RIVERA, D., CRUZ-PIRIS, L., LOPEZ-CIVERA, G., DE LA HOZ, E. and MARSAMAESTRE, I., 2015, "Applying an Unified Access Control for IoT-based Intelligent Agent Systems, 2015 IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA), pp. 247-251.

Available at http://dx.doi.org/10.1109/SOCA.2015.40

Applying an unified access control for IoT-based Intelligent Agent Systems

Diego Rivera*, Luis Cruz-Piris†,German Lopez-Civera‡, Enrique de la Hoz§ and Ivan Marsa-Maestre¶
Computer Engineering Department, Universidad de Alcalá
Alcalá de Henares, Spain
Email: {*diego.rivera, †luis.cruz, ‡g.lopez, §enrique.delahoz, ¶ivan.marsa}@uah.es

***Abstract*—The rise of the Internet of Things (IoT) paradigm has allowed the design and development of new services interconnecting heterogeneous devices. However, the complexity of these new systems hasn't been followed by the increase of intelligence and reasoning of the devices connected. On the other hand, intelligent agent systems have developed precisely these characteristics so the combination of both paradigms by modelling intelligent agents in IoT devices is a very promising approach that will enable a more powerful and smart IoT. The interconnection of agents through a Internet-based network implies addressing critical issues that affect all network communications, such as security, privacy and access control, specially given the sensitivity of the information exchanged by agents. In this paper, we propose the application of User-Managed Access (UMA) to provide an unified access control schema for an heterogeneous hybrid architecture of IoT devices and intelligent agents.**

I. Introduction

In the last years, the Internet of Things (IoT) [1] has been established as an evolution of the Internet where every object is interconnected. This has lead to complex systems composed of many heterogeneous devices that allow the creation of novel services related with diverse topics such as energy management, smart buildings, etc.

These services rise new challenges due to its heterogeneous character and its complexity. Traditionally, the devices that compose IoT environments lack intelligence [2], meaning that they are not able to make autonomous decisions and change their behaviour proactively. The ability of IoT devices to sense and process data from their environment gives an ideal scenario for multi-agent systems to be applied.

This type of systems allow to model complex problems as sets of autonomous intelligent agents that are able to adapt their behaviour depending on the interactions between them [3].

Including these new features in IoT devices allow them to reason and negotiate with the IoT environment to achieve a global goal that satisfies the requirements and preferences of each single device. The possibility of the devices to act autonomously decreases the complexity of the management of the IoT system and increases its flexibility to react to the external events affecting it. Since the birth of the IoT concept, there have been several works that propose the combination of intelligent agents and IoT devices [4], and the interest on the topic has continued to our days [5] [6] [7].

The distribution of agents in a Internet-based network implies that the challenges related with Internet communications have to be addressed exactly as in every other Internet scenario, specially given that the data exchanged by the agents can be sensible and that the negotiation operations should be protected by an access control system.

The communications between agents must rely on Internet-based protocols. These communications can follow many different schemes, dependant on the specific scenario where the network is deployed. Among these schemes, it is possible to highlight those based on HTTP client/server architectures (such as RESTful services) and those based on lightweight message exchange protocols through a centralized manager server in charge of coordinating the communications (MQTT, CoAP, AMQP, etc.). The message exchange schemes are closer to the usual communication schemes between software agents.

There are many access controls schemes defined for Internet services. Nowadays, OAuth 2.0 [8] is a *de facto* standard to solve the situation of allowing a third party to access data of our own, while offering a high level of granularity and ease of use. On top of OAuth 2.0, a profile called User-Managed Access (UMA) has been defined [9].

In this paper, we propose using this profile to implement access control functionalities in multi-agent IoT-based systems.

The paper is organized as follows. Section II provides an overview of related agent-based IoT systems and a brief description of the access control scheme that we propose to use. In section III we study the main security and privacy considerations for these systems, and describe the utilization of the UMA scheme in them. Section IV describes an example scenario of application and finally, conclusions and future work is determined in the last section.

II. Related work

A. Intelligent Agents in IoT

As it has been pointed out in section I, there is an important base of work in the field of agents interacting through a network in a IoT environment.

For instance, papers like [2] or [10] propose the creation of the concept "Agents of Things" as an extension of the IoT paradigm where devices are provided with intelligence and reasoning capabilities. They propose a hierarchical layered architecture based on the typical IoT architecture, but adding specific layers to deal with the agent specific issue. A middleware layer is defined to communicate the lower layers (network communications and sensoring) with the reasoning and application software.

In [11], a platform based on IoT Smart Objects that uses agents and Cloud Computing is presented, and also defines an architecture for it. The architecture of this proposal consists on defining a set of smart object agents embedded in a *cyberphysical environment* (the physical and logical context for the agents) and a Cloud Computing platform on top of them allowing the creation of virtual smart objects.

Another similar architecture, based on Web-based schemes and protocols is shown in [7]. In this case, the architecture uses a gateway layer to convert and aggregate IoT data and a Cloud database to manage it. Agents communicate through standard Internet protocols, through HTTP REST interfaces. REST-based communications are also used in [12] to develop agents in a Virtual Machine environment.

Multi-agent systems are also been used to develop Smart Cities (highly based on IoT) as it is shown in [13]. In this specific context, agents are used to improve the development of smart efficient energy grids. Agents are designed to collaborate in order to pursue a common goal (keeping the network operational and satisfy the highest possible quality standards).

Other proposal related with Smart Cities can be found in [14], where distributed agents are used in a IoT system to control traffic-light intersections. The system uses a self-organization approach to ensure the optimization of traffic-light in crossroads, using environmental data from IoT-like devices. Although this architecture defines local clusters of agents, the system is connected to the Internet to interconnect the local systems.

B. User-Managed Access (UMA)

User-Managed Access (UMA) is a profile of OAuth 2.0. designed to allow users to authorize arbitrary third parties to access their resources (personal data, content, services, etc.) defining who and what can get access to them.

The protocol specifications are being developed by a working group of the Kantara Initiative [15] with the intent to contribute the draft work to the Internet Engineering Task Force (IETF).

The UMA IETF draft defines the following basic entities participating in the UMA schema:

- **Resource Owner (RO)**: An entity capable of granting access to a protected resource as defined in OAuth. The RO is typically an end-user.
- **Requesting Party (RP)**: end-user that uses a Client to seek access to a protected resource.
- **Client**: Application that allows RPs to request protected resources with the RO authorization.
- **Resource Set (RS)**: One or more protected resources that a Resource Server manages as a set, in an abstract manner [16]. In this context, this means that the whole set is governed by the same access policies, as defined in the Authorization Server.
- **Resource Server**: Application that allows the ROs to manage the protection of the RSs through the Authorization Server.
- **Authorization Server (AS)**: Server that issues authorization data and permission tokens to a Client and protects RSs managed at a Resource Server. It offers two OAuth protected APIs (Authorization API and Protection API) and a RS management interface for ROs.

The UMA process relies on the use of three tokens. Two of them are OAuth Access Tokens: Authorization API Token (AAT) and Protection API Token (PAT). These two tokens are used to protect the access to the AS APIs. The other one is the Requesting Party Token (RPT) and it is used by a RP Client when accessing protected resources.

The UMA protocol has three phases:

1) **Protect a Resource**: The RO, which has a RS in a Resource Server, registers the RS into the AS using the Protection API (using a previously obtained valid PAT). Out of band, the RO configures the access policies associated with the RS scopes in the RS management interface (a scope represents a specific action to perform over the RS).
2) **Get authorization**: The client approaches the resource server seeking access to a protected resource. The Client, on behalf of a RP, must first obtain authorization data and a RPT through the Authorization API of the AS, in order to successfully access the protected resource.
3) **Access a Resource**: Once the Client has a valid RPT with sufficient authorization data associated with it, it will be able to access the protected resource.

III. Studying agent protection in IoT environments

A. Security and privacy considerations

Despite of the usage of intelligent agents in a wide variety of fields, the protection of the information that they exchange has not been so well studied. As in many applications all the agents that compose the system are deployed in a highly controlled environment (e.g. internal memory of a computer or a local network), security does not seem a critical problem. However, intelligent agents have been found useful in distributed environments where the information travels through not so secure channels.

In an IoT environment agents can also represent its owner preferences and personal information. This can lead to privacy issues like information leakage or user profiling that have to be addressed. For example, in a air conditioning coordination system, a user specifies its preferences to the agent that manages the cooling of its home/workplace. That preferences can also represent the daily schedule of their owner and could lead to a privacy abuse if they are not properly protected.

The security methods applied must minimize the exposure to information processing and collection. As it can be seen in the proposal described in [17] the protection of agent communication must cover all aspects of security, including confidentiality, integrity, authentication and access control:

- Confidentiality and Integrity can be addressed by using well-established protocols that already implements

TLS [18], we can protect the communications from eavesdropping and ensure it is not altered by third parties.

- Authentication can also be managed by TLS if a PKI (Public Key Infrastructure) is deployed and each agent has its own certificate. An alternative approach is to delegate the authentication to higher layers where the owner can authenticate himself and send to the agent some kind of token that identifies it in the system.

On top of these two aspects Access Control is not so well studied. In a multi-agent system different levels of roles can be performed between agents, therefore different permission levels must be considered. Furthermore, as agents can act on behalf of its owner, they will inherit their privacy concerns and must be able to choose which agents they want to communicate with.

In the next subsection we describe a interoperable system based on UMA enabling an unified access control that includes agent communication.

B. An access control schema for IoT agents

In this section we are going to describe the flows of the access control scheme defined in subsection II-B applied to IoT multi-agent systems. In figure 1 it is shown a schema where the communication is modelled using a centralized server to coordinate and control the message exchange.

The Message exchange server of this system will be modelled as a Resource Server as it was defined in the previous section. This implies that all operations between agents will be seen as Resource Sets. The message exchange service can be modelled as a queue system (i.e. MQTT or AMQP protocols). In that case, the Resources to protect will be each queue in the server.

Each queue will represent a communication channel between agents (one-to-one, one-to-many, etc.). Using each one of these queues, the agents are able to perform the specific operations needed to achieve their goals.

IoT devices will contain intelligent agents in combination with other IoT-related functionalities. A Client is also embedded in the device in order to manage requests to use the Message Exchange service.

The Authorization Server will be deployed as an independent server accessible by all the other entities in the schema.

As UMA profile describes, the protection and access of a Resource will be performed in three phases:

1) The Agent/IoT Device Owner will create a communication queue in the Message Exchange service and will register it within the Authorization Server, configuring the permissions for users or group of users.
2) Once the queue is protected, the authorization flow is started. Another (or the same) Agent/IoT Device Owner (a user) will use a Web Client to interact with the Resource Server and the Authorization Server in order to obtain the permission required for its agent to access to the registered operation. This phase ends with the configuration of the Agent/IoT Device by the owner, giving to the device a token representing the acquired permissions.
3) In the third phase, the properly configured Agent/IoT Devices will perform the operations required to perform autonomously their tasks (i.e. negotiate for a common goal) through the registered queue.

After the process is complete, all the operations will be protected by the access control rules configured in the Authorization Server.

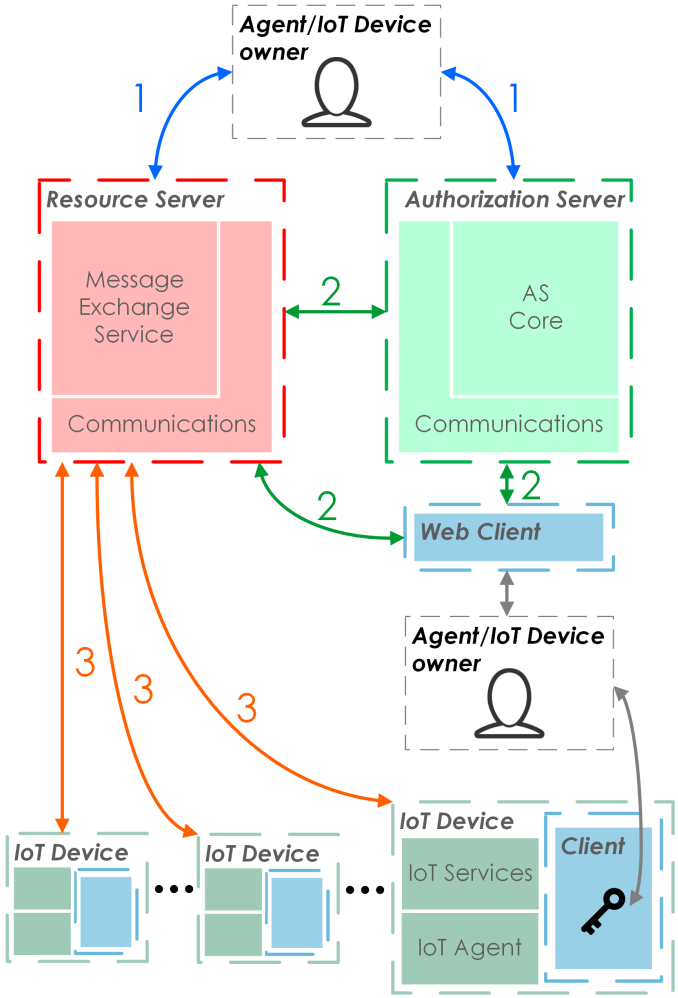

Fig. 1: UMA-based access control schema for IoT agents

IV. Example: Hybrid system with intelligent agent and IoT devices

In this section we propose an example scenario that combines IoT devices, software intelligent agents and new hybrid elements as it is shown in figure 2. The access control needs of every element will be addressed using the same authorization scheme based in UMA.

A. Description

The existence of a set of IoT devices containing sensors (traffic state, weather sensors, etc.) deployed distributed

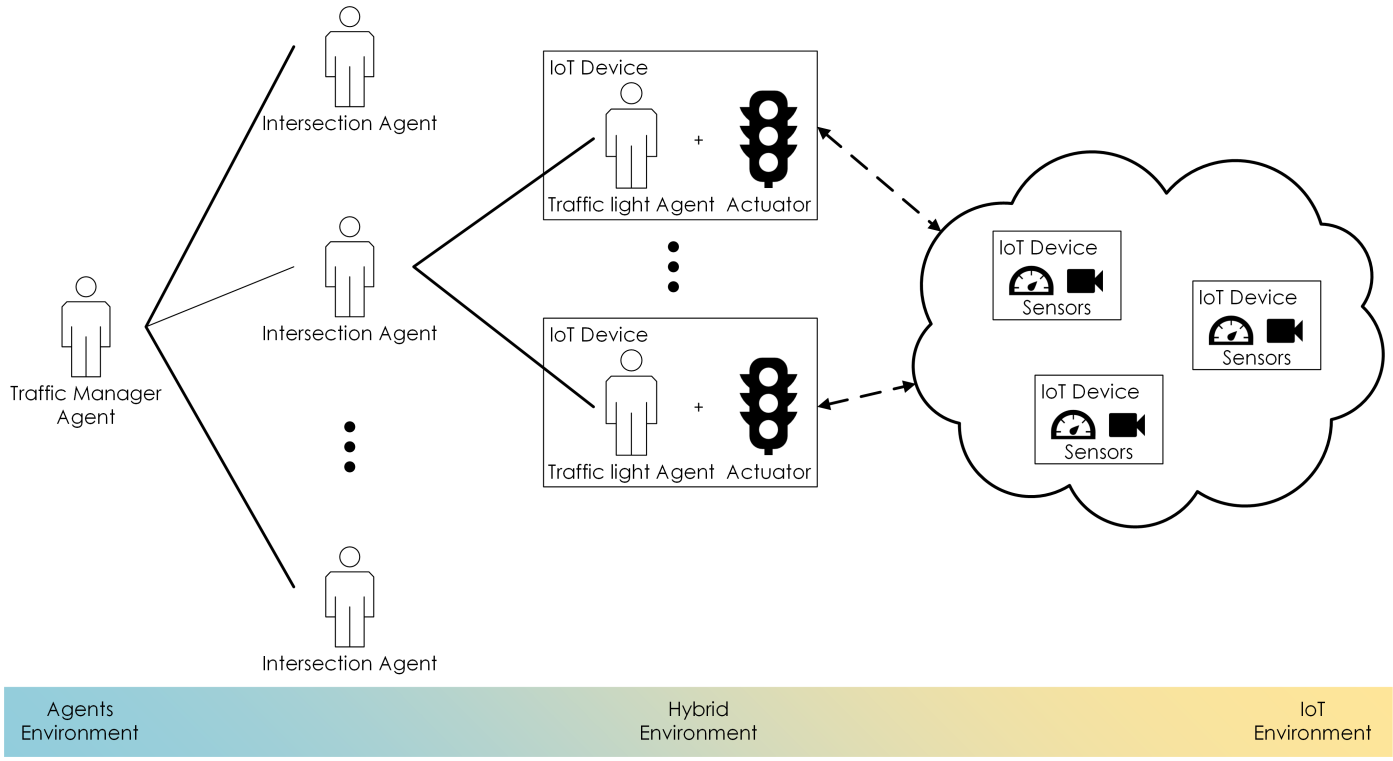

Fig. 2: Example of hybrid system with intelligent agent and IoT devices

through a city and giving real time information is assumed for the example purposes. Each traffic light has also an IoT device composed of an actuator that is able to change the light shown and an intelligent agent making the behavioral decisions.

Each "Traffic Light Agent" starts a negotiation process with the rest of "Traffic Light Agents" from their intersection after retrieving the information offered by the sensors deployed in the nearby roads. This negotiation process is performed through the correspondent "Intersection Agent". This hierarchically superior agent is in charge of taking decisions if the negotiation process fails. Finally, decisions taken by each "Intersection Agent" are executed after a new negotiation process with the rest of the "Intersection Agents". If the agents are not able to agree, the agent with highest hierarchical range ("Traffic Manager Agent") will enforce a solution taking into account the global preferences of the system.

B. *Security considerations*

The IoT devices (sensors and agents/actuators) are exposed directly to the Internet. "Intersection Agents" and the "Traffic manager Agent" can be integrated in a software platform that is accessed through Internet via a gateway. It is important to assure that other external elements are not able to interfere in the negotiation process or provide malicious information to the system. Consequently, only authorized sensors must be able to communicate information periodically to each "Traffic Light Agent", and the communication hierarchy between agents must be guaranteed.

The rule definition to achieve these goals is a complex task. Given that all entities in the system should meet the same security restrictions, using a unified authorization service implies a simplified management of the process.

C. *Communications through a Message Exchange Service*

Using a basic publication/subscription scheme managed through queues (typical in message exchange protocols such as MQTT or AMQP), each system entity will own a token that will allow it to perform actions (publish, subscribe or both) associated with a specific queue.

As an example, it is possible to define a queue named `/system/intersection_negotiation`. In this queue, every "Intersection Agent" will be able to do publish and subscribe actions in order to perform the negotiation process. The "Traffic Manager Agent" will also be allowed to do the same actions. The Message Exchange server, when receiving a message through this queue, will verify against the Authorization Server the permissions (depending on who sends it, the action performed and the queue used). If the verification is successful, the message will be processed, and will be discarded in other case.

V. Conclusion and Future Work

In this paper we have shown a proposal to apply a schema that unifies access control systems between intelligent agents, IoT devices and hybrid elements. This schema tries to seamlessly apply access control policies independently of the nature of the entities that interact in an IoT environment. This allows an easy migration of the policy concept to agent communication and the management of different roles and permissions in the decision-making process during agent negotiation. This

implies an improvement in the access control management system.

This proposal represents an alternative approach to solve security issues that have emerged by using distributed multi-agent systems in IoT. The protection of these systems is critical not only because of privacy issues, but also because of the threat that an attack represents over infrastructures that manage services like urban transport or energy delivery. With this approach we can employ Internet to interconnect agents and devices while the security is not reduced and the environment can still take advantage of intelligent agents.

As future work we consider the evaluation of the impact in the performance of integrating this schema in a distributed multi-agent system. This way we can evaluate its behavior and identify use cases where it can be deployed.

ACKNOWLEDGMENT

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness through the grant IPT-2012-0839-430000 (project SITAC) and the grant TIN2013-47803-C2-2-R (project EDUCERE).

REFERENCES

[1] K. Ashton, "That "internet of things" thing," *RFiD Journal*, vol. 22, no. 7, pp. 97–114, 2009. [Online]. Available: http://www.rfidjournal.com/articles/view?4986

[2] A. M. Mzahm, M. S. Ahmad, and A. Y. Tang, "Enhancing the internet of things (iot) via the concept of agent of things (aot)," *Journal of Network and Innovative Computing*, vol. 2, no. 2014, pp. 101–110.

[3] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[4] N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes, "Hive: Distributed agents for networking things," in *Agent Systems and Applications, 1999 and Third International Symposium on Mobile Agents. Proceedings. First International Symposium on*. IEEE, 1999, pp. 118–129.

[5] A. M. Mzahm, M. S. Ahmad, and A. Y. Tang, "Computing hardware analysis for agents of things (aot) applications," in *Information Technology and Multimedia (ICIMU), 2014 International Conference on*. IEEE, 2014, pp. 223–228.

[6] H. Yu, Z. Shen, and C. Leung, "From internet of things to internet of agents," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. IEEE, 2013, pp. 1054–1057.

[7] P. Leong and L. Lu, "Multiagent web for the internet of things," in *Information Science and Applications (ICISA), 2014 International Conference on*. IEEE, 2014, pp. 1–4.

[8] D. Hardt, "The oauth 2.0 authorization framework," Internet Engineering Task Force (IETF), RFC, 10 2012. [Online]. Available: http://tools.ietf.org/html/rfc6749

[9] T. Hardjono, E. Maler, M. Machulak, and D. Catalano, "User-managed access (uma) profile of oauth 2.0," Kantara Initiative, Recommendation, 04 2014. [Online]. Available: https://docs.kantarainitiative.org/uma/rec-uma-core-v1_0.html

[10] A. M. Mzahm, M. S. Ahmad, and A. Y. Tang, "Agents of things (aot): An intelligent operational concept of the internet of things (iot)," in *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*. IEEE, 2013, pp. 159–164.

[11] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Integration of agent-based and cloud computing for the smart objects-oriented iot," in *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*. IEEE, 2014, pp. 493–498.

[12] A. Gouaïch and M. Bergeret, "Rest-a: An agent virtual machine based on rest framework," in *Advances in Practical Applications of Agents and Multiagent Systems*. Springer, 2010, pp. 103–112.

[13] M. Roscia, M. Longo, and G. C. Lazaroiu, "Smart city by multi-agent systems," in *Renewable Energy Research and Applications (ICRERA), 2013 International Conference on*. IEEE, 2013, pp. 371–376.

[14] C. E. Turcu, V. G. Gaitan, and C. O. Turcu, "An internet of things-based distributed intelligent system with self-optimization for controlling traffic-light intersections," in *Applied and Theoretical Electricity (ICATE), 2012 International Conference on*. IEEE, 2012, pp. 1–5.

[15] "Kantara initiative." [Online]. Available: https://kantarainitiative.org/

[16] T. Hardjono, E. Male, M. Machulak, and D. Catalano, "Oauth 2.0 resource set registration," Kantara Initiative, Recommendation, 04 2015. [Online]. Available: https://docs.kantarainitiative.org/uma/rec-oauth-resource-reg.html

[17] J. M. Such, A. GarcíA-Fornes, A. Espinosa, and J. Bellver, "Magentix2: A privacy-enhancing agent platform," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 96–109, 2013.

[18] T. Dierks, "The transport layer security (tls) protocol version 1.2," The Internet Engineering Task Force, RFC 5246, 08 2008. [Online]. Available: https://tools.ietf.org/rfc/rfc5246.txt