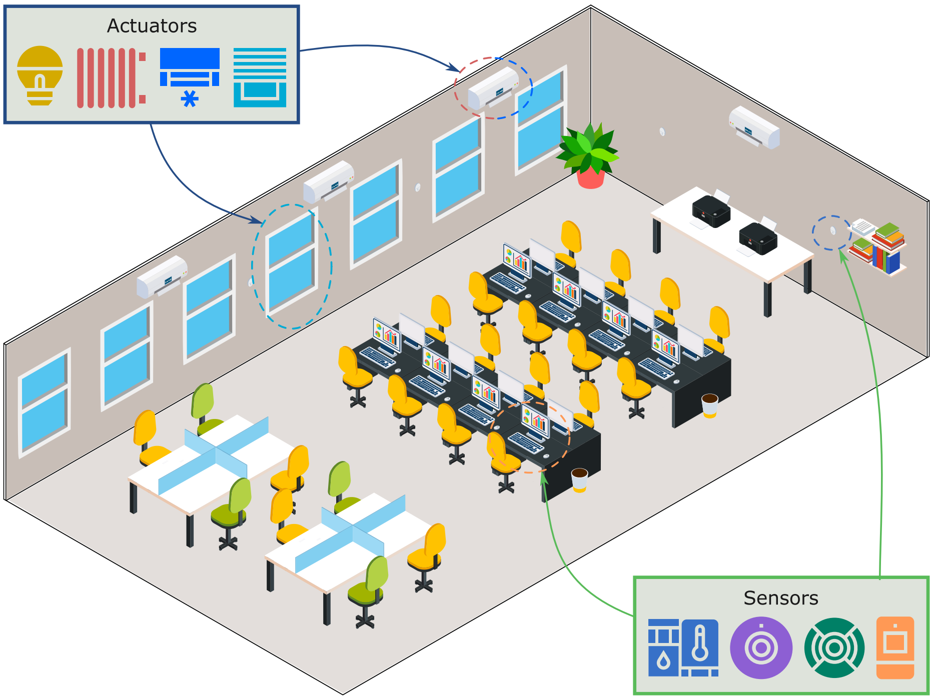

Figure 1. Example of a smart space based on an open office. Green arrows show examples of deployed sensors and blue arrows show examples of actuators (Figure based on icons created by macrovector from www.freepik.com and by Freepik from www.flaticon.com).

In the following sections, we describe in more detail the background technologies used for the protection of the data in the smart space scenario (section 2), we define the scenario in which we are going to apply these technologies (section 3). The application of access control mechanisms in the proposed scenario has been the subject of tests which are described in section 4 along with the obtained results. Finally, in section 5, we discuss the results obtained.

2. Background technologies

2.1. IoT-related architectures and protocols

The fast development of IoT-related systems has to lead to a high heterogeneity in technologies, architectures, and protocols which coexist nowadays in different degrees of integration and interoperability. In [26] there is a survey on the architectonic approaches, the functional blocks, and the main technologies related to these systems. Most of the IoT architecture proposals are layer-based (usually classified in three layers and five layer architectures, although this depends on the actual model [27]) and determine a series of protocols and mechanisms to provide specific services in each layer.

Communications protocols are, therefore, a fundamental part of any IoT system. Even more, because there is a departure from the traditional Internet network architectures. From the physical layer (where protocols and mechanisms have been designed to offer low-powered low-rate wireless communications) to the top application layer, there are many proposals to communicate the things in these pervasive scenarios, which do not always follow the most common paradigms in the current Internet. For instance, although the use of HTTP-based communications is still extended in these scenarios [28], protocols based on the publish/subscribe are quite common to communicate the elements in smart spaces scenarios and other IoT environments, as they provide a more efficient and scalable communications model [29]. Some of the most widespread protocols of this type are “Constrained Application Protocol” (CoAP) [30] (which is actually a hybrid protocol between request/response and publish/subscribe models), “Data Distribution Service” (DDS) [31], “Advanced Message Queueing Protocol” (AMQP) [32] or “Message Queue Telemetry Transport” (MQTT) [33]. These protocols are based on exchanging messages between clients allowing them to act as message publishers or as queue subscribers. In the last two protocols, the exchange is coordinated by a centralized server called “Broker”.

Regarding access control, it is an issue widely studied issue in Internet-based services. In fact, there exist many different access control models defined in the literature such as “Role-Based Access Control” (RBAC) [34], “Attributed-Based Access Control” (ABAC) [35] or “Capabilities-Based Access Control” (CapBAC) [36], among others, which have already been used in IoT environments [37].

Apart from the different models defined to solve the access control tasks, there are a number of well-known mechanisms used on Internet nowadays and which are related to this issue. Examples of these mechanisms are the “Extensible Access Control Markup Language” (XACML) [38], or OAuth2 [39]. While the first one defines a modeling language to specify access control policies and a usage schema and it is widely used in Web services, the second one is nowadays almost a *de facto* standard in access control delegation for Web applications.

In the last years, a profile of OAuth has been developed with the goal of extending the use cases where it is applicable. It has been named “User-Managed Access” (UMA) [25] and, although it was not specifically designed to work in IoT platforms, it has been identified as an interesting approach to enable access control in such scenarios [40].

2.2. User-Managed Access in IoT scenarios

The usage of the UMA mechanism in pervasive environments can be achieved by modeling the communication procedures of publish/subscribe protocols as resources which can then be protected with user-defined policies. This proposal was explored by the authors of this work in [24] and it is summarized here.

One of the main concepts regarding the proposal is that, given that the broker is a central component of many of the publish/subscribe protocols, we can model it to be both a broker and a Resource Server in the UMA sense (see [24] or the profile specification for details on the functional entities defined by UMA [25]).

The goal of this modeling process is to integrate with one functional entity the tasks of both components (that is, the management of message queues and the interaction with authorization components. In Figure 2, we show the main components of the proposal, and the authorization process, communication process and user interaction flows.

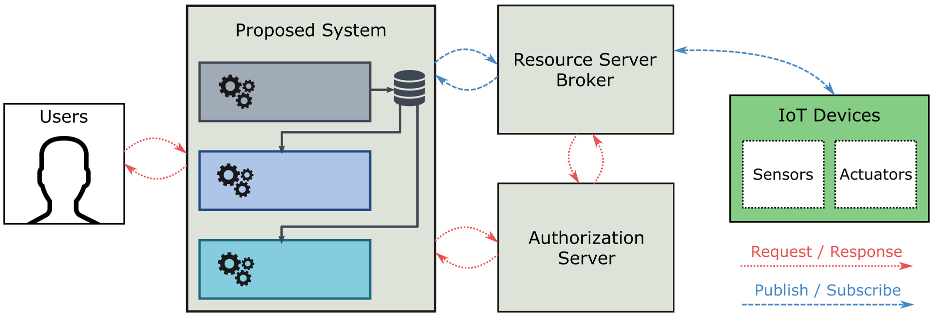

Figure 2. Communication flows between the main components of the proposed authorization system (Figure based on icons created by Freepik from www.flaticon.com).

Besides the hybrid Resource Server Broker, the access control scheme is composed of the following entities: an Authorization Server which will store and apply the user-defined control policies, and will enforce the resource protection; a client interface used by users to interact with the authorization system and to request the access token needed to access the resources from the IoT devices; and finally, the IoT devices composing the pervasive smart space (that is, devices provided with sensors and/or actuators and communications capabilities). These devices should retrieve the access token from the client to be able to communicate using the subscription or publishing actions of the specific protocol they use.

The actual protection and access to the resources (that is, to the publish/subscribe queues) are performed in a three phases scheme, in which two OAuth tokens are involved and used to access the authorization server protected APIs (Protection API Token (PAT) and Authorization API Token (AAT)):

The first phase allows the protection of a queue. In this phase, the queue must be registered as a resource in the authorization server and the policies applied to it must also be registered by the queue owner. This phase corresponds to the resource protection phase in UMA.

The second phase corresponds to the authorization request to access a specific queue. In this phase, the owner of the device requesting access to a queue must acquire a token (called Requesting Party Token (RPT)) through its client interface and then send it securely to the device. This last task can be performed manually or using the token provision mechanism described in [24]).

The third phase is the actual access to the queue to perform the specific message exchange needed by the device. The RPT token obtained in the previous phase must be used here to guarantee this access. The specific details on how issuing this token depend on the protocol used in each case, but in general, in a previously configured device, the RPT can be issued to the Resource Server Broker in the first connection messages (*e.g.* CONNECT message in MQTT) and then validated in the Authorization Server. The acquired permissions would allow to publish messages in the queue or receive them by subscribing the device to it until the connection is finished.

3. Open smart space for workplaces

Nowadays it is very common to build workplaces or offices following an open space design approach. These offices aim to reduce the separation walls as much as possible, in order to achieve a higher degree of flexibility when distributing work and tasks and a lower impact when possible organization changes are needed [41]. Among the multiple sources of comfort for employees, the appropriate air conditioning and lighting configurations in work stations are some of the most important factors to take into account to enforce ergonomics in these environments [42]. Although big spaces have always been a challenge when trying to personalize these factors according to employee's preferences, the open office spaces present even more difficulties due to the interference in the space occupied by various employees. In this section, we are going to describe a scenario in which, through the construction of a pervasive smart space based in IoT technologies, it would be possible to assign the best work station in the office to each employee, considering their personal preferences, and ultimately increasing the average comfort level in the open space. Due to the high amount and personal nature of the of the information collected, this scenario will be an interesting use case to be protected using the access control scheme defined in the previous section.

3.1. Dynamic assignment of work stations from user's preferences

The implantation of new working methodologies for teams, or the infrastructure flexibility requirements of the spaces in offices regarding the different projects or challenges taking place in each moment, have led to the construction of open space offices. In this kind of office, the private closed spaces have been reduced to meeting rooms, directive offices or specially assigned spaces such as reprography rooms, dining rooms, etc.

On the other hand, optimizing the maximum possible energy efficiency by appropriately determine the best possible ambient conditions is a goal pursued by any firm, not only due to its benefits in the firm economy but for the improvement in its public image. The evolution of IoT technologies in the last decade have improved the monitoring capabilities along with the development of automatic decision-making processes, with a limited cost of deploying the systems.

In this section, we are going to define a set of sensors, actuators and information sources which could compose the pervasive space and which could be used by a specific decision-making system to dynamically assign the work stations. In Table 1, there is a list of the main sensors used in the proposed scenario, including their location, the approximate frequency between notifications from each one and the estimated privacy level required depending on the type of information they provide.

The sets of sensors shown in Table 1 allow the system to have an accurate real-time model of the smart space status. It is important to obtain, besides the information given by sensors, the employees' preferences as they spend the day at the office. Two additional external data sources have been defined to provide this type of information:

Table 1. Configuration values of deployed sensors in the smart space.

		Temperature / humidity	**Presence**	**Sound**	**Light**
Location	*Across the office*	x		x	
	In desks	x			x
	Above work station		x		
Frequency of notifications		$\approx 5m$	When status changes	$\approx 5m$ / By threshold	$\approx 5m$
Privacy		Low	Medium / High	Medium	Low

polls about previous preferences and real-time feedback requests. The specific features of each source are detailed in the following paragraphs.

- Polls: These questionnaires are composed by a form which is asked to be filled by the employees, detailing their preferences in questions such as ideal temperature levels or lighting levels in their current work station. Additionally, the should define the tasks they usually perform during their work day and if they require any special consideration (*e.g.* Being close to printers, schedule frequent meetings, etc.).
- Real-time feedback: The system is also able to communicate with employees through a custom application. This application can, from time to time, request some feedback from employees regarding their comfort degree in their current work station. These requests are usually shorter and more specific than the ones requested in polls. For instance, a requested feedback question could be *"Is the temperature too high right now?"* or *"Is the lighting adequate for your current tasks?"*.

Finally, the smart space is completed with a set of actuators which allow to remotely control the main environmental characteristics such as the air conditioning, the lighting (both natural and artificial), etc.

3.2. Pervasive Smart space computing system

The main goal of the pervasive smart system proposed is to automatically provide a work station assignment in an open workplace, and ultimately, increase the comfort degree for employees. To achieve this, the system is feed from three data sources, as it is shown in Figure 3:

- Source A: Sensors. Installed across the office, they offer neutral information (in the sense that they are not influenced by the employees personal perceptions) about the environmental status of the smart space in each moment.
- Source B: External data sources. The generation of possible recommendations and the scheduling of these recommendations will be based on external data sources such as weather forecasts, meeting agendas, or any events planned in the office.

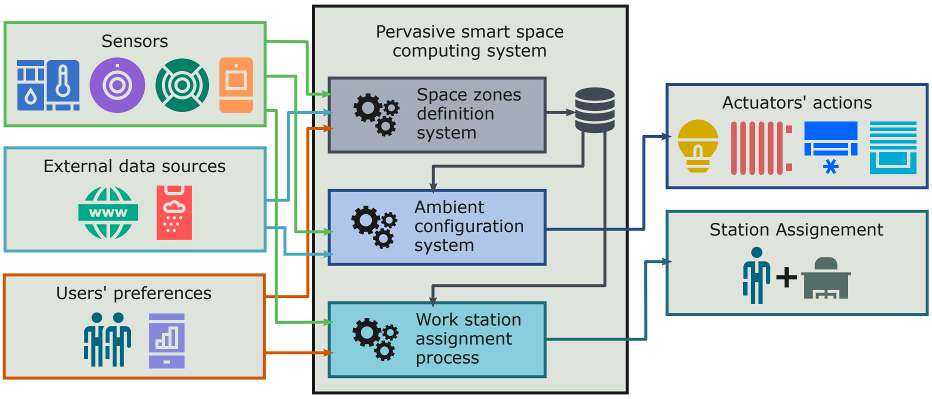
Sensors
External data sources
Users' preferences
Pervasive smart space computing system
Space zones definition system
Ambient configuration system
Work station assignment process
Actuators' actions
Station Assignement
www

Figure 3. General diagram of the smart space computing system, its data sources and its outputs (Figure based on icons created by Freepik from www.flaticon.com).

- Source C: Users' preferences. Using the poll and real-time feedback requests, the system will be able to collect employees' preferences and take them into account in the workspace assignment decision-making process.

Using the A, B, and C data sources, the space zones definition system will be able to divide the work stations into groups or zones which would comply, as much as possible, with each user preferences, ensuring this way the viability of the station assignment. Once the list of possible work station groups has been defined, the system must take the necessary actions to change the environmental characteristics of each space to adapt it to the requirements. This task is carried out by the "Ambient configuration system", which is connected to the actuators installed across space and is able to send orders to vary the ambient conditions. Finally, the "Work station assignment process" is in charge of monitoring the free work stations in each zone using the data source A, and then, assign to each employee the station closer to their personal preferences.

3.3. Resource protection in the smart office

In section 3.2 we have described the basic components of the proposed system and its modules. From the security point of view, there are two important aspects to issue: The protection of data from data sources and the protection of information flows used to request actions to the installed actuators.

The protection of incoming data flows from the specified data sources is important to assure that this information is only accessible by the authorized system modules. Some values such as those generated by temperature sensors generate flows of low requirements regarding privacy. The employees' personal preferences, however, are sensitive information which must be carefully protected. In a similar way, it is crucial to protect the data flow between the system and the actuators, not only because it could compromise the privacy of the generated information, but because the risk of an attacker altering the office conditions and the possible consequences of such attacks.

The system's devices, sensors, and actuators use a publish/subscribe communications scheme. In section 2.1 we have listed some of the most used protocols which follow this pattern, and in the case of our proposed system, we have chosen MQTT due to its simplicity and its wide use in this kind of scenarios. In Figure 3, we have shown a general diagram of the functional elements composing the smart space system. To address communications and access control, at least a broker and an authorization server should be added to these modules. The broker would oversee manage (that is received and send) messages generated by the system, while the authorization server will manage the creation and validation of the access tokens used to protect the services and resources in the system, following the policies defined by users or administrators.

3.3.1. Information flows protection

The authorization server of the system has been developed by following the UMA (the OAuth profile described in section 2.2) specification. In order to guarantee the protection of every element in the system, we are going to show the different types of information flows identified in the system according to their origin or destinations, and then we are going to describe the modules of the system related to them.

- Sensor information: The sensor publish the data they generate in a queue named as "/SENSORS/TYPE_OF_SENSOR/ID_X", where "SENSORS" identifies that the source of data is a sensor, "TYPE_OF_SENSOR" would determine the specific sensor family to which the data belongs to (*i.e.* temperature, occupation, luminosity, etc.) and "ID_X" would be the specific identifier of a given sensor. Each device in which sensors are integrated must be configured with the actions it is expected to perform: The sensor read frequency, the broker address, the identifier of the queue and the access token which allows him to publish in that queue (RPT).
- Commands to actuators: The smart devices containing system's actuators subscribe themselves to specific queues to receive the orders of the system. These queues follow the structure: "/ACTUATORS/TYPE_OF_ACTUATOR/ID_Y". These devices must be previously configured to determine the broker address, the queue identifier, the set of actions to perform depending on the received message, and the token which allow for the subscription action (RPT).
- Pervasive smart space computing system: The proposed system is based on different communication patterns. It uses the publish/subscribe model to read queues containing sensor data and sends messages to the actuators' queues to give them orders. Both tasks are based on the MQTT protocol. It must also be able to base the communications on request/response communication patterns. For instance, it uses the HTTP protocol to request external information from services such as weather forecast. It also will enable an HTTP API to receive the employees' feedback information. These communications are protected exactly in the same way that the publish/subscribe queues. The services exposed by the system are protected following the UMA specifications, and any subscriptions or requests to the messaging service are based on the RPT which is associated with the required permissions.

4. Tests and results

The use of sensors and actuators in smart spaces has been validated enough nowadays to guarantee the viability of the system in terms of its hardware deployment. Given that, instead of deploying the complete system we have designed a prototype device which is able to control sensors and actuators, and which would be able to validate the access control system. The results of these tests are focused on the delay measurement introduced by using the protection scheme in a conventional scenario.

4.1. Implementation and set-up

The design of the prototype has been focused on the implementation of the input and output communications needed by the smart space system, in order to emulate the message exchange process of the real scenario.

For the development and deployment of the entities used by the system, we have relied in Web servers, both local and cloud-based (specifically, we have used Amazon Web Services (AWS) [43]) to implement the servers (authorization server and resource server broker), Arduino devices [44] and Raspberry Pi [45] boards to implement the IoT devices. The actually built prototype can be seen in Figure 4.

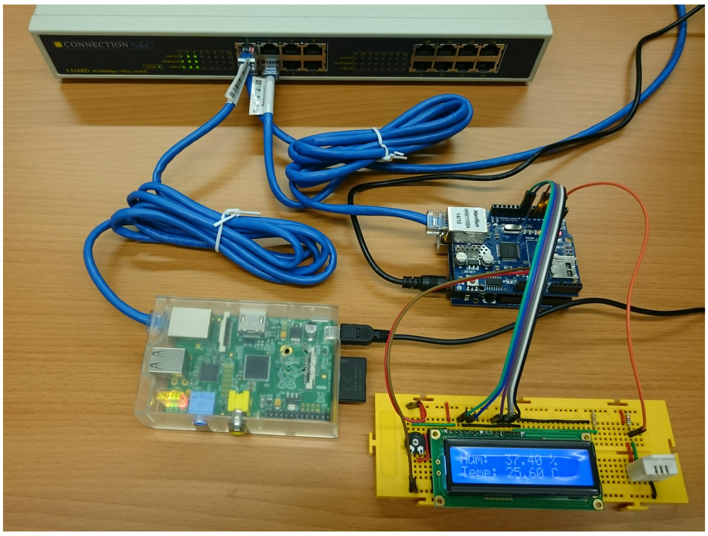

Figure 4. Prototype implementation using Arduino and Raspberry Pi boards.

The specific entities implemented are those described in section 2.2, which were identified as the main entities needed for the access control process.

The resource server broker has been implemented using two servers connected internally, as the goal is to offer both the broker functionalities of the publish/subscribe MQTT protocol and the functionalities related with the access control resource server in a single module. The first functionalities are provided by an MQTT Mosquitto broker [46], while the latter has been implemented through a Mosquitto plugin intended for the customization of authorization processes (mosquito_pyauth [47]). Additionally, we have included a custom HTTP API and interface in this server to be able to access it through the client. This is implemented using the Django Framework [48]. The client itself has also been implemented using the same development technology.

The authorization server is also written in Django, including the authorization and protection APIs of UMA. These APIs are protected by OAuth through a specific package for that framework, the Django-OAuth-Toolkit [49].

Finally, the devices implement an MQTT client for the communications with the system, and a user interface for their configuration. We have used Arduino Uno R3 boards [50] equipped with an Ethernet Shield [51] and we have connected to them a temperature and humidity sensor (AM2302/DHT22 [52]). We have also included an LCD display to show the messages received in the device (HD44780 [53]).

4.2. Deployment scenarios

We have defined three different scenarios to deploy the entities to evaluate cases in which the systems are deployed in internal servers or using external providers. Table 2 shows the deployment platform of each implemented entity for each one. In Scenario 1 (S1) all entities are deployed locally and using resource-constrained devices for both the IoT device and the resource server broker. In Scenario 2 (S2), the authorization server is deployed in the Cloud using AWS, and in Scenario 3 (S3), both the resource server broker and authorization server are located in the Cloud. The Client has been deployed in a local server in all cases.

Table 2. Testing scenarios deployment platforms for each entity

	S1	**S2**	**S3**
IoT device	Arduino	Arduino	Arduino
Resource Server Broker	Raspberry Pi	Raspberry Pi	AWS
Authorization server	Local Server	AWS	AWS
Network	LAN	WAN	WAN

4.3. Implementation results

We have defined a test to measure the time delay in the scenarios defined in section 4.2. All the tests depend on an initial state of the system, in which all the devices are already powered up and stable. Also, the needed RPTs for subscription and publishing are already stored in the device and validated, and there has already been a subscription to the queue used in the test.

The actual test consists of repeating the publication of an MQTT message to a specific queue, and its reception in a device subscribed to the queue. This is performed using the broker as an echo server: The IoT device publishes a message and receives it back through a queue to which it has been subscribed.

For each message, we obtain a timestamp when the publishing action is carried out by the device, and then again when the device receives the message back through the subscription queue. In parallel, we measure the time spent by the permission validation request which is carried out by the resource server broker to the authorization server.

The test has been carried out with and without using the access control mechanism, in order to obtain a reference time spent in each scenario for the message transmission. Using this reference time and the measurements with access control active, it is possible to determine the amount of time specifically spent in processing each message in the broker.

According to [24], the time measurement can be decomposed using the following equations:

$$T_{T\_NoAuth} = T_{P\_IoTD} + T_{Tx\_Net} + T_{P\_RSB} \tag{1}$$

$$T_{T\_Auth} = T_{P\_IoTD} + T_{Tx\_Net} + T_{P\_RSB} + T_{P\_PyAuth} + 2 * T_{P\_Intros} \tag{2}$$

In equation 1, the complete time spent by a message from its transmission to its reception without using the access control mechanism (T_{T_NoAuth}) is calculated by adding the time required by the device to process the message (T_{P_IoTD}), the time spent by the message when being transmitted through the network (T_{Tx_Net}), and the time required by the resource server broker to process the message and send it back (T_{P_RSB}). If the access control is activated, the total time required (T_{T_Auth}) would be calculated by adding the previous values and the specific time overhead produced by that module. This time can be also decomposed in the time required for processing the access control in the authorization module (T_{P_PyAuth}) and the time spent in validating the permissions in the authorization server through an introspection endpoint (T_{P_Intros}). Given that there are two actions carried out for each message (publishing and subscription), this last value must be added twice. This sum is expressed in equation 2.

From these equations, it is possible to determine the time overhead added by the information protection mechanism (T_{P_PyAuth}), substituting the expression of equation 1 for its result in 2 as shown in equation 3, and then isolating the T_{P_PyAuth} variable as shown in equation 4.

$$T_{T\_Auth} = T_{T\_NoAuth} + T_{P\_PyAuth} + 2 * T_{P\_Intros} \tag{3}$$

$$T_{P\_PyAuth} = T_{T\_Auth} - (T_{T\_NoAuth} + 2 * T_{P\_Intros}) \tag{4}$$

We have repeated the test 500 times to obtain significant results and we have calculated the mean value and standard deviation of each measured time in each scenario. The results are shown in Table 3.

Table 3. Mean delay times measured by sending and receiving messages in an IoT device, with and without the access control mechanism (milliseconds).

	S1		**S2**		**S3**	
	μ	σ	μ	σ	μ	σ
T_{T_NoAuth}	4.88	0.11	4.88	0.11	41.95	0.32
T_{T_Auth}	240.68	9.18	382.82	7.05	68.69	6.96
T_{P_Intros}	105.11	4.82	177.51	10.20	13.10	3.70
T_{P_PyAuth}	25.58	-	22.92	-	0.54	-

Analyzing the obtained values, it is possible to establish two main conclusions from these results:

1. The processing time of the authorization module is dependent on the platform where it is deployed. T_{P_PyAuth} has similar values in scenarios S1 and S2, where the resource server broker is hosted in a Raspberry Pi board. This time is reduced in more than a 95% when the RS Broker is hosted in a Cloud environment (AWS).

2. The T_{P_Intros} represents a considerable part of the total time when the test is performed activating the access control system. Moreover, this time fluctuates depending on the deployment scenario (due to the use of different networks and having different authorization server processing times).

Following the results obtained using the prototype implementation, it is proven the viability of the access control proposal in scenarios like the one described in this work. The results show that it is possible for Arduino-based and other resource-constrained devices to provide a high granularity in data access protection when combined with the adaptation of a schema originally created for web-based environments, which enables a user-centered access control scheme for pervasive computing environments and smart spaces using a policy-based mechanism similar to Internet services.

By analyzing the obtained delay measurements in the message exchange, it is shown that most part of the resource accessing process time it is consumed by the transmission of messages between the different entities of the system. In the S1 and S2 scenarios, where the IoT device and the resource server broker are deployed in a local network, T_P_Intros represents more than 40% of the total time. However, in a scenario such as S3, where the entities are distributed in a Cloud environment, the value of the T_P_Intros is reduced to 19% of the total time. The same happens with the processing times of the authorization module, which are significantly reduced when using high-resource devices (*i.e.* Raspberry Pi vs. AWS server).

5. Discussion

In this work, we have shown a possible real scenario in which applying new IoT-based technologies would allow to increase the comfort of employees during their work hours. We have identified in this complex smart space scenario the main information flows which should be defined and enabled for the system to work properly, and we have proposed a custom access control to protect personal information used by the system. In order to determine the viability of the solution, we have developed a prototype IoT device which is able to interact with sensors and actuators in a smart space, and then we have deployed three testing scenarios (local, Cloud and hybrid scenarios).

The results about time overhead produced by using the access control scheme prove the viability of the system in ways. First, the total time needed for a full communication cycle in the system (that is, sending and receiving a message using a publish/subscribe protocol), is in a milliseconds degree, which is a more than acceptable value for this environment, as the environmental changes and decision-making processes could be executed within intervals of even minutes. Additionally, the results of the tests carried out in the S3 scenario show that when the system modules are deployed and optimized for Cloud-based services, the overhead of introducing the access control system is much less important for the overall system.

This work is intended to be expanded in future research by defining the specific access control policies to protect the personal and environmental information shared in these smart space scenarios.

References

[1] O. Vermesan and P. Friess, *Internet of things: converging technologies for smart environments and integrated ecosystems*. River Publishers, 2013.

[2] J. Yun, I.-Y. Ahn, S.-C. Choi, and J. Kim, "Tteo (things talk to each other): Programming smart spaces based on iot systems," *Sensors (Basel, Switzerland)*, vol. 16, p. 467, Apr 2016.

[3] S. Yamamoto, N. Kouyama, K. Yasumoto, and M. Ito, "Maximizing users comfort levels through user preference estimation in public smartspaces," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pp. 572–577, IEEE, 2011.

[4] M. Janneck, S. Jent, P. Weber, and H. Nissen, "Ergonomics to go: Designing the mobile workspace," *International Journal of Human-Computer Interaction*, vol. 34, pp. 1052–1062, Nov 2018.

[5] B. Mamo, "The use of context aware pervasive systems in the area of human factors and ergonomics," in *International Conference on Applied Human Factors and Ergonomics*, pp. 43–52, Springer, 2017.

[6] M. Vega-Barbas, D. Casado-Mansilla, M. A. Valero, D. López-de Ipina, J. Bravo, and F. Flórez, "Smart spaces and smart objects interoperability architecture (s3oia)," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pp. 725–730, IEEE, 2012.

[7] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "Rfid technology for iot-based personal healthcare in smart spaces," *IEEE Internet of things journal*, vol. 1, no. 2, pp. 144–152, 2014.

[8] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 134–143, 2010.

[9] E. Laftchiev and D. Nikovski, "An iot system to estimate personal thermal comfort," pp. 672–677, IEEE, Dec 2016.

[10] H. Yang and Y. Pan, "Elements and structure of the smart lighting design in the office," *Journal of Ergonomics Society of Korea*, vol. 35, pp. 29–38, Feb 2016.

[11] A. Abuarqoub, H. Abusaimeh, M. Hammoudeh, D. Uliyan, M. Abu-Hashem, S. Murad, M. Al-Jarrah, and F. Al-Fayez, "A survey on internet of things enabled smart campus applications," ICFNDS '17, pp. 1–7, ACM, Jul 19, 2017.

[12] A. C. Cordero, U. Rahe, H. Wallbaum, Q. Jin, and M. Forooraghi, "Smart and sustainable offices (sso): Showcasing a holistic approach to realise the next generation offices," *Informes de la Construcción*, vol. 69, p. e221, Jan 1, 2017.

[13] D. G. Korzun, A. M. Kashevnik, S. I. Balandin, and A. V. Smirnov, "The smart-m3 platform: Experience of smart space application development for internet of things," in *Conference on Smart Spaces*, pp. 56–67, Springer, 2015.

[14] D. Levonevskiy, I. Vatamaniuk, and A. Saveliev, "Processing models for conflicting user requests in ubiquitous corporate smart spaces," *MATEC Web of Conferences*, vol. 161, p. 3006, 2018.

[15] M. C. Schall, R. F. Sesek, and L. A. Cavuoto, "Barriers to the adoption of wearable sensors in the workplace: A survey of occupational safety and health professionals," *Human Factors: The Journal of Human Factors and Ergonomics Society*, vol. 60, pp. 351–362, May 2018.

[16] Y. Sahni, J. Cao, and J. Shen, *Challenges and Opportunities in Designing Smart Spaces*, pp. 131–152. Internet of Everything, Springer, 2018.

[17] P. A. Nixon, W. Wagealla, C. English, and S. Terzis, *Security, Privacy and Trust Issues in Smart Environments*, pp. 249–270. Smart Environments, Hoboken, NJ, USA: John Wiley & Sons, Inc, Jan 28, 2005.

[18] J. Kim and R. Dear, "Workspace satisfaction: The privacy-communication trade-off in open-plan offices," *Journal of Environmental Psychology*, pp. 18–26, Jan 2013.

[19] C. Röcker and A. Feith, "Revisiting privacy in smart spaces: Social and architectural aspects of privacy in technology-enhanced environments," in *Proceedings of the International Symposium on Computing, Communication and Control*, pp. 201–205, 2009.

[20] J. S. Kumar and D. R. Patel, "A survey on internet of things: Security and privacy issues," *International Journal of Computer Applications*, vol. 90, no. 11, 2014.

[21] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237–262, 2017.

[22] W. Oyomno, P. Jäppinen, and E. Kerttula, "Privacy preservation for personalised services in smart spaces," in *Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future*, pp. 181–189, IEEE, 2011.

[23] I. Chong, A. Xiong, and R. W. Proctor, “Human factors in the privacy and security of the internet of things,” *Ergonomics in Design: The Quarterly of Human Factors Applications*, p. 106480461775032, May 2, 2018.

[24] L. Cruz-Piris, D. Rivera, I. Marsa-Maestre, E. de la Hoz, and J. R. Velasco, “Access control mechanism for iot environments based on modelling communication procedures as resources,” *Sensors*, vol. 18, no. 3, p. 917, 2018.

[25] E. Maler, D. Catalano, M. Machulak, and T. Hardjono, “User-managed access (uma) profile of oauth 2.0,” *Kantara Initiative*, 2016.

[26] P. Ray, “A survey on internet of things architectures,” *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291 – 319, 2018.

[27] P. Sethi and S. R. Sarangi, “Internet of things: architectures, protocols, and applications,” *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[28] M. Laine, “Restful web services for the internet of things,” *[Online] Saatavilla: http://media Tkk Fi/webservices/personnel/markku_laine/restful_web_services_for _the_internet_of_things Pdf*, 2012.

[29] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, “Publish/subscribe-enabled software defined networking for efficient and scalable iot communications,” *IEEE communications magazine*, vol. 53, no. 9, pp. 48–54, 2015.

[30] Z. Shelby, K. Hartke, and C. Bormann, “The constrained application protocol (coap),” tech. rep., 2014.

[31] G. Pardo-Castellote, “Omg data-distribution service: Architectural overview,” in *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pp. 200–206, IEEE, 2003.

[32] S. Vinoski, “Advanced message queuing protocol,” *IEEE Internet Computing*, vol. 10, no. 6, 2006.

[33] S. Katsikeas, K. Fysarakis, A. Miaoudakis, A. Van Bemten, I. Askoxylakis, I. Papaefstathiou, and A. Plemenos, “Lightweight & secure industrial iot communications via the mq telemetry transport protocol,” in *Computers and Communications (ISCC), 2017 IEEE Symposium on*, pp. 1193–1200, IEEE, 2017.

[34] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[35] E. Yuan and J. Tong, “Attributed based access control (abac) for web services,” in *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*, IEEE, 2005.

[36] 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*, vol. 58, no. 5-6, pp. 1189–1205, 2013.

[37] G. Zhang and J. Tian, “An extended role based access control model for the internet of things,” in *Information Networking and Automation (ICINA), 2010 International Conference on*, vol. 1, pp. V1–319, IEEE, 2010.

[38] A. Anderson, A. Nadalin, B. Parducci, D. Engovatov, H. Lockhart, M. Kudo, P. Humenn, S. Godik, S. Anderson, S. Crocker, *et al.*, “extensible access control markup language (xacml) version 1.0,” *OASIS*, 2003.

[39] D. Hardt, “The oauth 2.0 authorization framework,” tech. rep., 2012.

[40] D. Rivera, L. Cruz-Piris, G. Lopez-Civera, E. de la Hoz, and I. Marsa-Maestre, “Applying an unified access control for iot-based intelligent agent systems,” in *Service-Oriented Computing and Applications (SOCA), 2015 IEEE 8th International Conference on*, pp. 247–251, IEEE, 2015.

[41] T. J. Van Der Voordt, “Productivity and employee satisfaction in flexible workplaces,” *Journal of Corporate Real Estate*, vol. 6, no. 2, pp. 133–148, 2004.

[42] B. P. Haynes, “The impact of office comfort on productivity,” *Journal of Facilities Management*, vol. 6, no. 1, pp. 37–51, 2008.

[43] “Amazon web services.” `http://aws.amazon.com/es/`. Accessed: 2019-02-19.

[44] “Arduino board.” `http://www.arduino.cc/`. Accessed: 2019-02-19.

[45] “Raspberry pi board.” `https://www.raspberrypi.org/`. Accessed: 2019-02-19.

[46] “Mosquitto mqtt broker.” `http://mosquitto.org/`. Accessed: 2019-02-19.

[47] M. Bachry, “Mosquitto pyauth plugin.” `https://github.com/mbachry/mosquitto_pyauth`. Accessed: 2019-02-19.

[48] “Django web framework.” `https://www.djangoproject.com/`. Accessed: 2019-02-19.

[49] Evonove, “Django-oauth-toolkit.” `https://github.com/evonove/django-oauth-toolkit`. Accessed: 2019-02-19.

[50] “Arduino uno r3 board.” `http://www.arduino.cc/en/Main/ArduinoBoardUno`. Accessed: 2019-02-19.

[51] N. O'leary, "Mqtt client arduino library." `https://pubsubclient.knolleary.net/`. Accessed: 2019-02-19.

[52] "Digital-output relative humidity & temperature sensor/module datasheet." `https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf`. Accessed: 2019-02-19.

[53] "Hitachi dot matrix liquid crystal display controller/driver datasheet." `https://www.sparkfun.com/datasheets/LCD/HD44780.pdf`. Accessed: 2019-02-19.