UNIVERSIDAD DE MURCIA

FACULTAD DE INFORMÁTICA

A Proposal for User's Privacy Management in Context-Aware Systems

Propuesta para la Gestión de la Privacidad de los Usuarios en Sistemas Sensibles al Contexto

Author

Alberto Huertas Celdrán

Thesis advisors

Dr. Félix Jesús García Clemente

Dr. Gregorio Martínez Pérez

Dr. Manuel Gil Pérez

Murcia, February 2017

Abstract

The evolution of technology and computing paradigms has increased the amount of heterogeneous and sensitive information handled by Information Management Systems (IMS). This fact has influenced the emergence of new challenges that, in certain cases, still require important research and development efforts to be solved. Currently, IMSs should ideally be able to handle and protect personal and contextual information in order to incorporate them to their management processes. Otherwise, sensitive pieces of information, such as the users' locations, identities, activities, and contextual information could be obtained by malicious users or service providers and misuse it. IMSs should allow users to control what contextual and personal information they want to reveal, where and when the information will be exchanged, and to whom. Semantic web techniques have been found as a promising way to accomplish the management and protection of the previous pieces of information in IMSs.

The main objective of this PhD Thesis is the definition, design, and deployment of context-aware solutions that allow protecting sensitive information as well as controlling the behavior of the system resources. In order to reach this objective, we will start studying and analyzing the evolution of information management systems from location-based solutions to context-aware systems, as well as how web semantic techniques can help in managing and protecting the information considered by these systems. Subsequently, we plan to propose location-based and context-aware systems in charge of exchanging and protecting the users' information in intra- and inter-context scenarios by considering semantic web techniques. Finally, we also plan to propose location-based and context-aware systems to manage automatically network resources by taking into account aspects like QoS, energy efficiency, or performance.

To reach our objectives we have proposed:

- A privacy-preserving and context-aware solution to exchange location information. This proposal allows the development of context-aware applications by considering the privacy of the users' location. Users define their privacy-policies to control where, when, how, and to whom their location information will be revealed to others. This fact allows users to control the sensitive information by avoiding to rely on third parties. Moreover, our solution has a set of predefined queries that provides information to the context-aware applications. This information is provided by taking into account the privacy-policies defined previously by users in our middleware.

- A context-aware solution to protect the users' information not only regarding users, but also services. This is designed as a recommender system capable of providing recommendations to users by considering the information they reveal to the services that provide the recommendations. Using this solution, users can control their information by hiding their identities to services, hiding their locations, and providing fake positions, among other ways to increase the privacy protection of their personal and context-aware information.

- A multi-context solution in charge of protecting the users' information when they move between independent contexts. In order to protect their information, users just have to choose the most suitable profile according to their interests in the context where they are. Furthermore, using our solution users are able to modify the profiles adding, deleting, or modifying some of the policies that form the profiles. Users can decide at real-time what, where, when, how, to whom, and at which level of precision they want to release their information to other users belonging to different contexts. This information can be the *space* in which they are located with different levels of granularity; the users' *personal information* with different levels of precision; the users' *activity*; and the information oriented to the *context* in which they are located.

- A mobility-aware policy-based system capable of reducing the energy consumption in networks oriented to the SDN paradigm. The policies defined in our solution allow the SDN paradigm to switch on/off network resources when they are consuming energy in an inefficient way, as well as create virtualized network resources like proxies to reduce the network traffic generated by users consuming services close to the network infrastructure. Network administrators can define policies that will decide the list of potential actions to be taken by the SDN components, in accordance with the energy consumption, the users' mobility, and the network statistics.

- A proposal oriented to ensure the QoS and end-user experience in dynamic scenarios of mobile networks. Our proposal is in charge of managing the SDN resources at run-time, using high-level policies. Among the different sets of policies, we emphasize here the use of mobility-aware management-oriented policies, defined by the service provider network administrator to decide the actions made by the SDN according the network infrastructure statistics and location, and the mobility of users and services. These policies are oriented to guarantee end-user experiences in very crowded places (e.g., stadiums, shopping malls, or unexpected traffic jams).

The following PhD Thesis is a compilation of the next published articles, being the PhD the main author in all of them:

1. Alberto Huertas Celdrán, Félix J. García Clemente, Manuel Gil Pérez, Gregorio Martínez Pérez. "SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications", IEEE Systems Journal, vol. 10, no. 3, pp. 1111-1124, September 2016.

2. Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez. "PRECISE: Privacy-aware recommender based on context information for Cloud service environments", IEEE Communications Magazine, vol. 52, no. 8, pp. 90-96, August 2014.

3. Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez. "What private information are you disclosing? A privacy-preserving system supervised by yourself", in CSS'14: Proceedings of the 6th International Symposium on Cyberspace Safety and Security, pp. 1221-1228, Paris (France), 20-22 August 2014.

4. Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez. "MASTERY: A multicontext-aware system that preserves the users' privacy", in NOMS'16: Proceedings of the IEEE/IFIP Network Operations and Management Symposium, pp. 523-528, Istanbul (Turkey), 25-29 April 2016.

5. Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez. "Policy-based management for green mobile networks through Software-Defined Networking", Mobile Networks and Applications, Springer Mobile Networks and Applications, Published online 05 December 2016.

6. Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez. "Enabling highly dynamic mobile scenarios with Software Defined Networking", IEEE Communications Magazine, Feature Topics Issue on SDN Use Cases for Service Provider Networks, Accepted, 2017.

Agradecimientos/Acknowledgements

En primer lugar me gustaría dar las gracias a mis padres por el cariño, educación y valores que me han transmitido desde pequeño. A mi madre, María Dolores, por todo su enorme trabajo y cariño para sacar adelante a la familia. Mil gracias. A mi padre, Rufino, por su dedicación en mi educación y por los valores de esfuerzo y compromiso que me transmitió. Aunque te fuiste muy pronto, siempre serás mi ejemplo a seguir. A mi hermana, Gloria, porque siempre ha estado a mi lado en los momentos difíciles. A toda mi familia, tíos, primos y abuelos por sus muestras de cariño y ayuda todos estos años. Mil gracias a todos.

También quiero agradecer a mis amigos los buenos momentos vividos, así como el apoyo en los momentos no tan buenos. Gracias por vuestro tiempo, ayuda y consejos.

Finalmente, también me gustaría agradecer a mis directores, Félix, Gregorio, y Manuel, la posibilidad de haber aprendido de ellos valores como la perseverancia, el esfuerzo o la superación en el trabajo; sin olvidar la parte más importante de la vida, la felicidad.

First I would like to thank my parents for the love, education and values transmitted since I was a child. To my mother, María Dolores, for all her great work and love to carry out the family. Many thanks. To my father, Rufino, for his dedication in my education and for the values of effort and commitment he transmitted to me. Although you left very soon, you will always be my model. To my sister, Gloria, because she has always been at my side in the difficult moments. To all my family, uncles, cousins and grandparents for their signs of affection and help all these years. Thank you all.

I also want to thank my friends for the good times lived, as well as their support in the not so good moments. Thanks for your time, help and advices.

Finally, I would also like to thank my directors, Félix, Gregorio, and Manuel, the possibility of learning from them values such as perseverance, effort, or improvement at work; without forgetting the most important part of life, the happiness.

Table of Contents

Chapter 1

PhD Thesis

1.1. Introduction and Motivation

The goal of this chapter consists on introducing and motivating the purpose of this PhD Thesis by showing how the evolution of technology has increased the complexity of the Information Management Systems (IMS) and how new paradigms and solutions are needed to face their increasing complexity. In particular, the increasing of heterogeneous information provided by new technologies and paradigms, the evaluation processes to make decisions and protect the privacy of sensitive pieces, and the diversity of components distributed along different organizations are one of the most important aspects that have increased the complexity in IMSs thus creating the need to design new mechanisms to address them.

The intention of this chapter is also to provide a clear vision of the objectives of this PhD Thesis, the contributions that have allowed to overcoming the detected open challenges, and a structured description of the proposals made to improve or partially solve the previously identified challenges.

The methodology of this PhD Thesis has been conducted by following a scientific process based on the study of the state of the art of IMSs and how these systems have evolved from traditional to current solutions. During this study we have focused the efforts on knowing how the advances made by new paradigms and technologies have influenced the management of the information and protection of the privacy of sensitive pieces, as well as their influence in the management and control of the behavior of the IMSs. This analysis has detected some of the main challenges of this domain allowing the definition of new proposals in different application scenarios using specific technologies explained along this chapter.

1.1.1. Information management systems

The early days of IMSs were focused on the management and storage of business information in companies and public organisms, such as administrations or universities. The evolution experienced by software and hardware technologies changed the focus and functionality of information management systems. Nowadays, these systems provide services that are consumed not only by companies or administrations, but also by people at anyplace and anytime. In order to provide these services to this wide variety of users, information management systems should now evolve to:

- Gather and handle large volumes of heterogeneous information.
- Evaluate and protect the privacy of sensitive pieces of information.
- Consider independent administrators distributed along different organizations.
- Manage diverse components with different requirements and locations.

These facts have increased the complexity of information management processes, thus requiring additional research efforts on new management mechanisms that consider the previous requirements. These mechanisms should be as automatic as possible in order to allow dynamic detections of events that imply the reconfiguration of the management processes. Furthermore, automatic mechanisms avoid delays during the management processes due to possible human fails or misconfigurations, and help to reduce the complexity of the management of distributed heterogeneous components.

The first proposal in charge of categorizing the different areas that should be considered by information management systems to reduce the complexity of the management process was called FCAPS (Fault, Configuration, Accounting, Performance, and Security). This solution was proposed by ISO (International Organization for Standardization) in the OSI Systems Management (OSI-SM) [1]. Among the five proposed areas, this PhD Thesis is focused on security and configuration, although the management of these areas also affects to important aspects of others, like for example, the systems' performance, or the fault tolerance.

The security management area is oriented to emphasize security considerations like the protection of the information handled by information management systems. The increment of the information managed by these systems during the last decades has influenced the necessity of protecting some sensitive pieces of information. For example, in Spain, depending on the nature of the information, the *Ley Orgánica de Protección de Datos* [2] categorizes it into three levels of security. The basic level contains personal information like, for example, the name, age, sex, address, telephone, or bank account number. The medium level contains information related to financial operations and personalities; for example, patrimonial assets, habits of consumers, criminal records, or curriculum. Finally, the last category has the highest level of security and is composed of information such as the ideology, political affiliations, religion, or health. The protection of these pieces of information is known in the literature with the term of privacy. Privacy refers to the rights of persons and organizations to determine for themselves when, where, how, and what information about them can be revealed [3]. In that sense, the consideration of the information's privacy also increases the complexity of the management process performed by information management systems. These systems should allow their users to control what pieces of information they want to reveal, the place where these pieces of information can be revealed, the moment or period of time, the situation in which the information should be revealed, and the person(s) or organization(s) to whom the information can be revealed.

On the other hand, another area proposed by FCAPS, in which this PhD Thesis is focused on, is the management of the systems' configuration and behavior to control the deployment of hardware and software components. In this sense, the networking paradigm is one of the topics in which industry and academic sectors are making more efforts to manage at real-time the resources composing the network infrastructure. Networks are dynamic systems composed of distributed and heterogeneous resources managed by different administrators. When the network status varies depending on different aspects like, for example, the number of connected users or the availability of resources, it is needed to reconfigure the network resources in order to continue providing services. The Software-Defined Networking (SDN) paradigm arose with the goal of reducing the complexity during the network management process by separating the control and data planes, and "softwarizing" the configuration of the network resources. However, performing reconfiguration and management of distributed resources is still a difficult task that needs of new mechanisms to ease it. These new mechanisms should allow networks administrators to control at real-time the network resources according to the current status of the network. The behavior of network resources should be adapted automatically, taking into account the network status and the decisions of administrators oriented to guarantee network requirements like the Quality of Service (QoS), energy efficiency, or fault tolerance.

Several proposals have been made during the last decades to reduce the complexity in the management processes performed by IMSs, as commented at the beginning of this section. One of the most relevant ones was made by the IETF with the definition of a new paradigm called Policy-Based Management (PBM) [4]. This paradigm arose with the aim of separating the behavior of systems from their functionality. This separation allowed the flexible, automatic, and dynamic management of the systems behavior and their information while reducing maintenance costs. One of the main goals of PBM consists on managing systems, information, and resources at a high abstraction level. By using this paradigm, systems administrators define policies, or rules, indicating the actions that should be applied when certain events are triggered. These rules are composed of conditions and decisions, where conditions are representations of the prerequisites that must be accomplished in order to enforce the actions established by the decisions. According to the PCIM (Policy Core Information Model) [5] developed jointly by IETF and DMTF, policies are stored in a repository called Policy Repository, and the entities in charge of checking if conditions are accomplished and making the policy decisions are called Policy Decision Point (PDP). Finally, entities enforcing decisions of the policies are called Policy Enforcement Point (PEP).

In order to express the intention of administrators and information's owners, several technologies and policy-based languages have been proposed during the last decades. eXtensible Access Control Markup Language (XACML) [6] is one of most well-known languages. It is accepted in industry and academia as de facto standard and it is mainly focused on access control management in distributed systems. KAoS (Knowledge Acquisition in automated Specification) [7] is another well-known language designed for goal-directed software requirements analysis. KAoS provides the capability of assigning system-level and organizational objectives rather than lower-level processor action-oriented descriptions. Among the proposed technologies, semantic web techniques [8] are a promising way to manage the information and the behavior of systems. Administrators of systems or users can manage the behavior of the systems' resources and the handled information by using semantic rules, also called policies. These policies let control the system's behavior at run-time and dynamically taking into account the preferences of the administrators or the owners of the information. Furthermore, ontologies [9] allow the formal representation of the information in a way that together with certain governing rules it is possible to infer new knowledge by using semantic reasoners. Furthermore, ontologies also allow sharing knowledge between independent systems and using semantic reasoning about the context to offer advanced services to customers. By considering these advantages, in this PhD Thesis we use semantic web techniques to model, manage, and protect the personal and contextual information considered by our proposals.

1.1.2. Location and context-awareness

Users' location is a concept that information management systems started to consider with the rise of mobile devices and the mobile paradigm [10]. Systems that provide information considering the location of persons or devices are called *Location-Based Services* LBS [11]. Some of LBSs go a step further providing useful services by taking into account the users' location to infer other aspects about the place where they are. To reach it, current information management systems monitor the users' location in order to consider it during the management processes. Depending on the focus of services, an LBS can be person-oriented or device-oriented. In the person-oriented approach, the focus on applications uses the position of a person to enhance the service. One example of a LBS belonging to this approach could be a friend finder application. On the other hand, device-oriented LBSs may also focus on the position of a person, but they do not need it. In this approach, objects like cars in navigation systems could also be located. In addition to this classification, there are two application designs distinguished: push and pull services. In push services, users receive information without having to actively request it. Pull services, in contrast, mean that users actively request the information. Most of the early location services have been pull services, although during the last years push services have gained popularity in certain domains.

Helped by the location information provided by the mobile paradigm, the *pervasive computing* paradigm arose at the beginning of the '90s. The pervasive concept was introduced by Weiser [12] as the seamless integration of devices into the users' everyday life. The main goals of pervasive computing were the effective and efficient use of the elements that compose the smart spaces where users were, the invisibility or complete disappearance of computing technology from the user's perspective, the number of users and computing resources that compose the environment and the scalability of the system, and the context-awareness. The pervasive paradigm made people the central focus rather than computing devices and technical issues. The users' mobility records were an essential piece of information for this paradigm and they were provided by LBSs. In this sense, information management systems considered not only the users' location, but also information about the users' life. As pervasive computing is still a very active and evolving field, the rate of penetration into the everyday life varies considerably depending on technical and non-technical factors such as the infrastructure, computation resources, security, or economics.

The *context-awareness* concept [13] is the core of the *pervasive computing* paradigm [12]. Context-awareness is a mobile paradigm where services discover the information of the context or environment in which users are located, and adapt their behavior taking into account that information without users' interaction. It is important to notice that users do not interact with context-aware systems, but they consent the acquisition and management of their information. Context-aware solutions use location-based systems so as to obtain the users' location with which to gather information about the people, objects, and elements that compose the context or environment in which the users are located. In the literature, we can find several definitions of context. The first one appeared in 1994 and it was made by Schilit et al. [14]. They described context as location, identities of nearby people, objects, and changes to those objects. After that, several authors included new aspects like, for example, identity and time (Ryan et al. [15]), emotional states and focus of attention (Dey [16]), context dimensions (Prekop and Burnett [17], and Gustavsen [18]), and any environmental data (Tajd and Ngantchaha [19]).

As it is stated by the previous definitions of context, the number of heterogeneous pieces of information managed by the context-aware paradigm is really high. The evolution experienced from traditional paradigms to the new ones has influenced this fact. First, the mobile paradigm incorporated the location of users and elements into the management processes of IMSs. Later, pervasive systems added other important pieces of the users' life like the users' identities, emails, notifications, and information related to smart spaces. Finally, the information about the context in which users are located, together with the information considered by the previous paradigms, has increased the complexity of the management processes. Figure 1.1 shows the evolution of IMSs considering the influence of new emerging paradigms and the most representative pieces of information usually managed by them. By considering that an important number of these pieces of information are sensitive or private, current IMSs need to consider the privacy of this information. In this sense, additional research on automatic mechanisms in charge of protecting the privacy of the sensitive information handled during the management processes of IMSs is needed. Otherwise, malicious users could know more information about the life of a given user that himself/herself.

The context-awareness is a paradigm that can be helpful in many different scenarios like, for example, the network management. Network administrators can incorporate

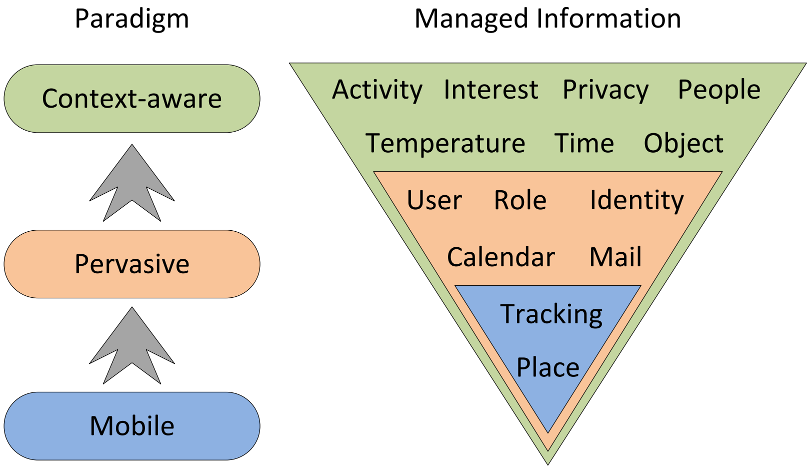


Figure 1.1: Evolution of information management systems

contextual information to manage dynamically the network resources by adapting their behavior according to the network status. The location of connected devices and network resources, the kind of devices connected to the network infrastructure, or the status and statistics of the network have been found as much appreciated pieces of contextual information that can be considered by network administrators before taking management decisions. These decisions could go from the reconfiguration of a given network component, which is not being used in an energy efficient way, to the deployment of new resources located in a given place to ensure the QoS of certain users.

1.1.3. Context-aware application scenarios

This section shows different scenarios where the PBM paradigm helps to information management systems with the processes of handling and protecting the information, as well as the management of the systems' configuration and behavior.

Management of location privacy

Protecting the privacy of the users' information in location-based services being part of users' everyday life represents a novel challenge [20]. Nowadays, location-based services run on computers, smartphones, tablets, and smart watches, providing users with added-value services. Examples of these services could be social networks, car navigation systems, or recommender systems. The protection of large volumes of heterogeneous information related to the users' location is a complex task that requires an automatic mechanism to address it accordingly. Location policies are a promising way to accomplish this protection on real-time and dynamically. In this sense, location policies should allow users to:

- Mask their location by generating one or more fictitious positions for a particular user. In this way, other users cannot distinguish the real position of the target.

- Hide their location when they do not want to release it to others. This will avoid that requester(s) know the position of the target.

- Indicate the maximum accuracy at which users want to be located. Depending on the environment in which users are located, several levels of granularity can be defined such as country, city, building, or floor, among others.

- Define the minimum level of nearness at which users want to be located. Nearness levels correspond to the same values defined for the granularity policies.

Hybrid recommender systems

The increasing volume of information received by people in their daily lives usually presents the challenge of deciding what information is useful for them, and which does not. Recommender systems are tools that can be used to suggest items that may not have been found by users themselves [21]. Traditional recommenders, like for example those based on Content-Based (CB) [22] and Collaborative Filtering (CF) [23], tend to use simple models in order to provide recommendations. The CB approach operates with the similarity of the items, so similar items to the ones liked by the target user are recommended. However, classifying items is a hard task that usually requires human knowledge. In that sense, the CF methods surfaced to overcome this drawback, considering stereotype-based models to establish the similarity between users. So, the CF models targeted to recommend items that people with similar preferences had liked.

The advent of mobile devices allowed the use of location information to improve the recommendations of traditional systems. Location-based recommender systems provide recommendations by considering the distance between users and items, as well as their subsequent movements. The ability to combine users' location and movements, together with other aspects like users' preferences, items' properties, or users' ratings, provides more valuable information that can help to suggest more accurate items of potential interest to users. The apparition of the context-aware paradigm provoked the consideration of contextual information to recommend items located close to the users. An example of contextual information could be the time, companion, or weather conditions found in the environment in which users are located. Usually, these context-aware recommenders not only consider pieces of contextual information, but also information related to other approaches like locations, preferences, properties, or likes. The combination of the previous information during the recommendation process lets a better adaptation of the recommendations to the context status. An example of this fact could be that context-aware recommender systems are able to recommend umbrellas or raincoats when it starts raining close to a shopping center. The combination of contextual information, together with information belonging to other approaches, also presents certain privacy challenges that have to be addressed adequately. Users should be able to dynamically control what pieces of their information they want to reveal to recommender systems. In this sense, users should define their privacy preferences by using policies related to their location, identity, and personal information.

Protecting information in eHealth scenarios

The evolution of traditional health systems has been influenced by the progress experienced by technologies, communications, and medical services. During the last years, there has been a lot of research in the healthcare topic with the goal of evolving from traditional paper-based systems towards electronic-based systems that manage digital records. Personal Health Records (PHR) and Electronic Health Records (EHR) are the electronic versions of patient health information. The former is controlled by patients themselves, while the latter is managed by healthcare systems. The provision of health services using digital technology is known in the literature as eHealth [24].

Despite the profits presented by this evolution, some open challenges have also appeared like the necessity of a common infrastructure and standard information models to guarantee the interoperability of systems. Furthermore, the large volume of data related to EHR and PHR, along with the contextual information provided by the proliferation of context-aware services ubiquitously accessible, means that managing and protecting the privacy of the patients' information is an even greater challenge. In order to partially address this challenge, context-aware applications can be useful and helpful in managing the patients' information, with concern for patients' privacy and how personal information, location, and context information are revealed. In this sense, users of context-aware eHealth systems should be able to manage dynamically the privacy of their medical records, personal information, locations, and information related to the environment or context in which they are located. In order to cover these requirements, the PBM paradigm can help with the definition of policies that allow both users and administrators to manage and control sensitive information.

Networking paradigm

Computer networks are dynamic and complex systems, and therefore their configuration and management continue to be challenging. Networks are composed of a large number of resources such as switches, routers, firewalls, and middleboxes in charge of forwarding packages between them. Network administrators are responsible of configuring and managing these resources, representing a really difficult task due to the number of different events occurring simultaneously and the heterogeneity of the network resources. To automatize this management, the Policy-Based Network Management (PBNM) paradigm [25] allows network administrators to define policies to control the behavior of the network resources as well as the packages traveling along the network. Using policies, network administrators are able to specify, for example, which kind of services has more priority to guarantee the QoS, or what network resources should be switched off because they are consuming energy in an inefficient way.

Despite the progress carried out by the PBNM paradigm, the recent technology advancements in mobile devices and networks have encouraged users' mobility, thus being location one of the most important aspects for knowing where devices, resources, or people are. Combining location and mobility information, together with other important contextual information like the status of the network resources, or the statistics of the packages traveling along the network, have made the network management an actual difficult task. Nowadays, network administrators must define increasingly sophisticated policies and complex tasks, which requires considering the previous context-aware information. Furthermore, the rigidity of the infrastructure provides few possibilities for on demand innovation or improvement, since network devices have generally been closed, proprietary, and vertically integrated.

1.2. Objectives

Once introduced the core topics in which this PhD is focused on, the next step consists on defining the sub-objectives that we pretend to reach, and the contributions made to reach the main objective of this PhD Thesis: the definition, design, and deployment of policy-based and context-aware systems that protect automatically the privacy of the users' information as well as the ability to control the behavior of the systems resources.

The specific goals pursued within this thesis, which are derived from the main objective described above, are the following ones:

- Study and analyze the evolution of information management systems from location-based solutions to context-aware systems, as well as how web semantic techniques can help in managing and protecting the privacy of the information considered by these systems.

- Propose, implement, and validate a location-based and context-aware system in charge of exchanging and protecting the users' location by considering semantic web techniques.

- Design and implement a context-aware solution capable of exchanging in a privacy-preserving way personal and contextual information between users and services.

- Design, implement, and validate a multi-context solution capable of exchanging sensitive information between users located at different contexts.

- Design, implement, and validate a location-based and context-aware system to manage automatically network resources taking into account aspects like QoS, energy efficiency, or performance.

The validation of the proposals enumerated earlier has been carried out through intensive simulations in a controlled lab environment. All these experiments have been transferred subsequently to different application scenarios. It is also worthy to emphasize that this PhD Thesis has been conducted by following a scientific process based on the study of the state of the art and the analysis of the results provided by the development of the existing context-aware solutions.

In order to accomplish the first objective, this PhD Thesis started studying and analyzing the current state of the art of information management systems. Specifically, we focused our efforts on how location-based and context-aware solutions can influence the information management processes of IMSs. After this study, we realized that despite the countless advances made by existing context-aware solutions, the majority of them do not protect the privacy of the users' information, or they have not been taking into consideration in their design the protection of important pieces of sensitive information. In order to find the best way to protect the users' information in context-aware systems, we studied and analyzed the state of the art of privacy-preserving solutions within this kind of systems. With this study, we discovered that the majority of solutions do not manage dynamically the information nor provide users with effective means to protect important pieces of sensitive information. In this sense, we started analyzing techniques to protect this information. The main conclusion of this analysis was that systems based on policies are a promising way to manage the privacy of the users' information dynamically. Specifically, semantic web techniques have evolved enough to be used as a mechanism to model the contextual information, to define policies that protect the users' information to infer new knowledge by means of semantic reasoners, and to share information between independent systems.

In order to reduce the previous drawbacks and accomplish the second, third, and fourth goals defined in this thesis, we have designed, implemented, and evaluated several location-based and context-aware solutions in charge of allowing users to protect the privacy of their information. During the research performed to know the current state of the art of information management systems, we also found a lack of proposals considering contextual information to control the behavior of network resources. In this sense, and related to the fourth goal proposed, we have proposed, designed, implemented, and validated context-aware solutions that control automatically the behavior of the network resources taking into account aspects like the QoS or the energy efficiency.

1.3. Contributions

The complexity of IMSs, influenced by the evolution of computing paradigms and solutions, has generated new open challenges in the information management processes. As it has been defined in the previous section, the goal of this PhD Thesis is focused on allowing these processes to control the privacy of the sensitive information as well as to manage automatically the systems resources considering the information provided by the context-aware paradigm. In this sense, and oriented to the objectives defined in the previous section, this PhD Thesis presents the next contributions:

- A context-aware architecture in charge of controlling automatically the information of users and contexts as well as the systems behavior. It is required a common architecture for the whole set of objectives defined in the previous section. This context-aware architecture should be based on semantic web techniques to gather and handle large volumes of heterogeneous information; evaluate and protect the privacy of sensitive pieces of information; consider independent administrators distributed along different organizations; and manage diverse components with different requirements and locations.

- A set of policies to protect dynamically the privacy of sensitive information considered by the context-aware paradigm. This contribution is oriented to overcome the first four objectives of this PhD Thesis. In this sense, users should be able to decide when, where, how, and to whom they want to disclosure private pieces of information handled by the context like, for example, locations, activities, and identities, among others. In this sense, we have defined a set of privacy-preserving and context-aware policies in charge of allowing users to protect their information at real-time and dynamically.

- A privacy-preserving mechanism to exchange personal and contextual information between independent and different contexts. Following the previous contribution, this one is also oriented to the first four objectives. Specifically, the exchange of sensitive information in multi-context scenarios in a secure way is a capability required in the majority of current IMSs. In this sense, we have proposed a solution oriented to semantic web to model the information in a formal way and allow the secure exchange of information, taking into account the privacy preferences of the owners' information.

- The management of the network resources by considering location-based and context-aware information. Oriented to the last objective of this PhD Thesis, it is required a mechanism to control dynamically and at real-time the network infrastructure, taking into account the information about the context of both where the network provides services and end users are receiving these services. In that sense, the last contribution of this PhD Thesis is oriented to combine the network management with the contextual information so as to guarantee some requirements like, for example, QoS or energy efficiency.

1.4. Related Publications

This PhD Thesis has conceived from the early beginning as a compilation of publications, so that the results are essentially exposed in the articles that compose it. The users' mobility has brought an evolution from proposals that protected the users' information in specific contexts (intra-context solutions) to systems that control the users' information by considering multiple and independent contexts (multi-context solutions). Among the solutions that protect the information in intra-context scenarios, and following the methodology defined in the previous section to cover the first and second objectives of this thesis, we have designed, implemented, and validated a context-aware solution that allows the development of applications by considering the privacy of the users' location. The results derived from the evaluation of the state of the art, as well as the definition and implementation of a context-aware system, have been reported in the IEEE Systems Journal, Special Issue on Intelligent Internet of Things, in an article entitled "SeCoMan: A Semantic-Aware Policy Framework for Developing Privacy-Preserving and Context-Aware Smart Applications" [26]. In this paper, we proposed a system that provides support for developing context-aware and smart applications, preserving the users' privacy in a semantic-oriented vision. Our system is in charge of obtaining the information from the context or environment in which users are located, modeling and protecting their information, and providing context-aware applications. To perform the previous process, we have designed and implemented an architecture oriented to the web semantic techniques [27]. The information about users and contexts is modeled by using a collection of ontologies defined in OWL 2 [28]. The users' information is protected by using policies defined in SWRL [29]. These policies allow users to share their location to the selected users, at the desired granularity, at the right place, and at the right time. Finally, the information about the contexts and users is provided to context-aware applications by using queries defined in SPARQL [30]. In this publication, we included the analysis and comparison of the different context-aware systems. We also included the results regarding the implementation and evaluation of a prototype for a supermarket context, which was chosen due to the amount and richness of the different location-based services that might be potentially provided. A performance analysis showed reasonable times while managing, protecting, and providing information about users and their contexts.

Afterwards, and considering the first and third goals defined in this thesis, we investigated further into the protection of the users' information in context-aware solutions. As a result of this research, it was proposed and implemented a platform oriented to protect the users' information regarding services. This result was defined in an article entitled "PRECISE: Privacy-Aware Recommender Based on Context Information for Cloud Service Environments" [31], which was published in IEEE Communications Magazine, Feature Topics Issue on Context-Aware Networking and Communications. In this paper, we defined a privacy-preserving and context-aware system that provides context-aware recommendations by considering the information that users want to reveal to given services. By using our solution, users can release their locations to specific services, hide their positions to specific users, mask their locations to other users by generating fictitious (fake) positions, establish the granularity and closeness at which they want to be located by services or users, and preserve their anonymity to specific services. To this end, we have designed and deployed an architecture oriented to the MCC paradigm composed of three layers. In our architecture, services are allocated at the Software as a Service (SaaS) layer of the MCC paradigm, providing users with recommendations about context-aware information. The central element of our system is a middleware allocated at the Platform as a Service (PaaS) layer, which preserves the users' information, and manages the information about the context and space that can be provided by independent systems. In this work, we have also included a comparison between the context-aware solutions that can be found in the literature and the main differences with our solution. Among these differences, we highlight the amount of location-aware policies that our solution provides allowing users to have a higher control of their location information.

Multi-context solutions consider intra- and inter-context scenarios in order to protect the users' information when they move between independent contexts or environments. In that sense, the results obtained after covering the first and fourth goals of this thesis was defined in an article entitled "What Private Information are you Disclosing? A Privacy-Preserving System Supervised by Yourself" [32], which was published and presented on the Proceedings of the 6th International Symposium on Cyberspace Safety and Security. In this article, we defined a first approximation to multi-context solutions. Specifically, we proposed a framework called CAPRIS (*Context-Aware PRIvacy-preserving system Supervised by users*) in charge of protecting the users' information in the context where they were. Using CAPRIS, users were able to decide at real-time what, where, when, how, to whom, and at which level of precision they want to release their information. This information can be the *space* in which they are located with different levels of granularity; the users' *personal information* with different levels of precision; the users' *activity*; and the information oriented to the *context* in which they are located. Using CAPRIS, users did not have to manage their privacy, but they just have to choose the most appropriate group of policies suggested by the system administrator using our solution.

As evolution of CAPRIS, covering the same goals, we have defined a solution that protects the privacy of the users' information in multi-context scenarios, incorporating the user consent to reveal his/her personal information. This proposal has been defined in a conference paper entitled "MASTERY: A Multicontext-Aware System that Preserves the Users' Privacy" [33], which was published and presented on the IEEE/IFIP Network Operations and Management Symposium. MASTERY suggests to the users several sets of privacy-preserving and context-aware policies, called *profiles*. In order to protect their information, users just have to choose the most suitable profile according to their interests in the context where they are, being able to modify these profiles by adding, deleting, or modifying some of the policies shaping them. Finally, when the information is going to be shared the owner receives a notification at real-time and he/she decides if grant or deny the exchange of information. The policies that compose the privacy-preserving and context-aware profiles can be categorized into two different groups: intra- and inter-policies. They allow users to protect their location, personal information, activities, and context-aware information. Specifically, intra-policies protect the users' information inside of specific contexts, and inter-policies preserve the users' information between different contexts. In order to protect the information, the intra- and inter-policies' groups are composed of *disclosure* and *reveal* policies. Disclosure policies are in charge of indicating what information of the users can be shared, while the reveal policies indicate where, when, and how the information can be shared.

Regarding network management, important improvements in terms of energy saving can be achieved by managing the network resources at run-time, considering the mobility of users and services. In that sense, and in order to address the first and fifth goals defined in this PhD Thesis, we have proposed a mobility-aware policy-based system in charge of reducing the energy consumption in networks oriented to the SDN paradigm. This solution was published in an article entitled "Policy-Based Management for Green Mobile Networks Through Software-Defined Networking" [34], which was published in Mobile Networks and Applications. The policies defined in our solution allow the SDN paradigm to switch on/off network resources when they are consuming energy in an inefficient way, as well as create virtualized network resources like proxies to reduce the network traffic generated by users consuming services close to the network infrastructure. Network administrators define policies that will decide the list of potential actions to be taken by the SDN components, in accordance with the energy consumption, the users' mobility, and the network statistics. We also propose an architecture in charge of managing the resources of mobile networks considering the previous policies and the ontology that models the concepts that belong to the mobile network topic. With this ontology, we provide a set of primitives with which to describe a collection of the resources managed by the SDN paradigm and the relationships among them.

Finally, and also oriented to the first and fifth goals, we have designed a proposal oriented to ensure the QoS and end-user experience in dynamic scenarios of mobile networks. This work was defined in the article entitled "Enabling highly dynamic mobile scenarios with Software Defined Networking" [35], which is currently accepted for publication in IEEE Communications Magazine, Feature Topics Issue on SDN Use Cases for Service Provider Networks. Our proposal is in charge of managing the SDN resources at run-time, using high-level policies. Among the different sets of policies, we emphasize here the use of mobility-aware management-oriented policies, defined by the service provider network administrator to decide the actions made by the SDN according the network infrastructure statistics and location, and the mobility of users and services. These policies are oriented to guarantee end-users experience in very crowded places (e.g., stadiums, shopping malls, or unexpected traffic jams). To this end, the policies decide when the SDN should balance the network traffic between the infrastructure located close to the congested one; when the SDN should create or dismantle physical or virtual infrastructure in case of the congested one is not enough to accomplish the end-user demand; and when the SDN should restrict or limit specific services or network traffic in critical situations produced by large crowds using services at specific areas. We also proposed an architecture that depicts the components needed to manage the network resources considering the previous policies.

1.5. Related Work

1.5.1. Policy-based management

In the current literature, it can be found several works that trace the history and evolution of the policy-based management paradigm [36, 37]. The early works oriented to this paradigm were focused on emphasizing security considerations. In this sense, security policies were the first policies in charge of defining rules according to which access control were regulated [38]. Access control mechanisms [39] manage if the access to given resources should be granted or denied according to the security policies defined by the system administrator. Security policies can be grouped in different access security levels with diverse criteria for defining what should and should not be allowed. The access-control matrix [40] was introduced to protect different objects in shared computers. Different access rights protected objects like files, memory, or terminals which were shared between different domains. Each entry of the matrix contained a list of access attributes which define the access rights of that domain to the objects. Attributes could be of different forms, such as read, write, owner, call, control, etc. Access Control Lists (ACL) [41] was another solution proposed as an alternative approach of the access-control matrix presenting the matrix information in a column fashion.

Confidentiality and integrity are two important aspects belonging to the security topic which are considered by information management systems. The confidentiality policy model was the earliest formal model designed to prevent the unauthorized disclosure of information [42]. This model was proposed to formalize the security of the United States Department of Defense. It followed the state machine concept, where are defined a set of access-control policies grouped by states, and the transition functions between them. Security states ranged from the most sensitive *(Top Secret)* to the least sensitive *(Public)*. Users of this model could access and create content only at or above their own security level. Finally, although Role-Based Access Control (RBAC) [43] is not directly concerned with policy specification, it was accepted as a security model that permits the specification and enforcement of organizational access control policies. Specifically, RBAC is a security mechanism that allows the authorization management by separating user assignment to roles. On the other hand, information integrity is the assurance that the data has not been altered or damaged by malicious users or systems errors [44]. In this sense, integrity policies describe how the validity of the information in the system should be maintained. The first model for integrity policy established the required considerations for secure computers systems [45].

Privacy is another key aspect of the security domain. Privacy is the ability of individuals or organizations to control the terms under which their private information is acquired and used [46]. During the last years have been proposed many works based on privacy policies and oriented to protect the sensitive information. Privacy policies represent long-term promises made by an enterprise to its end users and are determined by business practice and legal concerns [47]. In this sense, a system which facilitated privacy policy authoring, implementation, and compliance monitoring was proposed in [48]. Another solution was proposed in [49], where a framework was able to express and enforce privacy policies by giving a formal definition of purpose, and proposing a modal-logic language for formally expressing purpose constraints. In [50], it was described a platform for Enterprise Privacy Practices (E-P3P) that defines technology for privacy-enabled management and exchange of customer data. This solution separates the enterprise-specific deployment policy from the privacy policy that covers the complete life cycle of collected data. Regarding the evaluation of the privacy policies effectiveness, several tests were performed in [51] to determine if apps with privacy policies were more likely to protect personal information than apps without privacy policies. In [52], an empirical study of online privacy policies, as well as tools for users with privacy concerns were presented. Finally, related to the e-commerce domain, in [53] was investigated the relationship between privacy policies and users' reactions, showing privacy risks transfer the effects of a privacy policy's contents on user behavior.

Among the wide spectrum of active research topics where policies can be applied to simplify management tasks [54], the networking paradigm has focused attention both from industry and the academic research community in recent years. In the literature we can find several proposals oriented to the PBNM with different purposes [55].

Regarding the security and privacy of networks, a framework to refine policies in policy-based management systems was proposed in [56]. The refine process was made following list of steps to convert high-level goals into low-level policies. Another proposal, was defined in [57], where a policy-based access control framework overcame harmful interference created by malfunctioning devices or malicious users. In this solution was designed a set of policy-based components integrated with the algorithms employed by software-defined radios to detect interference caused by malfunctioning devices. The proposed policy-based components ensured that a radio did not violate the requirements established by policies. This work also proposed secure policy management and distribution mechanisms to avoid that malicious users added or modified the existing policies. FRESCO [58] was an OpenFlow security application development framework designed to facilitate the rapid design, and modular composition of detection and mitigation modules. Regarding the anonymity of network communications, in [59] was studied the compatibility of the Destination Addressing Control System (DACS) scheme for the cloud environment with virtualization technology. Specifically, they proposed the consideration of the DACS into the PBNM to manage network resources through policies. Ethane [60] was another solution that allowed network administrator to define access control policies using the Flow-based Security Language (FSL). During the last years, with the advent of SDN, the network management has evolved to become more dynamic [61]. In this sense, OpenSec [62] was a framework based on OpenFlow that allowed network administrators to create and implement security policies. These policies defined which security services must be applied and specified security levels that define how OpenSec reacts if malicious traffic.

Energy-efficient network approaches are also focusing the efforts of industry and academic during the last years. In this sense, in the current literature can be found a deep survey with a deep comparison of a number of energy-efficient network approaches [63]. The solution presented in [64] quantified power consumption of mobile communication systems. It established that there is a high potential to reduce the energy consumption when improving the energy efficiency of the BSs at low traffic load. Another proposal to enhance efficiency of power amplifier for wireless Base Stations (BSs) was proposed in [65]. The variation of the traffic patterns over time, in order to decide when BSs should sleep or not, was considered in [66]. Another solution was proposed in [67], where a control mechanism enabled small cells to switch off all components while not serving active connections. In order to speed up the decision process of switching on/off BSs, a transfer actor-critic algorithm (TACT) was proposed in [68], making use of historical data from neighbor regions. In [69], it was proposed the use of software routers for emulating network equipment functionality and discussed about the benefits of the emulation environment. Finally, another solution oriented to the energy efficiency was proposed in [70]. Specifically, it was defined a policy-based platform oriented to the management of resources in fog computing. This solution expanded the fog computing concept to support secure collaboration and interoperability between different user-requested resources. Several scenarios were proposed to show the necessity of policy management as a core security management module in a fog ecosystem.

The QoS is another important topic belonging to networks in which several policy-based solutions have been proposed. Procera [71] was an event-driven network control framework that used high-level policies to manage and configure the network state. This solution enabled dynamic policies, which were translated into a set of forwarding rules to manage the network state by the controller. Oriented to overcome the limitations of the standardized objective functions, it was proposed a new objective function for IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) based on Fuzzy Logic that allows to select the best paths to the destination [72]. A proposal about a novel solution to transfer learning applied to spectrum management in cognitive radio networks to improve the QoS was performed in [73]. In this proposal was demonstrated that transfer learning achieves a significantly higher QoS and throughput than distributed reinforcement learning. When several policies coexist, conflict resolution is a crucial aspect when managing a system using policies. Indeed, in [74], it is addressed the problem of conflict resolution when using policies to provide QoS. Following the SDN approach, PolicyCop [75] was a QoS policy management framework oriented towards OpenFlow and based on SDN. PolicyCop allowed to define QoS SLAs and control the policy enforcement. This solution also monitored the network status to reconfigure the network parameters through policies.

1.5.2. Context-awareness

During the last years, many context-aware services have been proposed in order to make life easier. Despite the context term was proposed in 1994, it is commonly agreed that the first context-aware solution found in the literature was proposed in 1991 [76]. They introduced a novel system for the location of people in an office environment called Active Badge. This system was able to know the location of users in order to forward phone calls to the phones close to the user. The users' location was known because users wore badges transmitting signals to a centralized location system. After that, many different solutions have been proposed in different topics, as the one presented by Wood et al. in [77], for example. They described a *teleporting* solution, which was able to make the user' environment available in any computer running a web browser with Java. By using this solution, users did not have to carry any computing platform and they were able to execute their applications on any nearby machine. These context-aware systems opened new possibilities to users in terms of acquiring custom services by gathering context information, especially in applications where the high mobility of users increased their usability. Car navigation systems, emergency services, or recommender systems are well-known examples of context-aware solutions where the mobility of users increases their usability.

Nowadays, a depth survey on context-aware systems analyzes a large number of solutions considering different topics [78]. Systems like Feel@Home [79], Hydra [80], CroCo [81], SOCAM [82], and CoBrA [83] provide support to "security and privacy" features, as our solution. Feel@Home was a context-aware framework that supported communications between contexts or domains, considering intra- and inter-domain interactions. Hydra was an ambient intelligence middleware oriented to the IoT, which integrated the device, semantic, and application contexts to offer context-aware information. On the other hand, CroCo was a cross-application context management service for heterogeneous environments, while SOCAM used a collection of ontologies to shape the quality, dependence, and classification of the context information. This collection was built on a common upper ontology for all contexts, as well as for domain-specific ontologies that defined concepts of each one. Another related work in this context, which is not included in the survey presented in [78], is CoCA [84]. This proposal presented a collaborative context-aware service platform, where a neighborhood-based mechanism to share resources was introduced. CoCA inferred users' location by considering information about the context and the location of the elements. SHERLOCK [85] was a framework with functionalities for location-based services that used semantic technologies to help users to choose the service that best fits their needs in the given context. Hydra [80] was another solution oriented to Internet of Things. It was a middleware in charge of delivering solutions to wireless devices and sensor used in ambient awareness. It considered powerful reasoning toward various context sources including physical device based, semantic and abstract layer based. PerDe [86] was a development environment focused on pervasive computing applications oriented towards the user's needs. It provided a domain-specific design language and a set of graphical tools to cover some development stages of pervasive applications. DiaSuite [87] was another a development methodology that used a software design approach to develop applications in the domain of Sense/Compute/Control (SCC) applications. In addition, DiaSuite had a compiler that produced a dedicated Java programming framework, guiding the programmer to implement the various parts of the software system.

Some of the solutions described above make used of semantic rules for different purposes. Hydra, CroCo, SHERLOCK, and SOCAM used semantic rules to infer new information about a given context, taking into account information from others. Instead, CoCA made use of semantic rules to manage the ontologies, e.g., a property is the inverse of another property, as well as additional information about the domain. Yet, none of these four solutions used semantic rules to define policies oriented to protect users' privacy preferences. Users' privacy should be supported by any context-aware framework, with which the users are capable of dynamically restricting or disclosing information to others depending on their location and their preferences in terms of privacy. Consequently, the current trend in context-aware systems focuses on controlling the disclosure of users' location by using policies.

		Privacy-preserving support		
Solutions	**Managed Contextual Inform.**	**Identity**	**Location**	**Context**
SHERLOCK	Place, People, Object, Identity, Tracking	X	X	X
CoCA	Place, People, Object, Identity, Tracking, Calendar	X	X	X
PerDe	Place, People, Object, Identity, Tracking, Time, Activity	X	X	X
DiaSuite	Place, People, Object, Identity, Tracking, Temperature, Role	X	X	X
SOCAM	Place, People, Object, Identity, Tracking, Activity, Time	X	X	X
CroCo	Place, People, Object, Identity, Tracking, Activity, Time	Partially supported by ACL		
Hydra	Place, People, Object, Identity, Tracking, Time	Partially supported by ACL		
Feel@Home	Place, People, Object, Identity, Tracking, Calendar	Partially supported by ACL		
CoBrA	Place, People, Object, Identity, Tracking, Activity, Role	√	X	X
CoPS	Place, People, Object, Identity, Tracking, Time, Role, Notification	√	√	√
PPCS	Place, People, Object, Identity, Tracking, Time, Role, Activity, Interest, Mail	√	√	√

Table 1.1: Comparative of context-aware systems

There are a number of systems based on semantic web that manage policies to preserve users' privacy. For example, CoBrA presented a context-aware architecture that allowed distributed agents to share information with each other. CoBrA defined an ontology that shaped spaces composed of smart agents, devices, and sensors, and protected the privacy of its users by using rules that deduce whether they have the right permissions to share and/or receive information. Another example was PPCS [88], where a semantically rich, policy-based framework with different levels of privacy to protect users' information in environments with mobile devices was presented. Dynamic information observed or inferred from the context, along with static information about the owner, was taken into account to make access control decisions. Location and context information of the users were shared (or not) depending on their privacy policies. Another proposal supporting privacy policies without using semantic web technologies was CoPS [89]. In CoPS, users could control who access their context data, when, and at what level of granularity. In Table 1.1 is illustrated a comparative between the previous solutions taking into account the contextual information managed by them and their privacy support.

1.6. Conclusions and Future Work

The evolution of technology has increased the complexity of the information management processes performed by IMSs. Current IMSs handle large volumes of heterogeneous information, protect the privacy of sensitive pieces, allow different administrators to manage the resources, and consider distributed scenarios. The majority of IMSs are nowadays consumed by users, companies, or public administrations at anytime and anyplace. This fact has influenced that the location of users has become a very valuable piece of information in order to provide services located close to the users. With the consideration of the location, the pervasive and context-awareness paradigms have also included new pieces of information about the environment or context where users are, like for example, locations, activities, identities, time, emotional states, or any environmental data. This new heterogeneous information has increased the complexity of the previous management processes influencing the emergence of new automatic management mechanisms.

Controlling the behavior of the system resources as well as managing and protecting the users' information in IMSs that consider contextual information are open issues that still require efforts to be addressed. Administrators of IMSs systems should be able to consider the contextual information during the management processes in order to make decisions about the behavior of the system resources. Furthermore, users of IMSs should decide and control what information they want to reveal, where and when the information will be exchanged, and to whom. Semantic web techniques are a promising way to accomplish the management and protection of contextual and personal information in context-aware systems. This technology allows modeling the information in a formal way, exchanging information between independent systems, defining privacy-policies to protect the information, and inferring new knowledge by considering the information and the policies. By taking into account the previous facts, in this PhD Thesis has been proposed, analyzed, and implemented a set of solutions to enhance the privacy of the users' information during the management processes and to control the behavior of the systems resources considering contextual information.

Regarding the protection of the sensitive information in context-aware scenarios, we have proposed several solutions. The first one is a trusted middleware called SeCoMan that allows users to define their privacy-policies to control where, when, how, and to whom their location information will be revealed to other users. In this sense, users do not have to rely on the context-aware applications, so that our solution is in charge of providing a set of predefined queries that provides information to the context-aware applications. This information is provided by taking into account the privacy-policies defined previously by users in our middleware. In this solution, we have also designed a set of ontologies that model the space and contextual information considered in a supermarket context. Additionally, we have proposed another context-aware solution oriented to the MCC paradigm, which is in charge of protecting the users' information not only regarding users, but also services. In order to show the usefulness of our solution, we have developed a context-aware recommender capable of providing recommendations to users by considering the information they reveal to the services that provide the recommendations. Using our system, users can control their information by hiding their identities to services, hiding their locations, and providing fake positions, among others. The previous two solutions are oriented to protect the privacy of the users' information in a given context (intra-context scenario). Our last proposals in this particular research line are in charge of protecting the privacy of the users' information in multi-context scenarios (intra- and inter- context scenarios) incorporating the user consent to reveal his/her personal information. By using these solutions users do not have to define their privacy policies. In fact, CAPRIS and MASTERY suggest to the users several sets of privacy-preserving and context-aware policies called profiles which are defined by the administrators of the system where any of our solutions is being used. In order to protect their information, users just have to choose the most suitable profile according to their interests in the context where they are. Furthermore, using our solution users are able to modify the profiles adding, deleting, or modifying some of the policies that form the profiles. Finally, when the information is going to be shared the owner receives a notification at real-time and he/she decides if grant or deny the exchange of information.

On the other hand, regarding the other main topic in which this PhD Thesis is focused on, the management of the system's resources considering the information of the context where they are, we have focused our efforts in the networking paradigm. Our first solution is a mobility-aware policy-based system in charge of reducing the energy consumption in networks oriented to the SDN paradigm. The policies defined in our solution allow the SDN paradigm to switch on/off network resources when they are consuming energy in an inefficient way, as well as create virtualized network resources like proxies to reduce the network traffic generated by users consuming services close to the network infrastructure. Network administrators define policies that will decide the list of potential actions to be taken by the SDN components, in accordance with the energy consumption, the users' mobility, and the network statistics. Finally, our last proposal is oriented to ensure the QoS and end-user experience in dynamic scenarios of mobile networks. Specifically, we proposed a framework in charge of managing the SDN resources at run-time, using high-level policies. Among the different sets of policies, we emphasize here the use of mobility-aware management-oriented policies, defined by the service provider network administrator to decide the actions made by the SDN according the network infrastructure statistics and location, as well as the mobility of users and services.

This PhD Thesis improves some of the drawbacks found in the literature regarding IMSs oriented to context-aware scenarios. In this sense, regarding the protection of the sensitive information, our first contribution was focused on allowing users to control the disclosure of their locations regarding other users by defining privacy-policies. The second proposal of this thesis consists on a context-aware recommender system that allows users to protect other pieces of their sensitive information not only regarding users, but also services. Finally, the third and fourth proposals suggest users privacy profiles to avoid they define their own policies to protect the privacy of their information when they move between different contexts. On the other hand, regarding the management of the network resources taking into account the contextual information, our first solution allowed network administrators to define policies that create, dismantle, switch on/off network resources to reduce the energy consumption. Finally, our last policy-based proposal considers the users' mobility and contextual information during the management of network resources to ensure the QoS.

All these proposals are intended to reach the main objective of this PhD Thesis: the definition, design, and deployment of context-aware solutions that allow protecting sensitive information as well as controlling the behavior of the system resources.

As future work, we plan to allow users to protect their sensitive information during the management processes performed by IMSs. It is needed to consider the privacy of the users' information and contexts when administrators manage the systems resources. An example of this fact consists on allowing users to define the granularity at which they want reveal their location to network administrators when they are managing the network infrastructure taking into account the distance and location of devices. Regarding the network management topic, we plan to continue working on networks that combine concepts like Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) to ease the management of the network infrastructure and its services. In this sense, the Network Slicing technique can combine the previous technologies to manage the network resources and services depending on the current networks' requirements. These slices and their resources should be managed automatically considering the contextual information.

Publications composing the PhD Thesis

Chapter 2

SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications

Title:	SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications
Authors:	Alberto Huertas Celdrán, Félix J. García Clemente, Manuel Gil Pérez, Gregorio Martínez Pérez
Journal:	IEEE Systems Journal
Impact factor:	2.114 Q1 (2015)
Publisher:	IEEE
Volume:	10
Number:	3
Pages:	1111-1124
Year:	2016
Month:	September
DOI:	10.1109/JSYST.2013.2297707
State:	Published

IEEE SYSTEMS JOURNAL, VOL. 10, NO. 3, SEPTEMBER 2016

SeCoMan: A Semantic-Aware Policy Framework for Developing Privacy-Preserving and Context-Aware Smart Applications

Alberto Huertas Celdrán, Félix J. García Clemente, Manuel Gil Pérez, and Gregorio Martínez Pérez, *Member, IEEE*

***Abstract*—This paper is intended to provide a solution for developing context-aware smart applications preserving the users' privacy in the Internet of Things (IoT). In this sense, we present a framework called Semantic Web-based Context Management (SeCoMan) aimed at offering a set of predefined queries to provide applications with information about indoor location of users and objects, as well as context-aware services. SeCoMan uses a semantic-oriented IoT vision where semantic technologies play a key role. In fact, SeCoMan uses Semantic Web for modeling description of things, reasoning over data to infer new knowledge, and defining context-aware policies. SeCoMan also defines a layered architecture, including functions related to the management of the users' privacy in a manner that accommodate IoT requirements, in addition to not affecting system performance nor introducing excessive overheads. A thorough discussion on other related works, together with some experiments to measure the throughput and scalability, confirm that SeCoMan is a solution that improves the most relevant proposals existing so far.**

***Index Terms*—Context awareness, Internet of Things (IoT), pervasive computing, privacy preserving, semantic reasoner.**

I. Introduction

THE INTERNET of Things (IoT) enables the design and creation of smart objects, exploring new ways of user interaction in smart spaces as well as the development of smart services [1]. Smart spaces are characterized by being areas for cooperation of objects and systems, and for ubiquitous interaction with people. The deployment of smart applications is a complex process due to the lack of frameworks providing support for essential tasks, such as acquiring the information generated by the IoT from various sources, performing context interpretation and inferring new knowledge based on such context, managing rules to dynamically create new knowledge, defining basic location queries that provide context-aware information, allowing users to manage how the framework should use their locations regarding their privacy needs, sharing the knowledge among heterogeneous systems, and providing specific tools to develop smart applications. Many frameworks for developing smart applications use a semantic-oriented IoT vision, where semantic technologies play a key role. In fact, there are solutions based on Semantic Web in a manner that accommodate IoT requirements, but none of them fully supports all the previous tasks.

In order to conduct such tasks, we present in this paper a solution called Semantic Web-based Context Management (*SeCoMan*). Our main contribution behind SeCoMan is to provide support for developing context-aware smart applications preserving the users' privacy in a semantic-oriented IoT vision. Smart applications will be able to gather the information generated by the IoT using a set of queries predefined in SeCoMan, which are categorized into six groups: *operational queries*, providing context-aware information; *location queries*, yielding the indoor location of the elements (objects and people); *range queries*, supplying the elements contained in a given place; *closeness queries*, supplying the elements close to the requester; *navigation queries*, giving the path to arrive to a place or element; and *authorization queries*, providing specific information about the users' permission to stay in a place. The space and context information is shaped in a structured way by using a collection of ontologies [2]. Furthermore, the use of semantic reasoners allows us to infer new knowledge that can be easily shared with other independent systems.

In an IoT context, privacy is a critical issue often overlooked by schemes proposed to date. This fact has been recently identified in [3]. Perera *et al.* argue that privacy is a significant challenging issue in the IoT, and it is largely unattended at the context-aware middleware level in the existing solutions. To address this, SeCoMan supports *semantic rules* to define policies. These policies will allow users to share their location to the right users, at the right granularity, at the right place, and at the right time. Using location policies, users will be able to manage their privacy independently of the applications:

1) *hiding* their locations to other persons;
2) *masking* their locations with fictitious positions;
3) establishing the *granularity* at which they want to be located;
4) defining the level of *closeness* accepted to be located.

Manuscript received September 10, 2013; revised December 19, 2013; accepted December 29, 2013. Date of publication January 21, 2014; date of current version August 23, 2016. This work was supported in part by the Spanish Ministry of Science and Innovation (MICINN) through Project RECLAMO: Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management under Grant TIN2011-28287-C02-02 and through Project Mejora de Arquitectura de Servidores, Servicios y Aplicaciones under Grant TIN2012-38341-C04-03, by the European Commission through the European Regional Development Fund, and by the Séneca Foundation through the Funding Program for Research Groups of Excellence under Grant 04552/GERM/06.

A. Huertas Celdrán, M. Gil Pérez, and G. Martínez Pérez are with the Department of Information and Communication Engineering, University of Murcia, 30100 Murcia, Spain (e-mail: alberto.huertas@um.es; mgilperez@um.es; gregorio@um.es).

F. J. García Clemente is with the Department of Computer Engineering and Technology, University of Murcia, 30100, Murcia, Spain (e-mail: fgarcia@um.es).

Digital Object Identifier 10.1109/JSYST.2013.2297707

SeCoMan also manages authorization policies to control who can access (or stay in) a given space. The IoT information is provided by certain location systems and middleware, which are independent to the framework. This independence allows SeCoMan to choose the location systems and middleware depending on the characteristics of the environment.

The remainder of this paper is organized as follows. In Section II, we discuss the related work regarding other context-aware solutions. Section III presents the SeCoMan architecture, whereas the collection of ontologies managed by SeCoMan is described in Section III-B. Taxonomies of policies and queries are presented in Sections IV and V, respectively. Section VI shows the deployment of a smart application making use of SeCoMan that offers advanced services in a supermarket scenario. Section VII reports some experimental results to illustrate the performance of the SeCoMan framework. A thorough discussion comparing our approach with other related systems is performed in Section VIII, and finally, conclusions and future works are drawn in Section IX.

II. Related Work

The large number of objects involved in the IoT makes organization, representation, storage, and sharing is a potentially challenging task. In such a context, “semantic-oriented” IoT visions are available in the literature to provide modeling solutions for things description, reasoning over data generated by the IoT, semantic execution environments, and architectures that accommodate IoT requirements [3], [4]. A common ontology is a key factor to develop context-aware systems and smart applications, as it allows knowledge sharing between independent systems and uses semantic reasoning about the context to offer advanced services to customers.

A recent publication conducted a depth survey on context-aware systems oriented to the IoT, where a large number of solutions are analyzed by considering different topics [3]. Considering the semantic-oriented IoT vision, systems like Feel@Home [5], Hydra [6], CroCo [7], SOCAM [8], and CoBrA [9] provide support to “security and privacy” features, as our solution. Feel@Home is a context-aware framework that supports communications between contexts or domains, considering intra- and interdomain interactions. Hydra is an ambient intelligence middleware system oriented to the IoT, which integrates the device, semantic, and application contexts to offer context-aware information. On the other hand, CroCo is a cross-application context management service for heterogeneous environments, whereas SOCAM uses a collection of ontologies that shapes the quality, dependence, and classification of the context information. This collection is built on a common upper ontology for all contexts, as well as for domain-specific ontologies that define concepts of each one. Another related work in this context, which is not included in the survey presented in [3], is CoCA [10]. This proposal presents a collaborative context-aware service platform, where a neighborhood-based mechanism to share resources is introduced. CoCA infers users’ location by considering information about the context and the location of the elements.

Four of the five solutions described make use of semantic rules for different purposes, being Feel@Home the only one that does not. Hydra, CroCo, and SOCAM do use semantic rules to infer new information about a given context, taking into account information from others. Instead, CoCA makes use of semantic rules to manage the ontologies, e.g., a property is the inverse of another property, as well as additional information about the domain. Yet, none of these four solutions uses semantic rules to define policies oriented to protect users’ privacy preferences. Users’ privacy should be supported by any context-aware framework, with which the users are capable of dynamically restricting or disclosing information to others depending on their location and their preferences in terms of privacy. Consequently, the current trend in context-aware systems focuses on controlling the disclosure of users’ location by using policies.

There are a number of systems based on Semantic Web that manage policies to preserve users’ privacy. For example, CoBrA presents a context-aware architecture that allows distributed agents to share information with each other. CoBrA defines an ontology that shapes spaces composed of smart agents, devices, and sensors, and protects the privacy of its users by using rules that deduce whether they have the right permissions to share and/or receive information. Another example is PPCS [11], where a semantically rich policy-based framework with different levels of privacy to protect users’ information in environments with mobile devices is presented. Dynamic information observed or inferred from the context, along with static information about the owner, is taken into account to make access control decisions. Location and context information of the users are shared (or not) depending on their privacy policies. Another proposal supporting privacy policies without using Semantic Web technologies is CoPS [12]. In CoPS, users can control who can access their context data, when, and at what level of granularity. It organizes policies into different hierarchical levels, defining a default policy according to an optimistic or pessimistic approach.

Despite the work and progress made by the systems discussed, a lot of work is still required to improve key aspects, such as policies and context management, users’ privacy, availability and quality of services, and robustness. In Section VIII, we thoroughly discuss and compare our framework with others that also manage users’ privacy through policies.

III. SeCoMan Architecture

SeCoMan is a trusted third party that manages users’ privacy about their location. It supplies the context and space information provided by the IoT to smart applications that could be not reliable enough for managing this information. The SeCoMan architecture is composed of three layers to allow framework actors to manage the resources and develop applications more efficiently. Fig. 1 shows the components and actors forming the multilayered architecture of SeCoMan.

A. Actors

We defined three kinds of actors to interact with SeCoMan. First, the *Framework Administrator* manages the common

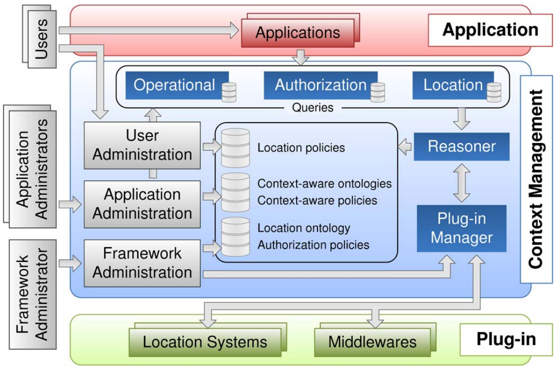


Fig. 1. Overview of the multilayered architecture of SeCoMan.

resources for all contexts, among which we emphasize the management of the *Location ontology* that models the space information and the definition of the *Authorization policies*. Furthermore, this administrator registers the smart applications that can make use of SeCoMan and indicates to the *Plug-in manager* module the location systems and middleware that has to be used to receive the space and context information.

On the other hand, each *Application Administrator* is in charge of managing the context of his/her own applications, handling the *Context-aware ontologies and Policies* as well as more sophisticated queries by combining the *Operational*, *Authorization*, and *Location* ones. Finally, the last kind of actor is *Users*. They are persons who use the smart applications to obtain information about the environment in which they are located. They define their location policies to manage their privacy directly in the framework without having to rely upon the applications. For that reason, our framework acts as a trusted third party for the users as applications might not be reliable enough with the location information that they manage.

B. Layers of the SeCoMan Architecture

The three layers composing the SeCoMan architecture shorten the complexity of the IoT infrastructures and provide the necessary resources for the previous actors; therefore, they can manage smart applications, the space and context information, and the plug-ins for the location systems and middleware. First, the *Application* layer contains smart applications that provide users with specific information about the spaces in which they are located. To that end, the *Applications* will make queries to the *Context Management* layer in order to obtain the space and context information desired by the users.

In order to manage the context, the Context Management layer uses ontologies to shape the information gathered from the *Plug-in* layer, the semantic rules to define the policies that control the system behavior, and the semantic reasoning to infer new knowledge, taking into account the previous information sources. To perform all these tasks, the *Operational*, *Authorization*, and *Location* modules provide smart applications with a number of queries predefined in the framework, which offer certain information regarding these topics. Queries are applied on the new knowledge inferred by the *Reasoner* module. This takes as input the ontological model, formed by the union of the ontologies updated according to the information collected by the *Plug-in Manager* module, and the semantic rules defined by the actors through their corresponding administration components.

Finally, the Plug-in layer obtains the space and context information about the elements that form part of the environment and their locations, as well as further information from these elements depending on the environment. This is composed of different plug-ins that interact, on one hand, with the *Middleware* (which in turn communicate with sensors or other devices to receive context information) and, on the other hand, with the *Location Systems* to obtain information about the space. This layer provides independence to SeCoMan with regard to the location system used, thus allowing Application Administrators to choose the best location system or middleware depending on the characteristics of the environment.

We describe here the main ontology managed by SeCoMan and an example about a supermarket scenario, which will be used through this paper to introduce all concepts related to our proposal. Using this scenario, we implemented a smart application offering location-based services to customers.

C. Location Ontology

SeCoMan defines a collection of ontologies to shape the space and context information. This collection is composed of an ontology called *Location* that models the indoor location, common for all contexts, and a set of ontologies for the smart applications that provide specific services in different contexts. Fig. 2(a) shows the Location ontology. This ontology models the space and provides a set of primitives with which to describe regions of the space and relationships among them.

The Location ontology is categorized into three different but related topics: element, authorization, and space. The top-level class in the element topic is *Element*, which refers to any entity that forms part of the environment (persons or objects). Elements can have several *Roles* and *Privileges* that can be used to provide personalized information. Note that Privilege is the most important class in the authorization topic. Privileges are used to allow Elements to perform certain actions, such as staying in a specific position. The Element class has two predefined subclasses, *System* and *Person*, which are defined to be disjointed. A Person defines the accuracy on the granularity and closeness at which he/she wants to release his/her location by using the *Accuracy*, *Granularity*, and *Closeness* classes. Finally, and in order to support location generalization, the Location ontology uses a hierarchical model for location. *Space* is the top-level class in this model, having five predefined subclasses, namely (from low to high accuracy): *Building*, *Floor*, *Area*, *Section*, and *Position*. Position establishes the *Geographical* or *Absolute Position* of an element, where several Positions form a Section that has two predefined subclasses, i.e., *Corridor* and *Room*.

The Location ontology entities are related each other by properties. A portion of these properties is used to establish new relationships through policies. For example, authorization policies use the *hasAuthzAccess* property to link Persons and

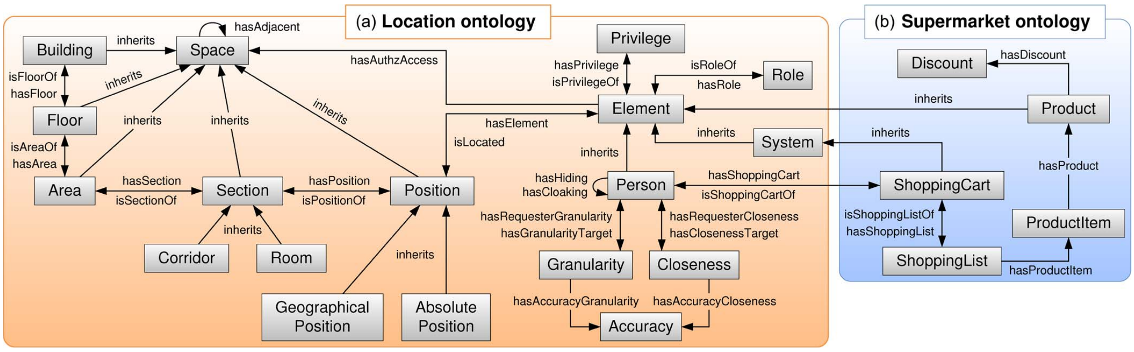


Fig. 2. Ontologies of SeCoMan. (a) *Location* ontology, common for all contexts. (b) *Supermarket* ontology shaping a specific context.

Spaces according to the persons' privileges; location policies generate the *hasCloaking* and *hasHiding* properties between two persons to mask and hide their location, respectively; and *hasRequesterGranularity* and *hasRequesterCloseness* are established by the location policies to link Persons to each other with a specific Granularity and Closeness.

D. Motivating Example

As a proof of concept to show the ease of adding new ontologies to SeCoMan, we implemented a smart application called *eCoMarket* (further details in Section VI). It offers a smart service to customers of a supermarket according to their location. eCoMarket defines an ontology called *Supermarket*, shown in Fig. 2(b), that shapes the supermarket context.

The top-level class in the Supermarket ontology is *Product*, representing an article of the supermarket. This class inherits from Element of the Location ontology, so that this relation is the connection between both ontologies. As it can be observed, a new ontology only has to inherit from the Element class of the Location ontology, or from one of its subclasses, in order to create a link between the two ontologies. In the supermarket context, Products can have *Discounts*, and some of them can belong to a *ShoppingCart* through a *ShoppingList* that contains one or more *ProductItems*.

For clarity, Fig. 3 shows a graphic representation of a basic example about a given instance of a supermarket, in order to clearly follow all elements introduced here. It is worth noting that we defined the corresponding data properties for all classes in the two ontologies shown in Fig. 2. For example, the name and the price of any article of the supermarket were modeled as data properties in the Product class. Although they were not drawn in Fig. 2 for simplicity, the complete definition of both ontologies—classes and object and data properties—can be accessed and downloaded from [13].

In this example, we created entities of the classes defined in the Location and Supermarket ontologies. In this sense, we have a supermarket with one Floor, two Areas, two Corridors, and five Positions. At the supermarket place, there are four persons who can use different Roles, Privileges, Granularities, Closeness, and Accuracies. Specifically, *Peter* has the *GoldCustomer* and *Hidden* roles (R), and he is located at *Position1*. This position belongs to the *ComputersCorridor*, which is in turn located at *ElectronicsArea*. Peter has a ShoppingCart containing a *Laptop*. On the other hand, *Natalie* is located at *Position5* and she has the *OpenCashRegister* privilege (P). Position5 belongs to the *PaymentCorridor* that is located at *PaymentArea*. *Andy* has the *AnonymChild* role, and he is located at *Position4* belonging to PaymentCorridor. Finally, *Margaret* is located at *Position2*, which belongs to the ComputersCorridor, and she has granularity (G) of Area and closeness (C) of Floor. Furthermore, the supermarket has two products: a *Laptop* whose price is 1000€ and is located at Position1; and a *Tablet* whose price is 400€ and is located at *Position3*, belonging to two different corridors and areas.

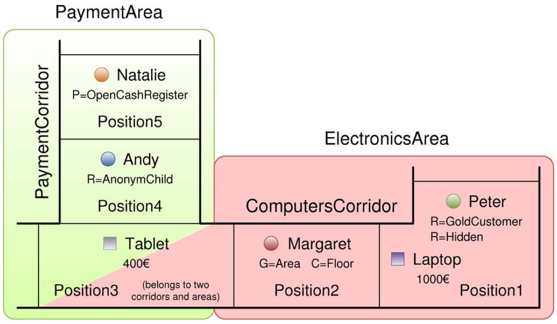


Fig. 3. Map of the supermarket example.

IV. SeCoMan Policies

Our framework dynamically controls users' privacy, their authorization to stay in certain spaces, and the context information, and generates new knowledge by using semantic rules, which form policies. SeCoMan uses rules that consist of two lists of predicates: the antecedent and the consequent parts of a rule. If all predicates of the antecedent part take the Boolean value true, all predicates in the consequent part are evaluated. It is important to know that, in our semantic rules, the predicates in the consequent part establish new relationships between entities of the ontologies and does not generate new entities.

The policies in SeCoMan are composed of the following elements: *Type* is the kind of policy; *Maker* is the person who defines the policy (possibly being the same as the Target); *Target* is the person whose information is managed by the policy; *Requester* is the person, or group of persons, who request information about the Target; *Place* is the region of the environment in which the policy is applied; and *Result* determines the relationship that the Requester will have about the Target information. Note that Result is the consequent part of the semantic rule, whereas the remaining fields belong to the antecedent part. SeCoMan also illustrates the possibility of including extensions to make richer and more powerful policies. For example, Makers may improve their policies by tuning elements like *Role*, *Privilege*, *Date*, or *Context*.

In the context of SeCoMan, our framework architecture manages three kinds of policies: *Operational policies*, defined by the Application Administrators to generate new knowledge related to the users' context; *Authorization policies*, defined by the Framework Administrator to decide the authorization of the users to stay or not in a specific space; and *Location policies*, defined by application-independent Users to specify the privacy preferences about their location. We show below an example for each of these policies, making use of the supermarket scenario defined in Section III-D.

A. Operational Policies

Operational policies are used to manage the information of the smart applications, generating new knowledge related to the context-aware ontologies. The antecedent part of the policy is composed of entities belonging to the collection of ontologies of SeCoMan, whereas the consequent one establishes relationships between two entities of which, at least, one of them has to belong to the Location ontology. In the supermarket example, let us suppose that the Application Administrator of the eCoMarket application ($Maker$) defines that "On July 2013, Persons who have the GoldCustomer role will obtain a 21% off in products located at ComputersCorridor," i.e.,

$$\begin{aligned}&Person(?target)\\&\wedge isLocated(?target, \#ComputersCorridor)\\&\wedge hasRole(?target, \#GoldCustomer)\\&\wedge Product(?product)\\&\wedge isLocated(?product, \#ComputersCorridor)\\&\wedge greaterThan\,(\#Today, date(2013, 06, 30))\\&\wedge lessThan\,(\#Today, date(2013, 08, 01))\\&\rightarrow hasDiscount21(?target, ?product).\end{aligned}$$

Applying this rule to the supermarket use case, Peter ($Target$) gets a 21% off ($Result$) on July 2013 in the Laptop he has in his ShoppingCart and in the Tablet located at Position3 as Peter has the GoldCustomer role and both products belongs to ComputersCorridor (see Section III-D).

B. Authorization Policies

Authorization policies are based on privileges to allow users to stay in certain locations according to their privileges [14]. By default, SeCoMan denies the authorization in the absence of rules. These policies are independent of the context; therefore, the consequent part in this kind of rule only generates relationships between entities belonging to the Location ontology (common for all contexts). In the supermarket example, let us suppose that the Framework Administrator ($Maker$) defines that "Persons located at PaymentArea with the OpenCashRegister privilege have authorized access to be there," i.e.,

$$\begin{aligned}&Person(?target) \wedge isLocated(?target, \#PaymentArea)\\&\wedge hasPrivilege(?target, \#OpenCashRegister)\\&\rightarrow hasAuthzAccess(?target, ?position).\end{aligned}$$

Applying this rule to the supermarket use case, Natalie ($Requester$ and $Target$) has authorized access ($Result$) to stay at PaymentArea, as she has the OpenCashRegister privilege.

C. Location Policies

Location policies generate new knowledge related to users' location privacy, these being independent of the context. Location policies are divided into four groups: $Cloaking$, $Hiding$, $Granularity$, and $Closeness$. In this kind of policy, the $Target$ is the same person as the $Maker$; therefore, he/she can define rules for the same requester specifying different roles, locations, dates, or times. We explain below in detail these four kinds of location policies managed by the framework.

1) Cloaking: Masking or cloaking is intended to generate one or more fictitious positions for a particular user; therefore, other users cannot distinguish the real position where the target is located. As an example, a cloaking policy applied to the supermarket scenario could be as follows:

$$\begin{aligned}&Person(\#Andy) \wedge hasRole(\#Andy, \#AnonymChild)\\&\wedge isLocated(\#Andy, \#PaymentArea)\\&\wedge greaterThan\,(\#Now, time(08, 59))\\&\wedge lessThan\,(\#Now, time(14, 00)) \wedge Person(\#Peter)\\&\rightarrow hasCloaking(\#Andy, \#Peter).\end{aligned}$$

Applying this rule to the supermarket use case, if Peter ($Requester$) asks about the position of Andy ($Target$) between 9:00 A.M. and 2:00 P.M., SeCoMan will generate one or more masking positions ($Result$) for Andy. This is due to Andy being at PaymentArea and he has the AnonymChild role. This provokes that Peter cannot distinguish if Andy is at Position4 (his real position) or at a fictitious one, such as Position3, for example.

2) Hiding: Users can define hiding policies when they do not want to release their location to others, thereby avoiding that requesters know the position of the target. A hiding policy example is shown in the following for the supermarket scenario:

$$\begin{aligned}&Person(\#Peter) \wedge hasRole(\#Peter, \#Hidden)\\&\wedge isLocated(\#Peter, \#ComputersCorridor)\\&\wedge Person(?requester)\\&\rightarrow hasHiding(\#Peter, ?requester).\end{aligned}$$

Applying this rule to the supermarket use case, if someone $(Requester)$ asks about Peter's location $(Target)$, SeCoMan will not return his location $(Result)$, as Peter is at ComputersCorridor and he has the Hidden role. This position is hidden, as requested by Peter, through the $hasHiding$ property.

3) Granularity: Granularity policies are used to indicate the maximum accuracy at which users want to be located. As stated in Section III-B, there are various levels of granularity that can be applied to a given location, namely: $Position$, $Section$, $Area$, $Floor$, and $Building$. An example of policy of this type in the supermarket scenario could be as follows:

$$\begin{aligned}&Person(\#Margaret) \wedge Person(?requester)\\ &\wedge Granularity(?granularity)\\ &\wedge hasGranularityTarget(\#Margaret, ?granularity)\\ &\wedge hasAccuracyGranularity(?granularity, \#Area)\\ &\rightarrow hasRequesterGranularity(?granularity, ?requester).\end{aligned}$$

Applying this rule to the supermarket use case, nobody $(Requester)$ is able to know that Margaret $(Target)$ is at Position2, because she has Granularity of Area. Therefore, other users can only know that Margaret is located at ElectronicsArea $(Result)$ as she does not want to be located with a Granularity below Area (established through $hasRequesterGranularity$).

4) Closeness: Closeness policies are defined to indicate the minimum level of nearness at which persons want to be located. Nearness levels correspond to the same values defined for the granularity policies. An example of this kind of policy, applied to the supermarket scenario, is given by

$$\begin{aligned}&Person(\#Margaret) \wedge Person(?requester)\\ &\wedge Closeness(?closeness)\\ &\wedge hasClosenessTarget(\#Margaret, ?closeness)\\ &\wedge hasAccuracyCloseness(?closeness, \#Floor)\\ &\rightarrow hasRequesterCloseness(?closeness, ?requester).\end{aligned}$$

Applying this rule to the supermarket use case, if someone $(Requester)$ wants to know who is in adjacent positions, corridors, areas, or floors, he/she will not know that Margaret $(Target)$ is located at his/her Floor, as she established Floor as her maximum level of closeness to be located $(Result)$ through the $hasRequesterCloseness$ property.

V. SeCoMan Queries

This section presents a set of queries allowing smart applications to provide the space and context information to their customers. Customers will be able to obtain such information, but they cannot define their own queries in order to avoid that they gain private information from others. Queries consider the information shaped in the SeCoMan ontologies described in Section III-B and the policies defined in Section IV.

To define the space queries in SeCoMan, we used the four categories of the taxonomy defined in [15], namely: *position* or *location*, *range*, *nearest neighbor* or *closeness*, and *navigation*. In addition, SeCoMan also provides specific information of the environment and authorization decisions about users to stay in a place through operational and authorization queries, respectively. We describe in detail in the following the six queries predefined in our framework, ending with a way of defining queries composed by the Application Administrators in order to provide advanced features to their smart application(s).

A. Operational Queries

Operational queries allow Users to get information related to them and the environment in which they are located, taking into account the operational policies defined in Section IV-A. Continuing with the supermarket example of Section III-D, we show in the following a function that provides information about the products contained in the requester's shopping cart.

```
1. productInfoList shoppingCartProducts(Person
   requester) {
2.   cart ← SupermarketOnt.hasShoppingCart(requester)
3.   productList ← SupermarketOnt.hasShoppingList(cart)
4.   for (Product product : productList) {
5.      name ← SupermarketOnt.hasId(product)
6.      amount ← SupermarketOnt.hasAmount(product)
7.      price ← SupermarketOnt.hasPrice(product)
8.      discount ← SupermarketOnt.hasDiscount(requester,
        product)
9.      price ← price * discount
10.     productInfoList.add(name, amount, price)
11.  }
12.  return productInfoList
13. }
```

We implemented some methods in an external class, called *SupermarketOnt*, in order to gather information shaped in the Supermarket ontology. The *shoppingCartProducts* function receives the requester's shopping cart and the information about its products (lines 2–7). The operational policies are then taken into account to apply discounts (line 8) for each of the products deposited in the shopping cart.

Applying this query to the supermarket use case, if Peter (*Requester*) wants to know the information about the products contained in his cart, he will get that it holds a Laptop whose price is 790€ (*Result*). Although the price of the Laptop is 1000€ (see Section III-D), Peter has a 21% off when applying the operational policy defined in Section IV-A.

B. Authorization Queries

Queries related to authorization allow Users and the Framework Administrator to know if the customers have authorization or not to stay in a given space. SeCoMan offers the *authorizationAccess* and *unauthorizedPersons* functions to obtain authorization information.

The *authorizationAccess* function, defined in the following, provides Users with information about the authorization to stay in their position. It receives as parameter the *Person* who requests the information about himself/herself and returns an authorization response (allowed or denied). Note that this and subsequent functions will make use of some methods implemented in an external class called *LocationOnt*, which provides information shaped in the Location ontology. The authorizationAccess function obtains the requester's space (line 2) and will return *allowed* or *denied* depending on whether the requester has access to stay in his/her current space (see Section IV-B).

```
1. authorizationResponse authorizationAccess(Person
   requester) {
2.     space ← LocationOnt.hasPosition(requester)
3.     authorization ← LocationOnt.hasAuthzAccess
       (requester, space)
4.     if (authorization == true)
5.         return allowed
6.     return denied
7. }
```

Considering the supermarket use case, if Natalie (*Requester*) asks about her authorization to stay where she is, at PaymentArea, the response will be *allowed*. This is due to the authorization policy, defined in Section IV-B, allowing Natalie to stay there as she has the OpenCashRegister privilege.

As opposed to the previous query, the *unauthorizedPersons* function, defined below, provides the list of persons without authorization to stay in a given space. The goal behind this function is to avoid that persons from hiding their position when they are in unauthorized spaces, not having into account the policies that defined them. The unauthorizedPersons function receives as parameter the *Space* in which the requester is interested and returns the list of unauthorized persons staying there.

```
unauthorizedPersonList unauthorizedPersons(Space
space) {
    personList ← getElements(space, "Person")
    unauthorizedPersonList ← emptyList
    for (Person target : personList)
        if (!LocationOnt.hasAuthzAccess(target, space))
            unauthorizedPersonList.add(target)
    return unauthorizedPersonList
}
```

Considering the supermarket use case, if the Application Administrator (*Requester*) asks about the list of unauthorized persons located at PaymentCorridor, he/she will obtain that Andy does not have authorization because he does not have the OpenCashRegister privilege. Natalie does not appear in that list because she does have such a privilege.

C. Location Queries

In SeCoMan, location queries can be used by Users to get the elements' position that form part of the environment, taking into account the policies defined in Section IV-C.

As an example, we defined the *elementLocation* function (given in the following), which returns a list of spaces in accordance with the two parameters received: the *Person* who requests the information, and the *Element* about which the requester is interested to obtain its position. It first checks the type of element of the target (line 2). If the target is a Person (line 3), the function checks whether he/she has a hiding, cloaking, or granularity policy with the requester. Hiding policies will return an empty list (lines 4 and 5), whereas the maximum accuracy at which the target wants to be located will be established by invoking the *getPositionsApplyingGranularity* function (lines 7 and 8). Instead, if there is a cloaking policy, the *getPositions ApplyingCloaking* function will return the real position of the target, as well as some fictitious positions to mask the former one (lines 10 and 11). In case the target is an object, no policy is applied (lines 14–16).

```
1. spaceList elementLocation(Person requester, Element
   target) {
2.    switch (LocationOnt.elementType(target)) {
3.      case "Person":
4.        if (LocationOnt.hasHiding(target, requester))
5.          spaceList ← emptyList
6.        else {
7.          if (LocationOnt.hasGranularityTarget(target,
            requester))
8.            spaceList ← getPositionsApplying
              Granularity(target, requester)
9.          else spaceList ← LocationOnt.hasPosition(target)
10.           if (LocationOnt.hasCloaking(target, requester))
11.             spaceList ← getPositionsApplying
                Cloaking(spaceList, target)
12.         }
13.         break
14.       case "Object":
15.         spaceList ← LocationOnt.hasPosition(target)
16.         break
17.     }
18.     return spaceList
19. }
```

Applying this query to the supermarket use case, if Peter (*Requester*) wants to know where Andy (*Target*) is, Peter will obtain that Andy has two locations, Position4 and Position3. This result is due to Andy having a cloaking policy, as defined in Section IV-C1, that returns Position3 as fake position.

D. Range Queries

Range queries can be used to identify all elements placed at a location meeting a certain criteria. As the previous one, these also consider hiding, cloaking, and granularity policies.

As an example, we defined the *rangeSpaceElements* function (given in the following), which provides the elements placed at a given space. This function returns a list of elements in accordance with the three parameters received: the *Person* who requests the information, the *Space* in which the requester is interested, and the *ElementType* that the requester wants to obtain. This function checks if the type of element is a Person (line 3), and if so, the function considers his/her policies; otherwise, the elements contained in the space are returned without applying any policy (lines 10–12).

```
1. rangeElementList rangeSpaceElements(Person
   requester, Space space, ElementType elementType) {
2.   switch (elementType) {
3.     case “Person”:
4.       rangeElementList ← emptyList
5.       elementList ← getElements(space, elementType)
6.       for (Person target : elementList)
7.         if (!LocationOnt.hasHiding(target, requester) &&
           !LocationOnt.hasCloaking(target, requester) &&
           (getGranularity(target, requester) <= space))
8.           rangeElementList.add(target)
9.       break
10.    case “Object”:
11.      rangeElementList ← getElements(space,
         elementType)
12.      break
13.  }
14.  return rangeElementList
15. }
```

Considering this query in the supermarket use case, if Andy (*Requester*) wants to know the Persons (elementType) located at ComputersCorridor (*Space*), he will obtain that nobody is located there. This result is due to Peter defining a hiding policy (see Section IV-C2) and Margaret having a granularity policy with a Granularity more than Area (see Section IV-C3).

E. Closeness Queries

Closeness queries can be used to find the nearby elements to Persons with a given level of proximity. The *hasAdjacent*, *hasPosition*, and *isPositionOf* properties, defined in the Location ontology [see Fig. 2(a)], aim to provide neighborhood and hierarchical relationships. This kind of query takes into account the policies defined by the target.

As an example, we defined the *closeElements* function (given in the following), which returns a list of nearby elements to the requester in accordance with the three parameters received: the *Person* who performs the query, the *Accuracy* indicating the proximity level at which the requester wants to get the elements, and the *ElementType* that the requester wants to obtain. This function invokes the *getAdjacentSpaces* function to retrieve the adjacent spaces to the requester's location (line 3). Then, if the type of the desired element is a Person (line 5), the function obtains the persons located at the spaces previously obtained (line 6) and applies their policies. Otherwise, if it is an object (line 11), the elements close to the requester are obtained without considering any policy (lines 11–13).

```
1. closeElementList closeElements(Person requester,
   Accuracy accuracy, ElementType elementType) {
2.   closeElementList ← emptyList
3.   spaceList ← getAdjacentSpaces(requester, accuracy)
4.   switch (elementType) {
5.     case “Person”:
6.       elementList ← getElements(spaceList, elementType)
7.       for (Person target : elementList)
8.         if (!LocationOnt.hasHiding(target, requester) &&
           !LocationOnt.hasCloaking(target, requester) &&
           (getCloseness(target, requester) <= accuracy))
9.           closeElementList.add(target)
10.        break
11.      case “Object”:
12.        closeElementList ← getElements(spaceList,
           elementType)
13.        break
14.  }
15.  return closeElementList
16. }
```

Considering this query in the supermarket use case, if Natalie (*Requester*) wants to know who is close to her with proximity of Corridor, she will obtain that Andy, located at PaymentCorridor, and Peter, located at ComputersCorridor, are close to her. Instead, Margaret does not appear in that list because she defined a closeness policy with a Closeness level of Floor, as defined in Section IV-C.4.

F. Navigation Queries

Navigation queries allow Users to find the path leading to the desired place or element. If the destination is a Person, his/her privacy policies are taken into account to get the path. This kind of query also considers the same policies as the ones required by the location queries (defined in Section V-C).

As an example, we defined the *getMinimumPaths* function (given in the following), which provides the list of spaces to reach the target from the requester's location in accordance with the two parameters received: the *Person* who wants to go from his/her current position to the destination position, and the *Element* indicating the destination of the path. This function obtains the spaces of the source and the destination invoking the *elementLocation* function (lines 2 and 3), as defined in the location queries of Section V-C. Once having the spaces, the function invokes the *pathFinder* function for each destination (line 6). pathFinder is a recursive function that checks if the source and the destination are in the same space. If so, *pathFinder* will return the response; otherwise, it recursively calls itself using each adjacent space to the source as the next unvisited source, keeping track of paths to avoid cycles. Finally, the path is returned to the user (line 7), if any.

```
1. pathList getMinimumPaths(Person requester, Element
   destination) {
2.    sourcePositionList ← elementLocation(requester,
      requester)
3.    destinationPositionList ← elementLocation(requester,
      destination)
4.    pathList ← emptList
5.    for (Position destinationPosition : destinationPosition
      List)
6.       pathList.add(pathFinder(sourcePositionList[0],
         destinationPosition))
7.    return pathList
8. }
```

Applying this query to the supermarket use case, if Peter (*Requester*) wants to know the path to go from his current position (Position1) to the Andy's position (Position4), getMinimumPaths will return to him a list of spaces with two alternatives as Andy has a cloaking policy (where Position3 is a fake position generated in Section IV-C1): $\langle Position1, Position2, Position3, Position4\rangle$ and $\langle Position1, Position2, Position3\rangle$. Another example is the case when Andy wants to know the path to go from his current position (Position4) to the Margaret's position (Position2). The response will be $\langle PaymentArea, ElectronicsArea\rangle$, as she holds a Granularity of Area (defined in Section IV-C3).

G. Composed Queries

Application Administrators can define more sophisticated queries by combining some of those described earlier and by subsequently filtering their output. Therefore, the output of a query is the input for the next one.

As an example, we defined the *complexPathsToOffers* function (given in the following), which provides routes to products on offer nearby the requester's current location, without going through the position where a given person is located. This function returns a list of minimum paths in accordance with the three parameters received: the *Person* who requests the information, the *Space* where the requester wants to get the products with certain discounts, and another *Person* to whom the requester wants to avoid in the path. complexPathsToOffers invokes the *elementLocation* function to obtain the user's location(s) to be avoided (line 3), and then obtains the products with discounts placed at the desired space by taking into account the operational policies defined in Section IV-A (line 4). For each product, the *getMinimumPathAvoidingPosition* function is invoked (line 6) to obtain the minimum path from the requester's location to the product without going through the position(s) where the unwanted user is located.

```
1. pathList complexPathsToOffers(Requester requester,
   Space space, Person avoidPerson) {
2.    pathList ← emptyList
3.    avoidPositionList ← elementLocation(requester,
      avoidPerson)
4.    productsOnOffer ← getProductsOnOffer(requester,
      space)
5.    for (Product product : productsOnOffer)
6.       pathList.add(getMinimumPathAvoidingPosition
         (requester, avoidPositionList, product))
7.    return pathList
8. }
```

Applying this query to the supermarket use case, consider that Peter (*Requester*) wants to know the minimum path(s) to products with some discount and located at the same Floor (*Space*), without having to go through the Andy's position. The function responses that there is a Laptop at the Peter's position (Position1), and there is no possible way of going to the Tablet article without passing through the Andy's position. This is due to the cloaking policy of Andy defined in Section IV-C1.

VI. Deployment of a Context-Aware Smart Application

We developed a smart application, called *eCoMarket*, that offers advanced services in supermarkets to validate the proper functioning of SeCoMan. Furthermore, the deployment of this application was also performed for measuring the throughput and scalability of SeCoMan. These results are subsequently presented in Section VII.

The eCoMarket application provides customers (Users) with the products' location and their information, the position of shopping carts and information about their products, products on offer, nearby friends with several levels of granularity, the path to reach people or products, the customers' authorization to stay at a given place, and privileges and roles of customers. Customers of the supermarket will be able to obtain previous information using an Android application that interacts with the eCoMarket application using the REST technology. With REST, users can use devices with limited computing resources to make their requests as such devices will only have to handle queries, receive responses, and then display them. On the other hand, remote method invocation is used to separate the Application and Context Management layers, thus balancing the workload across multiple computers in order to avoid bottlenecks, among others.

Semantic rules that form policies are expressed in Semantic Web Rule Language (SWRL) [16]. SWRL includes a type of axiom, called Horn clause logic, of the form $if \ldots\ then \ldots$, and it is the most used in Semantic Web. The space and context information is shaped in the Location and Supermarket

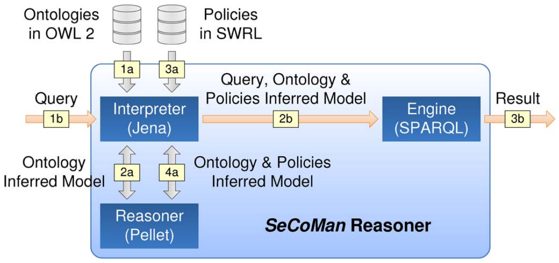


Fig. 4. Reasoning and query processes.

ontologies, respectively. Both of them are defined in Web Ontology Language (OWL 2) [17] and have been generated with the Protégé tool [18]. We have chosen OWL 2 rather than other languages, such as Resource Description Framework (RDF), RDF Schema, or DARPA Agent Markup Language+Ontology Interchange Language (DAML+OIL), because OWL 2 is more expressive than the rest. It was specifically designed as an ontology language, it is an open standard, and it is the main ontology language used nowadays in Semantic Web.

In order to infer new knowledge, all processes related to the *Reasoner* module of SeCoMan (shown in Fig. 1) are depicted in Fig. 4. The Reasoner component uses Pellet [19], which receives ontological models generated by the *Interpreter* component and returns inferred models with new knowledge. The Interpreter uses the Jena API [20] to generate ontological models with the information shaped in the ontologies and policies. Finally, the *Engine* component is in charge of translating the queries performed by the users into SPARQL queries [21], which are applied to the inferred model to get the result.

The Interpreter generates an ontological model from the Location and Context-aware ontologies (step 1a in Fig. 4). This model is sent to the Reasoner to obtain the inferred model with new information inferred from the ontologies (step 2a). Once the Interpreter receives the inferred model, it updates it with the new information provided by the Reasoner according to the policies defined in the system (steps 3a and 4a). Note that the previous process is made when new information from the environment is provided. Therefore, when queries are performed, SeCoMan always has available a consistent and updated inferred model, thus avoiding users having to wait the reasoning time shown later in Section VII-A. When users make a query (step 1b), the Interpreter invokes the *Engine* with the latest inferred model and the query (step 2b). The Engine component applies the appropriate SPARQL queries to the inferred model and returns the result (step 3b).

We developed two plug-ins in SeCoMan to obtain the space and context information. The first one obtains the space information through a REST client, which communicates with an indoor location system based on Wi-Fi. This location-based system obtains the environment map and the location of its elements, combining the fingerprinting technique with pictures about the environment [22]. The second plug-in obtains the context information through a REST client, which communicates with a radio-frequency identification (RFID) middleware [23]. This middleware is capable of getting the information of the products contained on the shopping carts. The space and context-aware information are provided to the Context Management layer by using REST.

TABLE I
INDIVIDUAL DISTRIBUTION OF POPULATION

Element	Amount	Percentage	Element	Amount	Percentage
Buildings	1	0.05%	Persons	200	8.2%
Floors	4	0.2%	Roles	10	0.6%
Areas	20	0.8%	Privileges	40	1.2%
Sections	80	3.4%	Others	90	3.55%
Positions	2,000	82%	**Total**	**2445**	**100%**

VII. Experimental Results

We conducted some experiments with the aim of measuring the throughput and scalability of our SeCoMan proposal. These experiments were intended to deal with three questions.

1) Is the computing time of reasoning acceptable?
2) How does it scales with different amount of information, such as the number of individuals and policies?
3) How does the query time varies when taking into consideration the previous premises?

As experimental setting, the SeCoMan framework and the conducted tests were carried out in a dedicated PC with an Intel Core i7-3770 3.40-GHz, 16-GB of RAM, and an Ubuntu 12.04 LTS as its operating system. The results shown in this section have been obtained by executing the experiments 100 times and computing their arithmetic mean.

A. Reasoner Performance

The Reasoner is an important part of SeCoMan as it *greatly* affects to the framework performance. In order to check the reasoning time and its scalability, several experiments were conducted. A way to measure the SeCoMan performance is making executions with different complexity. This complexity is related to the number of statements hold in the knowledge base, which depends on the number of individuals present in the ontology and the number of semantic rules that form the policies. Increasing the number of individuals and semantic rules will provoke an increment on the number of statements and thus on the complexity of the executions.

The number of individuals contained in our ontologies is referred as *population*. This was randomly generated for the experiments, but in a controlled way, in order to achieve the desired distribution for simulating a scenario as real as possible. Table I depicts the number of elements used in our environment and the percentages obtained for them.

Another issue to evaluate the Reasoner scalability is the way in which the population sizes are established. In this sense, we defined an initial population of 15 000 individuals, and we increased this population with other 15 000 individuals in each step. In order to show the complexity of our ontology, Table II shows the relationships between the individuals and the statements generated by the Reasoner. As observed, the number of statements (obtained after the reasoning process) is proportionally increased according to the number of individuals. Each population group will be used to later obtain the time

TABLE II
NUMBER OF INDIVIDUALS AND STATEMENTS PER POPULATION

Population	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Individuals	15,000	30,000	45,000	60,000	75,000	90,000	105,000	120,000	135,000	150,000
Statements	130,457	259,804	389,775	519,463	649,115	778,984	908,714	1,038,238	1,168,188	1,297,503

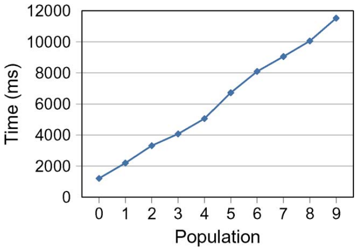


Fig. 5. Consistency checking time.

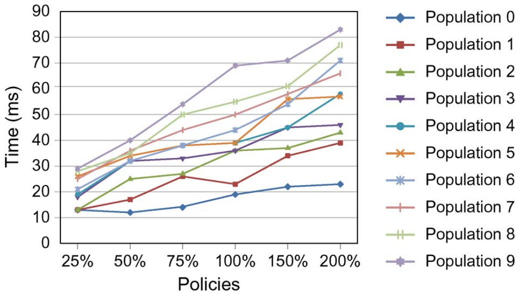


Fig. 6. Reasoning time for different populations and policies.

that SeCoMan needs to check the knowledge base consistency and infer new information.

Fig. 5 depicts the time, measured in milliseconds, used by the Reasoner to validate the ontology considering different population groups (see Table II).

Comparing the increase in individuals and statements with the reasoning time, we can observe that SeCoMan can support a very large number of individuals or statements within a reasonable reasoning time. Furthermore, the linearity property behind these results allows us to deduce that a better computer system setting would obtain a lower reasoning time.

The previous experiment has demonstrated a linear relationship between individual/statements and the reasoning time, but without considering policies. Thus, the main goal behind the next test is to check how policies can affect the reasoning time. In this sense, we defined several percentages of policies related with the persons contained in our population groups.

Fig. 6 depicts how the reasoning time varies depending on each population group (see Table II) and the percentages associated to the policies.

Policies have a very low impact in the reasoning time of our framework. For all populations, the difference between having a 25% and a 200% of policies is around a few milliseconds.

As main conclusion of this section, we have demonstrated with the previous experiments that when the number of individuals/statements is linearly increased in our ontology, the reasoning time also increases linearly. Furthermore, the semantic rules that form the policies do not have an important impact on the reasoning time.

B. Queries Performance

We want now to check how the query time varies when considering different sizes in the population and the number of policies. In this sense, we defined an experiment per each query defined in Section V: authorization, location, range, navigation, and closeness. These experiments consist on checking how the amount of individuals and the percentage of policies affect to the query response time.

Fig. 7 shows the results for each query. The x-axis corresponds to a given population group, the y-axis is the time in milliseconds, and each line symbolizes a given percentage of policies. Note that the closeness query time is not shown because this kind of query is composed of location and range queries (the response time would be the sum of both).

In order to obtain the query times for each query, we used the *unauthorizedPersons* function (defined in Section V-B) to check the authorization query time, whose results are shown in Fig. 7(a); the *elementLocation* function (defined in Section V-C) to check the location query time, whose results are shown in Fig. 7(b); the *rangeSpaceElements* function (defined in Section V-D) to check the range query time, whose results are shown in Fig. 7(c); and *getMinimumPaths* and *pathFinder* functions (defined in Section V-F) to check the navigation query time, whose results are shown in Fig. 7(d). pathFinder was implemented using the breadth-first search (BFS) algorithm. BFS is a graph search algorithm that begins with the source position and explores all the adjacent positions, examining each of the unvisited ones until finding the destination.

As shown in Fig. 7, the response time for all queries is mainly influenced by the population, as when we increased the population, the response time also increased. This is because there are more statements in the knowledge base; therefore, the complexity to answer the query is higher. Furthermore, we can observe that policies do not have a great impact in the response time as policies generate statements associated to persons, and as shown in Section VII-A, they are the 8.2% of the individuals contained in each population.

Fig. 7(b) shows that the location query time is much lower than for the rest, due to its complexity being lower. As shown in Fig. 7(d), the response time for the navigation query is much higher than for the rest of queries. We consider that its times are not an affordable time for answering a query. As we have demonstrated with the previous queries, this problem is not how to represent the information, but the complexity of the algorithm. Thus, improving the pathFinder function in order to decrease the time response is defined as future work.

As main conclusion of this section, we have demonstrated that for different kinds of queries the policies do not have a

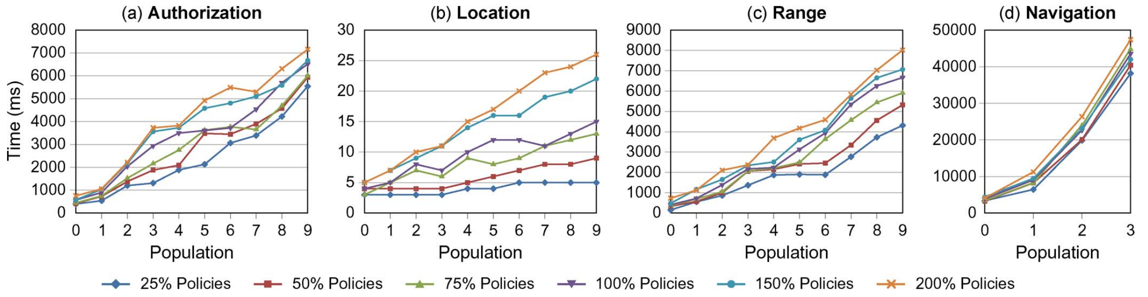


Fig. 7. Query time variation when considering different populations and policies. (a) Authorization. (b) Location. (c) Range. (d) Navigation.

significant impact in the framework performance. As it can be observed in Fig. 7, the kind of algorithm and the amount of individuals/statements are the reason for the increased time in queries and reasoning processes.

VIII. Discussion

As stated in Section II, not all context-aware systems manage policies, and those that manage policies are for different purposes, such as to protect the users' privacy. Here, we compare SeCoMan with those works presented in Section II that allow users to manage their privacy, showing in detail the policies provided by each one.

CoBrA allows users to protect their privacy through policies, indicating the personal information that they want to reveal. In SeCoMan, personal information is a topic that is beyond the scope of the Location ontology. SeCoMan preserves the privacy of the users by shaping their personal information in a context ontology whose information is managed using operational policies. On the other hand, the policies defined in CoBrA take into account the users' location and the context in which they are located, whereas SeCoMan allows actors to define richer policies than CoBrA. In our framework, users can share their location to the right users, at the right granularity, at the right place, and at the right time. Instead, the users of CoBrA cannot define policies to manage their location privacy, which is considered an important requirement in context-aware systems. SeCoMan provides four kinds of policies to allow users to manage their location privacy (see Section IV-C).

TABLE III
Comparative of Systems in Managing Policies to Preserve Users' Privacy

Policies		CoBrA	CoPS	PPCS	SeCoMan
Operational		√		√	√
Authorization					√
Location	Cloaking				√
	Hiding		√	√	√
	Granularity		√	√	√
	Closeness				√

CoPS solves the inability of CoBrA to manage the users' location privacy, although the former does not consider the personal information privacy. Using policies, CoPS allows users to decide to whom and at which precision they want to share their location and context information. The policies defined in CoPS are composed of several fields: subject, context, time, precision, application, and result. The structure of these policies has certain similarities with the SeCoMan policies. However, comparing field by field, we can select in SeCoMan a given subject, or a group of subjects, depending on their roles or privileges; context and time information are included in our policies explicitly; precision corresponds in our policies to a place, or a list of places, with different granularity; and the application and result are shaped in our policies to establish an internal classification to generate new knowledge. Therefore, SeCoMan covers users' location privacy of CoPS through the hiding and granularity policies. Furthermore, our framework allows users to generate fictitious positions for specific users and manage the level of closeness at which they want to be located by other users.

Finally, PPCS addresses the weaknesses of CoBrA and CoPS. Specifically, PPCS protects users' information and their location by allowing users to decide the granularity at which they want to share their information, to whom, and under what conditions. The time during which they want to reveal the information, or the place(s) where they want to share information, is also taken into account when users define their policies. However, and in addition to the cloaking, hiding, granularity, and closeness policies, SeCoMan grants or denies access to users to stay in certain locations depending on their privileges. In addition to that, SeCoMan also manages the context-aware information through operational policies.

Table III shows a comparison of the policies supported by the systems analyzed earlier. The rest of the proposals that were shown in Section II but were not included in Table III do not manage policies to protect users' privacy. Even solutions such as Feel@Home, Hydra, and CroCo define a module indicating that the users' privacy is protected but do not define how to develop it.

Regarding the SeCoMan performance, as we have demonstrated in Section VII, policies do not have a significant impact in reasoning and query times, allowing users to define as many policies as they want without degrading the performance. When we linearly increased the number of individuals/statements in our ontology, the reasoning time also increased linearly. These requirements should be supported by the works commented earlier. In this sense, CoBrA does not present experiments to know how these aspects affect to the system performance and scalability; CoPS demonstrates that semantic rules do not have a direct impact in the time of answering questions, as well as showing that the query time increases linearly when the system receives simultaneous queries; and PPCS demonstrates that the reasoning time is linearly increased when users increases linearly.

Furthermore, and setting CoBrA aside for not providing performance measures, the authors of CoPS and PPCS argued that query times increase linearly as the number of users also grow (similar conclusions to ours). Yet, none of them offers users further security aspects in comparison with SeCoMan, as shown in Table III and thoroughly discussed in this section.

IX. Conclusion and Future Work

In this paper, we have shown that, to the best of our knowledge, there is no framework that accomplishes all the essential requirements to develop context-aware smart applications using a semantic-oriented IoT vision. To this end, we presented a context-aware framework called *SeCoMan* that allows developing smart applications where users can share their location to the right users, at the right granularity, at the right place, and at the right time. Queries based on location, context awareness, and authorization are predefined in the framework to provide smart applications with the space and context information. Ontologies are the key for modeling the context, inferring new knowledge through semantic reasoners, and sharing this knowledge with independent systems. Moreover, the framework functions are defined in a manner that accommodate IoT requirements, and they neither affect system performance nor introduce excessive overheads.

As next steps of this research, we plan to integrate SeCoMan in the world of cloud computing [24]. Our idea is to offer the Context Management layer of SeCoMan as middleware, located at the Platform as a Service (PaaS) layer of the cloud architecture. This layer will provide the information needed by different context-aware applications located in the Software as a Service (SaaS) layer. Furthermore, we will benefit from other advantages of cloud computing, such as elasticity, monitoring, auditing, load balancing, and security issues. We also plan to improve users' privacy by adding anonymity and hashing policies to hide and disguise the identity of a user [25].

Finally, the support and implementation of outdoor based-location services is another research topic for future work, where global positioning systems, such as GPS or Galileo, can be used to get the position of people and objects in order to offer services based on outside locations.

References

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

[2] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" in *Handbook on Ontologies*. Berlin, Germany: Springer-Verlag, 2009, ser International Handbooks on Information Systems, pp. 1–17.

[3] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of things: A survey," *IEEE Commun. Surveys Tuts.* [Online]. Available: http://dx.doi.org/10.1109/SURV.2013.042313.00197

[4] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan, "Smart semantic middleware for the internet of things," in *Proc. 5th Int. Conf. Inf. Control, Autom. Robot.*, May 2008, pp. 169–178.

[5] B. Guo, L. Sun, and D. Zhang, "The architecture design of a cross-domain context management system," in *Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Apr. 2010, pp. 499–504.

[6] A. Badii, M. Crouch, and C. Lallah, "A context-awareness framework for intelligent networked embedded systems," in *Proc. 3rd Int. Conf. Adv. Hum.-Oriented Pers. Mech., Technol. Services*, Aug. 2010, pp. 105–110.

[7] S. Pietschmann, A. Mitschick, R. Winkler, and K. Meissner, "CroCo: Ontology-based, cross-application context management," in *Proc. 3rd Int. Workshop Semantic Media Adapt. Pers.*, Dec. 2008, pp. 88–93.

[8] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An ontology-based context model in intelligent environments," in *Proc. Commun. Netw. Distrib. Syst. Model. Simul. Conf.*, Jan. 2004, pp. 270–275.

[9] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *Knowl. Eng. Rev.*, vol. 18, no. 3, pp. 197–207, Sep. 2003.

[10] D. Ejigu, M. Scuturici, and L. Brunie, "CoCA: A collaborative context-aware service platform for pervasive computing," in *Proc. 4th Int. Conf. Inf. Technol.*, Apr. 2007, pp. 297–302.

[11] P. Jagtap, A. Joshi, T. Finin, and L. Zavala, "Preserving privacy in context-aware systems," in *Proc. 5th IEEE Int. Conf. Semantic Comput.*, Sep. 2011, pp. 149–153.

[12] V. Sacramento, M. Endler, and F. N. Nascimento, "A privacy service for context-aware mobile computing," in *Proc. 1st Int. Conf. Security Privacy Emerging Areas Commun. Netw.*, Sept. 2005, pp. 182–193.

[13] University of Murcia, Murcia, Spain, Complete definition of the SeCoMan ontologies. [Online]. Available: http://reclamo.inf.um.es/secoman

[14] J. M. Marín Pérez, J. Bernal Bernabé, J. M. Alcaraz Calero, F. J. Garcia Clemente, G. Martínez Pérez, and A. F. Gómez Skarmeta, "Semantic-based authorization architecture for grid," *Future Gen. Comput. Syst.*, vol. 27, no. 1, pp. 40–55, Jan. 2011.

[15] C. Becker and F. Dürr, "On location models for ubiquitous computing," *Pers. Ubiquit. Comput.*, vol. 9, no. 1, pp. 20–31, Jan. 2005.

[16] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, SWRL: A semantic web rule language combining OWL and RuleML, May 2004, W3C Member Submission.

[17] B. Motik, P. F. Patel-Schneider, and B. Parsia, OWL 2 web ontology language: Structural specification and functional-style syntax, Dec. 2012, W3C Recommendation.

[18] Stanford Center for Biomedical Informatics Research, Stanford, CA, USA, Protégé: A free, open source ontology editor and knowledge-base framework. [Online]. Available: http://protege.stanford.edu

[19] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics, Sci., Services Agents World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.

[20] The Apache Software Foundation, Forest Hill, MA, USA, The Apache Jena2 ontology API. [Online]. Available: http://jena.apache.org/documentation/ontology

[21] E. Prud'hommeaux and A. Seaborne, SPARQL query language for RDF, Jan. 2008, W3C Recommendation.

[22] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Proc. 3rd Int. Conf. Pervasive Comput.*, May 2005, pp. 116–133.

[23] Trascends, Glastonbury, CT, USA, Rifidi - Connect the Internet of Things. [Online]. Available: http://sourceforge.net/projects/rifidi

[24] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[25] E. Paintsil, "Evaluation of privacy and security risks analysis construct for identity management systems," *IEEE Syst. J.*, vol. 7, no. 2, pp. 189–198, Jun. 2013.

Alberto Huertas Celdrán received the M.Sc. degree in computer science from the University of Murcia, Murcia, Spain.

He is currently a Research Associate with the Department of Information and Communication Engineering, University of Murcia. His scientific interests include security, semantic technology, and policy-based context-aware systems.

Félix J. García Clemente received the M.Sc. and Ph.D. degrees in computer science from the University of Murcia, Murcia, Spain.

He is currently an Associate Professor of computer networks with the Department of Computer Engineering, University of Murcia. His research interests include security and management of distributed communication networks.

Manuel Gil Pérez received the M.Sc. degree in computer science from the University of Murcia, Murcia, Spain.

He is currently a Research Associate with the Department of Information and Communication Engineering, University of Murcia. His scientific activity is mainly devoted to security infrastructures, trust management, and intrusion detection systems.

Gregorio Martínez Pérez (M'80) received M.Sc. and Ph.D. degrees in computer science from the University of Murcia, Murcia, Spain.

He is currently an Associate Professor with the Department of Information and Communication Engineering, University of Murcia. His research interests include security and management of distributed communication networks.

Chapter 3

PRECISE: Privacy-aware recommender based on context information for Cloud service environments

Title:	PRECISE: Privacy-aware recommender based on context information for Cloud service environments
Authors:	Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez
Journal:	IEEE Communications Magazine
Impact factor:	4.007 Q1
Publisher:	IEEE
Volume:	52
Number:	8
Pages:	90-96
Year:	2014
Month:	August
DOI:	10.1109/MCOM.2014.6871675
State:	Published

PRECISE: Privacy-Aware Recommender Based on Context Information for Cloud Service Environments

Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, and Gregorio Martínez Pérez

The authors are with the University of Murcia.

This work has been partially funded with support from the Spanish MICINN (project RECLAMO, Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management, with code TIN2011-28287-C02-02, and project Mejora de Arquitectura de Servidores, Servicios y Aplicaciones, with code TIN2012-38341-C04-03) and the European Commission (FEDER/ERDF). Thanks also to the Funding Program for Research Groups of Excellence granted by the Séneca Foundation with code 04552/GERM/06.

ABSTRACT

Context-aware systems based on location open up new possibilities to users in terms of acquiring custom services by gathering context information, especially in systems where the high mobility of users increases their usability. In this context, this article presents a privacy-preserving solution offering context-aware services based on location in MCC. We propose a middleware, called PRECISE, which provides users with custom context-aware recommendations. These recommendations are given by considering the context information, and the users' locations, privacy policies, and previously visited places. MCC plays a key role in this solution, moving the data processing and storage needs to the cloud, as well as further advantages such as elasticity and load balancing. A thorough discussion when comparing PRECISE with other related works confirms that our solution improves the most relevant proposals so far.

INTRODUCTION

Context awareness is a concept that combines the environment, users' locations, identities of nearby people and objects, and changes in the previous terms [1]. The ability to determine users' and objects' locations at a particular time supplies valuable information to offer services from the elements of the environment where they are. These services can range from a simple location-based service (LBS) to emergency services, car navigation systems, or tour planning for tourists.

Nowadays, we can find some proposals that aim at offering context-aware services based on location, such as LOC8 [2], SOCAM [3], and CoCA [4]. LOC8 is a powerful framework that supports querying the environment in which users are located, whereas SOCAM infers information about a given context considering information from others. Instead, CoCA proposes a collaborative context-aware service platform where the users' location is inferred by considering information about the context and the location of the elements.

However, the ability to locate people raises serious privacy concerns, such as the right of users to determine when, how, and under what conditions their location can be released to other users or services. In this sense, other context-aware systems are focused on preserving location privacy by using policies. Among them, CoBrA [5] protects the privacy of its users by using rules, inferring whether they have the right permissions to share and/or receive information. Another example is CoPS [6], where users can control who can access their context data, when, and at what level of granularity. Finally, a more recent work, PPCS [7], proposes that the users' location and context information are shared (or not) depending on their privacy policies.

With previous solutions, users cannot protect their privacy against services, only against other users. Furthermore, they are oriented to offer local services in specific environments, setting aside distributed schemes. Providing context-aware services based on location is a complex task due to:

- The need to acquire the users' location and context-aware information from different sources
- The large amount of data required to shape the context-aware information
- The processing power requirements to analyze the information and provide the services
- The mobility of users when they make use of the services
- The management of the users' privacy according to their preferences
- The need to offer services that run on federated systems spanning different organizations

In attempting to meet these requirements, we figured out that the mobile cloud computing (MCC) paradigm can play a key role [8]. MCC provides mobile users with data processing and storage services in a cloud environment. Thus, mobile devices belonging to roaming users do not need large capacity for storage and processing. Furthermore, context-aware services based on location can share certain information to provide users with better and more personalized services, by using federation mechanisms as the one presented in [9]. To the best of our knowledge, there is not a wide spread of context-aware services based on location facing users' privacy concerns in MCC, this being a priority recently identified in [8].

0163-6804/14/$25.00 © 2014 IEEE

In this article, we present an evolution of our work called *SeCoMan* [10]. Concretely, we propose a privacy-preserving and context-aware system that provides location-based services (LBS) in the MCC paradigm, which allows users to define privacy policies regarding services (although it is able to do it between users too, as an extension of the SeCoMan approach). This implies that the architectures and the benefits obtained from both solutions are rather different. With PRECISE, users can:

- *Release* their locations to specific services
- *Hide* their positions to specific users
- *Mask* their locations to other users by generating fictitious (fake) positions
- Establish the *granularity* and *closeness* at which they want to be located by services or users
- Preserve their *anonymity* to specific services

The central component of our solution is a middleware, called Privacy-Aware Recommender Based on Context Information for Cloud Service Environments (PRECISE), placed between users and context-aware services. PRECISE is allocated at the platform as a service (PaaS) layer of the MCC paradigm. This manages the context-aware users' information, their privacy policies, and their behavior patterns (i.e., the users' movements while staying in the environment). Context-aware services are in turn allocated at the software as a service (SaaS) layer of MCC, providing users with recommendations about context-aware information. Thus, PRECISE is a context-aware system that receives the users' location, context information, and changes in both from independent middleware, which can belong to the cloud environment or not.

How to Preserve Users' Privacy in LBS

Users should be able to dynamically control their information by using rules that form policies, for example. Among the different sets of policies, we emphasize here the use of privacy-oriented policies, with which users can define their privacy preferences related to their location, identity, and personal information.

The set of rules forming the policies in PRECISE are composed of the following elements: *Type* is the kind of policy; the *target* is the person who defines the policy and whose information is being managed; the *requester* is the service or person who requests information about the target; the *place* is the region of the environment where the policy will be enforced; and the *result* determines the relationship the requester will have with the target's information. PRECISE is in charge of managing the context-aware information implicitly contained in each rule. Note that the context can be composed of some elements, such as *Profile*, *Date*, or *Time*.

Type∧*Target*∧*Requester*∧*Place*∧[*Profile*]∧[*Date*]∧[*Time*]→Result

In the above context, PRECISE allows users to define two kinds of policies, which are introduced below.

Location Management Policies

These policies are defined by the users to specify their privacy preferences regarding their location. They are divided into five groups: *Release*, *Hiding*, *Cloaking*, *Granularity*, and *Closeness*. In any of the foregoing policies, the target can define different policies with the same requester regarding certain *Places*, *Profiles*, *Dates*, or *Times*, whether this requester is a user or a service, depending on the kind of policy.

As PRECISE hides the users' location to services by default, release policies are oriented to reveal the users' locations to specific services. This will allow a requester (service) to know the target's position (user). On the other hand, hiding and cloaking policies are oriented to users, as, unlike in the previous case, the users' location is available to other users by default. They can hide their location to a given set of users by defining hiding policies. Cloaking (or masking) policies are intended to generate one or more fake positions for a particular user, so other users cannot distinguish the real position in which the target is. Granularity policies can also be defined indicating the maximum accuracy at which users want to be located, may it be applied to users or services. With this kind of policy, it is possible to define several levels of granularity depending on the context in which the service is being provided. An example of these levels could be the section or building where the user is located.

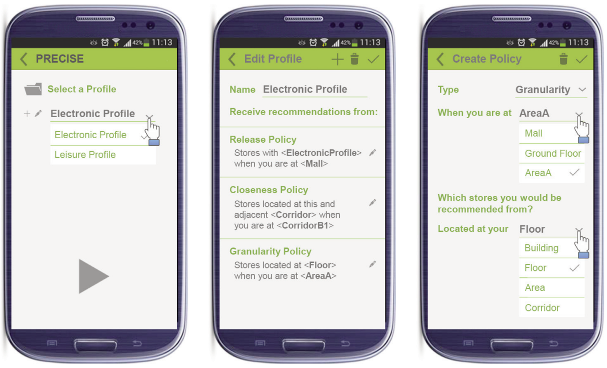
Finally, closeness policies can be applied to both users and services with the aim of indicating the minimum level of nearness at which the target (user) wants to be located. The nearness level is established with the same values as the ones defined for the granularity policies.

Anonymity Management Policies

PRECISE guarantees anonymous use of context-aware services, as users do not want a priori to reveal either their identity or personal information on what they consider sensitive data. To disclose such information, we define a new type of policy, called *Revealing*, oriented to services. This kind of policy can be specified by the users to receive custom recommendations from specific context-aware services. Otherwise, users will only receive general context-aware recommendations. In the literature, we can find several proposals in overcoming the anonymity of a user. For example, in [11], a solution for the anonymous use of services by using pseudonyms is proposed.

Depending on the users' needs and preferences, they can use some *profiles* defined by the system, or created by them, to reveal the right information (e.g., their location only) to the right requester (service), at the right place and time, in order to receive custom context-aware recommendations. A profile in our solution is related to a specific topic and is formed by a given set of policies. As an example, Fig. 1 shows how a user can select a given profile to start receiving recommendations, edit an already defined profile, or create a determined policy by using a mobile application. Note that the content of these three subfigures are subsequently explained in the next section.

Users should be able to dynamically control their information, by using rules that form policies, for example. Among the different sets of policies, we emphasize here the use of the privacy-oriented policies, with which users can define their privacy preferences related to their location, identity, and personal information.



Name Electronic Profile

Receive recommendations from:

Release Policy
Stores with <ElectronicProfile> when you are at <Mall>

Closeness Policy
Stores located at this and adjacent <Corridor> when you are at <CorridorB1>

Granularity Policy
Stores located at <Floor> when you are at <AreaA>

Type Granularity

When you are at AreaA
Mall
Ground Floor
AreaA ✓

Which stores you would be recommended from?

Located at your Floor
Building
Floor ✓
Area
Corridor

Figure 1. How a user can choose a given profile and how the policies can be created or edited through a mobile application: a) selecting a given profile to start receiving custom context-aware recommendations; b) editing a profile to receive recommendations when meeting a number of policies; c) defining a granularity policy about where receiving recommendations.

A Motivating Example

We present in this section an example to show how our solution manages the users' privacy in context-aware services based on their location. We define a scenario based on a mall where users receive recommendations about products and offers on their mobile devices. In this scenario, the main entity of PRECISE acts as the mall's middleware itself, giving recommendations according to the users' location, their profiles, and the context-aware information related to the space description of the environment, the description of the elements close to the user, and their relevance and meaning.

In our scenario, we have a mall with different stores and places to buy products and enjoy leisure time. This mall has the infrastructure to know the location of its users and a service to provide custom context-aware recommendations. Users' location is registered by the mall's middleware, that is, by PRECISE, to record their patterns of behavior when interacting with all services and stores provided in the mall; for example, time visiting a given store or routines of the users after staying in a given space. Using this information, PRECISE (the mall's middleware) is able to give useful recommendations to its users about products and offers provided by the stores located at the mall. These recommendations take into consideration the context-aware information, the users' location, some profiles defining typical customs of the users, and patterns about their behavior. Changes in the context are managed by PRECISE receiving the context-aware information from the infrastructure deployed at the mall. The map of the mall is graphically depicted in Fig. 2.

As an example to show how the previous mall's middleware (PRECISE) gives recommendations to its users, suppose a user (*Andy*) moving around the shopping center who is interested in electronic products. To receive recommendations, he uses his mobile device, access to the mall's middleware, and selects *ElectronicProfile* (Fig. 1a).

Electronic Profile

This profile contains policies that form the Andy's privacy preferences. The first rule consists of a *release* policy, indicating that his position has to be released to the services with *ElectronicProfile* when he is at the mall (Fig. 1b). Thus, he will only receive recommendations of electronic stores.

Person(*#Andy*)∧*isLocated*(*#Andy*, *#Mall*)∧ *Service*(*?Requester*)∧*hasProfile*(*?Requester*, *#ElectronicProfile*) → *hasRelease*(*#Andy*, *?Requester*)

The next rule is a *closeness* policy. It indicates that Andy only wants to receive recommendations of stores located in the same and adjacent corridors when he is located in *CorridorB1* (recommendation point 1 in Fig. 2). As a result, Andy will only receive recommendations of ElectronicStoreB.

Person(*#Andy*)∧*isLocated*(*#Andy*, *#CorridorB1*) → *hasGranularity*(*#Andy*, *#Floor*)

The last rule is a *granularity* policy (Fig. 1c), indicating that Andy only wants to receive recommendations of the stores placed on the same floor when he is at *AreaA* (recommendation point 2, Fig. 2). Thus, Andy will receive recommendations of ElectronicStoreA and ElectronicStoreB.

Person(*#Andy*)∧*isLocated*(*#Andy*, *#AreaA*) → *hasGranulariaty*(*#Andy*, *#Floor*)

Suppose that Andy is on the *UndergroundFloor* (recommendation point 3, Fig. 2), still using the ElectronicProfile set. The mall's middleware will recommend some products of ElectronicStoreA, ElectronicStoreB, and ElectronicStoreC, taking into account that Andy visited ElectronicStoreA after receiving the second recommendation. Moreover, Andy will receive no recommendation of CinemaA, as CinemaA's service is not contained in ElectronicProfile. After (maybe) visiting some electronic stores, Andy decides to change his profile for another (Fig. 1a): *LeisureProfile*.

Leisure Profile

As before, the first rule consists of a *release* policy. This indicates that Andy wants to release his position to the services with *LeisureProfile* when he is at the mall. He will then receive general recommendations about movies in theaters.

Person(*#Andy*)∧*isLocated*(*#Andy*, *#Mall*)∧*Service*(*?Requester*)∧*hasProfile*(*?Requester*, *#LeisureProfile*)→*hasRelease*(*#Andy*, *?Requester*)

The next rule is a *revealing* policy, and indicates that Andy wants to reveal his pseudonym to *CinemaA* just when he is on the *FirstFloor* (recommendation point 4, Fig. 2). Andy will start receiving custom recommendations about movies when he is on the first floor, considering

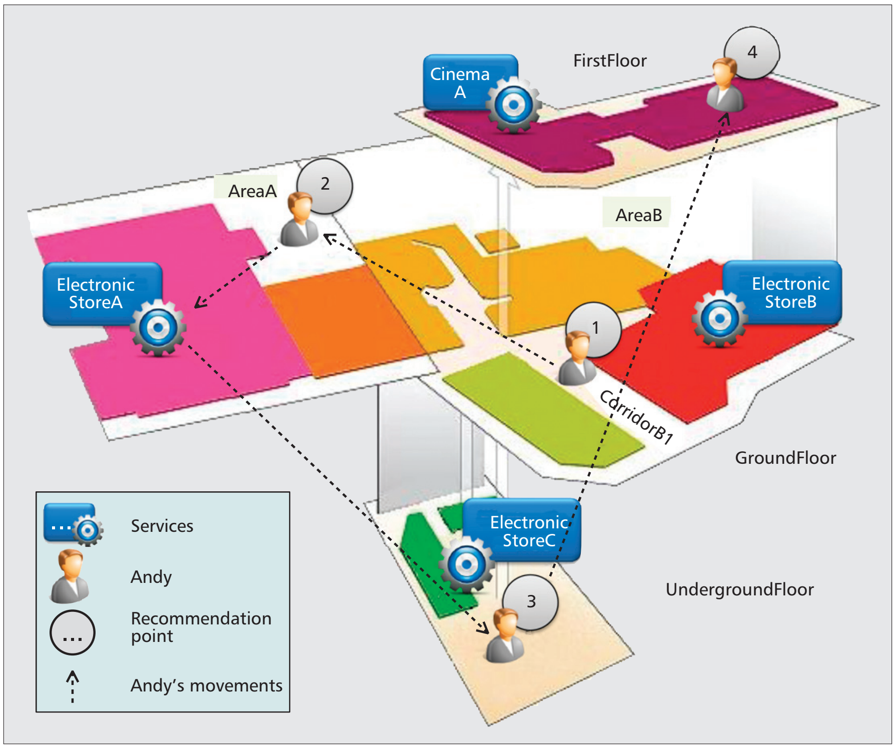


Figure 2. Overview of the scenario.

The storage and computing requirements are allocated at the IaaS layer of the MCC paradigm, in order to manage the large amount of data needed to shape the contextual information and the processing power to provide the services.

his records and preferences. This information is known by *CinemaA*'s service because Andy provided it during a previous registration process.

Person(?*Andy*)∧*isLocated*(#*Andy*, #*FirstFloor*)∧
→*hasRevealing*(#*Andy*, #*CinemaA*)

Finally, suppose that half an hour before the movie starts, Andy receives a notification of cancellation and recommendations about other interesting movies. This implies that when there are context changes, the mall's middleware will receive these changes and report such information to users who have the previous policies set, like Andy.

ARCHITECTURE

This section describes our architecture to deploy a privacy-preserving and context-aware system that provides services based on location. Figure 3 depicts the proposed architecture.

This architecture is oriented to the cloud in order to allow actors to use and manage the resources more efficiently. Context-aware services, *ElectronicStores* and *CinemaA* in the previous section, are located at the SaaS layer to provide recommendations according to their internal context. Instead, PRECISE (the mall's middleware) is a privacy-preserving middleware allocated at the PaaS layer to manage the global system context as well as the users' information.

The context awareness and space information are provided by independent middleware (the mall's infrastructure), which can belong to the cloud or not. The storage and computing requirements are allocated at the IaaS layer of the MCC paradigm in order to manage the large amount of data needed to shape the contextual information and the processing power to provide the services.

ACTORS

The PRECISE administrator (the mall's administrator in the previous section) manages the information related to users' locations and registration of the context-aware services. This actor also indicates to the *Communication manager* the different middleware required to receive location and context-aware information. On the other hand, the Service administrators (the administrators of *ElectronicStores* and *CinemaA*) are in charge of managing the context-aware information of their services, so each service has its own administrator.

The last type of actor in our solution is the user (Andy). Users are people who use PRECISE to obtain recommendations about the context in which they are located. They define their policies to manage their privacy directly in PRECISE, without having to rely on context-aware services.

COMPONENTS

The layers composing our system architecture provide the necessary resources so that the actors can use and manage the system. The *ElectronicStores* and *CinemaA* of the previous section are privacy-preserving services because they do not know Andy's identity, location, or personal

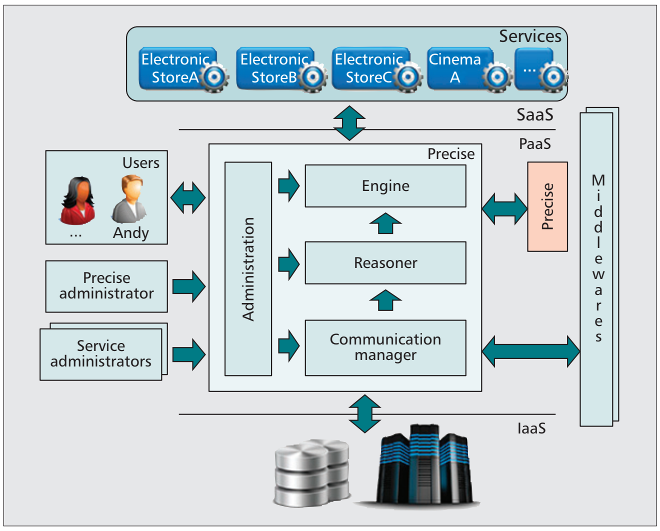


Figure 3. The PRECISE architecture.

information. As Andy is registered with *CinemaA* and defined a revealing policy for such a service, PRECISE will only reveal to *CinemaA* a pseudonym for him. Using that pseudonym, *CinemaA* will access Andy's information (provided by Andy during the registration process) and will be able to provide him with more personalized context-aware recommendations. Instead, the *ElectronicStores* will only provide Andy general context-aware recommendations.

As shown, PRECISE is a middleware that manages the context and preserves the users' privacy, without having to undergo a registration process. This implies that Andy, for example, will have different pseudonyms in different sessions, so PRECISE will not be able to track him. Thus, Andy's behavior pattern will just contain information about the current session. In another case, when Andy is registered in PRECISE, it will be able to access Andy's pseudonym and link his behavior patterns of different sessions, thereby providing him better suited context-aware recommendations.

In order to manage the context, PRECISE uses ontologies to shape the contextual and user information, semantic rules to define the policies, and semantic reasoning to infer new knowledge. The complete definition of the PRECISE ontologies can be accessed and downloaded from [12].

To perform all tasks, PRECISE has different components. The *Engine* component is in charge of requesting recommendations to the context-aware services, and provides users with recommendations based on previous ones. The *Reasoner* component makes the decision whether or not to request recommendations to specific services. This component infers the decision making as input for the updated ontological model formed by the union of the context-aware information and the users' information, and the semantic rules defined by the users.

The *Communication manager* component is in charge of receiving the context-aware and space information from the middlewares. This provides independence for PRECISE with regard to the sources of information.

Finally, the *Administration* component is responsible for supporting administrative tasks of the actors, including policy management, registration of the context-aware services, and management of the *Communication manager* component.

Sequence Diagram

The interaction between the PRECISE components is formed by four main blocks of steps, which are represented on the right of Fig. 4. This figure shows the sequence diagram, remarked on below, when Andy is at *AreaA* (recommendation point 2, Fig. 2). These blocks, from the perspective of PRECISE, are:

- Obtain the contextual and space information from independent middleware solutions and store it.
- Manage the context-aware information gathered earlier, and use Andy's location and his policies to decide with which services to ask for recommendations in a privacy preserving fashion.
- Ask for recommendations to the corresponding services and receive their products and offers.
- Provide Andy with recommendations by considering the services' recommendations, Andy's preferences, and his patterns of behavior.

Discussion

We present here a thorough comparison between our solution and the main related work introduced in the first section, which is shown in Table 1. First, LOC8 provides users with space and context information, although it does not allow users to define rules to dynamically manage their information or protect their privacy. In this sense, PRECISE lets users manage their information and privacy preferences by using rules to protect their identities, personal information, and locations against specific context-aware services or users.

SOCAM uses rules for inferring specific context information by considering other contexts and facts. CoCA is another solution that uses rules to dynamically manage context information. It infers users' locations taking into account the context and space information. Instead, PRECISE makes use of reasoners to infer new knowledge considering the context and space information. In spite of SOCAM's and CoCA's use rules, they do not allow users to define policies to manage their privacy preferences.

CoBrA allows users to protect their privacy by using policies, indicating the personal information that they want to reveal to other users. Instead, PRECISE is a privacy-preserving solution that, by default, protects users' identities, not revealing personal information to context-aware services. PRECISE allows users to define policies to specific services and users, not just to users as CoBrA does. On the other hand, the policies defined in CoBrA take into account the users' location and the context in which they are located, whereas PRECISE allows users to define richer policies than CoBrA. Users of CoBrA cannot define policies to manage their

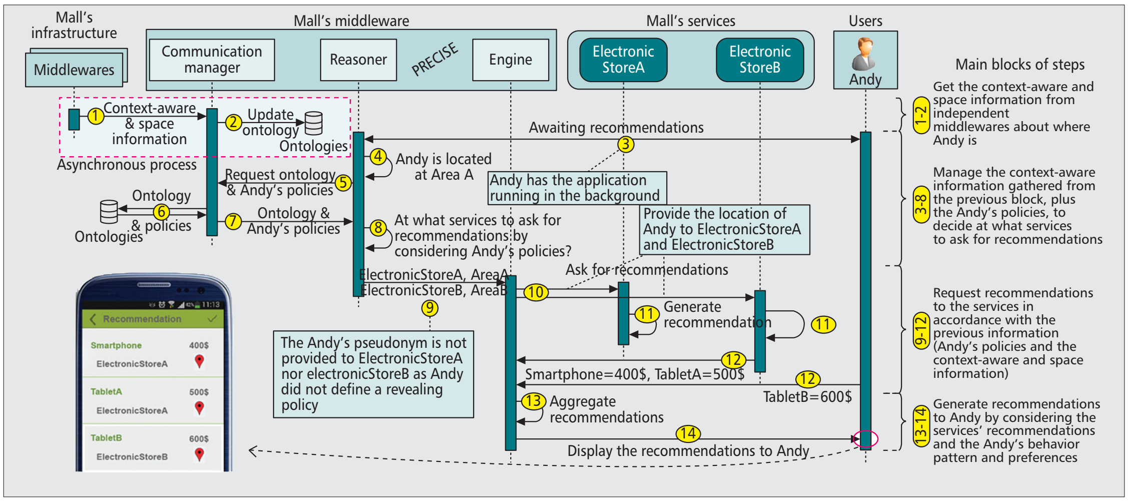

Mall's infrastructure
Middlewares
Mall's middleware
Communication manager
Reasoner
PRECISE
Engine
Mall's services
Electronic StoreA
Electronic StoreB
Users
Andy
Main blocks of steps
1 Context-aware & space information
Asynchronous process
2 Update ontology
Ontologies
Awaiting recommendations
3
4 Andy is located at Area A
Andy has the application running in the background
Request ontology & Andy's policies 5
Ontology 6 & policies
Ontologies
7 Ontology & Andy's policies
8 At what services to ask for recommendations by considering Andy's policies?
Provide the location of Andy to ElectronicStoreA and ElectronicStoreB
ElectronicStoreA, AreaA; ElectronicStoreB, AreaB
9
Ask for recommendations
10
11 Generate recommendation
11
The Andy's pseudonym is not provided to ElectronicStoreA nor electronicStoreB as Andy did not define a revealing policy
12 Smartphone=400$, TabletA=500$
12 TabletB=600$
13 Aggregate recommendations
14 Display the recommendations to Andy
Recommendation
Smartphone 400$ ElectronicStoreA
TabletA 500$ ElectronicStoreA
TabletB 600$ ElectronicStoreB
1-2 Get the context-aware and space information from independent middlewares about where Andy is
3-8 Manage the context-aware information gathered from the previous block, plus the Andy's policies, to decide at what services to ask for recommendations
9-12 Request recommendations to the services in accordance with the previous information (Andy's policies and the context-aware and space information)
13-14 Generate recommendations to Andy by considering the services' recommendations and the Andy's behavior pattern and preferences

Figure 4. Sequence diagram showing the interactions between the PRECISE components when Andy is located at AreaA.

location privacy, which is considered an important requirement in context-aware systems.

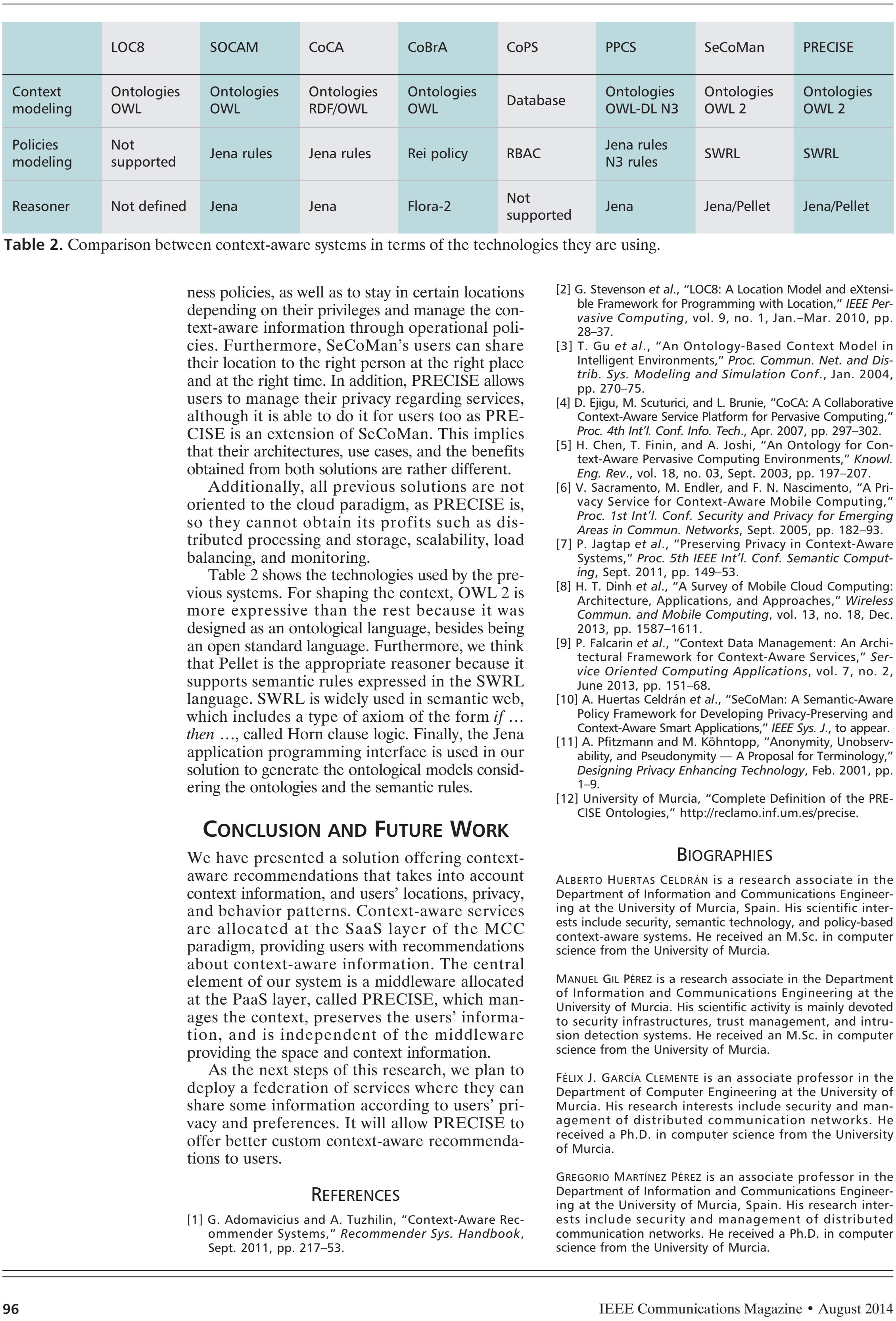
CoPS addresses the inability of CoBrA to manage users' location privacy, although CoPS does not consider privacy preferences about personal information. Using policies, CoPS allows users to decide to whom and at which precision they want to share their location and context information with other users. Instead, PRECISE's users can define polices to specific users, services, or a group of them depending on their location, profiles, context, and time. Therefore, PRECISE covers the users' location privacy of CoPS through hiding and granularity policies. Furthermore, PRECISE allows users to generate fictitious positions for specific users and manage the level of closeness at which they want to be located.

PPCS takes into account the shortcomings of the two previous systems. Specifically, PPCS protects users' information and locations by allowing users to decide the granularity at which they want to share their information, to whom, and where; and the time during which they want to reveal information. However, PPCS is not able to generate fictitious positions or establish the level of closeness at which users want to reveal their information.

Finally, SeCoMan addresses the drawbacks of PPCS, allowing users to define cloaking and close-

	LOC8	SOCAM	CoCA	CoBrA	CoPS	PPCS	SeCoMan	PRECISE
Context awareness	✓	✓	✓	✓	✓	✓	✓	✓
Cloud paradigm								✓
Rules		✓	✓	✓	✓	✓	✓	✓
User privacy				✓	✓	✓	✓	✓
Service privacy								✓
Anonymity policies				✓		✓		✓
Release policies								✓
Revealing policies								✓
Hiding policies					✓	✓	✓	✓
Cloaking policies							✓	✓
Granularity policies					✓	✓	✓	✓
Closeness policies							✓	✓

Table 1. Comparison of context-aware systems.



	LOC8	SOCAM	CoCA	CoBrA	CoPS	PPCS	SeCoMan	PRECISE
Context modeling	Ontologies OWL	Ontologies OWL	Ontologies RDF/OWL	Ontologies OWL	Database	Ontologies OWL-DL N3	Ontologies OWL 2	Ontologies OWL 2
Policies modeling	Not supported	Jena rules	Jena rules	Rei policy	RBAC	Jena rules N3 rules	SWRL	SWRL
Reasoner	Not defined	Jena	Jena	Flora-2	Not supported	Jena	Jena/Pellet	Jena/Pellet

Table 2. Comparison between context-aware systems in terms of the technologies they are using.

ness policies, as well as to stay in certain locations depending on their privileges and manage the context-aware information through operational policies. Furthermore, SeCoMan's users can share their location to the right person at the right place and at the right time. In addition, PRECISE allows users to manage their privacy regarding services, although it is able to do it for users too as PRECISE is an extension of SeCoMan. This implies that their architectures, use cases, and the benefits obtained from both solutions are rather different.

Additionally, all previous solutions are not oriented to the cloud paradigm, as PRECISE is, so they cannot obtain its profits such as distributed processing and storage, scalability, load balancing, and monitoring.

Table 2 shows the technologies used by the previous systems. For shaping the context, OWL 2 is more expressive than the rest because it was designed as an ontological language, besides being an open standard language. Furthermore, we think that Pellet is the appropriate reasoner because it supports semantic rules expressed in the SWRL language. SWRL is widely used in semantic web, which includes a type of axiom of the form *if* … *then* …, called Horn clause logic. Finally, the Jena application programming interface is used in our solution to generate the ontological models considering the ontologies and the semantic rules.

Conclusion and Future Work

We have presented a solution offering context-aware recommendations that takes into account context information, and users' locations, privacy, and behavior patterns. Context-aware services are allocated at the SaaS layer of the MCC paradigm, providing users with recommendations about context-aware information. The central element of our system is a middleware allocated at the PaaS layer, called PRECISE, which manages the context, preserves the users' information, and is independent of the middleware providing the space and context information.

As the next steps of this research, we plan to deploy a federation of services where they can share some information according to users' privacy and preferences. It will allow PRECISE to offer better custom context-aware recommendations to users.

References

[1] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," *Recommender Sys. Handbook*, Sept. 2011, pp. 217–53.

[2] G. Stevenson *et al.*, "LOC8: A Location Model and eXtensible Framework for Programming with Location," *IEEE Pervasive Computing*, vol. 9, no. 1, Jan.–Mar. 2010, pp. 28–37.

[3] T. Gu *et al.*, "An Ontology-Based Context Model in Intelligent Environments," *Proc. Commun. Net. and Distrib. Sys. Modeling and Simulation Conf.*, Jan. 2004, pp. 270–75.

[4] D. Ejigu, M. Scuturici, and L. Brunie, "CoCA: A Collaborative Context-Aware Service Platform for Pervasive Computing," *Proc. 4th Int'l. Conf. Info. Tech.*, Apr. 2007, pp. 297–302.

[5] H. Chen, T. Finin, and A. Joshi, "An Ontology for Context-Aware Pervasive Computing Environments," *Knowl. Eng. Rev.*, vol. 18, no. 03, Sept. 2003, pp. 197–207.

[6] V. Sacramento, M. Endler, and F. N. Nascimento, "A Privacy Service for Context-Aware Mobile Computing," *Proc. 1st Int'l. Conf. Security and Privacy for Emerging Areas in Commun. Networks*, Sept. 2005, pp. 182–93.

[7] P. Jagtap *et al.*, "Preserving Privacy in Context-Aware Systems," *Proc. 5th IEEE Int'l. Conf. Semantic Computing*, Sept. 2011, pp. 149–53.

[8] H. T. Dinh *et al.*, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches," *Wireless Commun. and Mobile Computing*, vol. 13, no. 18, Dec. 2013, pp. 1587–1611.

[9] P. Falcarin *et al.*, "Context Data Management: An Architectural Framework for Context-Aware Services," *Service Oriented Computing Applications*, vol. 7, no. 2, June 2013, pp. 151–68.

[10] A. Huertas Celdrán *et al.*, "SeCoMan: A Semantic-Aware Policy Framework for Developing Privacy-Preserving and Context-Aware Smart Applications," *IEEE Sys. J.*, to appear.

[11] A. Pfitzmann and M. Köhntopp, "Anonymity, Unobservability, and Pseudonymity — A Proposal for Terminology," *Designing Privacy Enhancing Technology*, Feb. 2001, pp. 1–9.

[12] University of Murcia, "Complete Definition of the PRECISE Ontologies," http://reclamo.inf.um.es/precise.

Biographies

Alberto Huertas Celdrán is a research associate in the Department of Information and Communications Engineering at the University of Murcia, Spain. His scientific interests include security, semantic technology, and policy-based context-aware systems. He received an M.Sc. in computer science from the University of Murcia.

Manuel Gil Pérez is a research associate in the Department of Information and Communications Engineering at the University of Murcia. His scientific activity is mainly devoted to security infrastructures, trust management, and intrusion detection systems. He received an M.Sc. in computer science from the University of Murcia.

Félix J. García Clemente is an associate professor in the Department of Computer Engineering at the University of Murcia. His research interests include security and management of distributed communication networks. He received a Ph.D. in computer science from the University of Murcia.

Gregorio Martínez Pérez is an associate professor in the Department of Information and Communications Engineering at the University of Murcia, Spain. His research interests include security and management of distributed communication networks. He received a Ph.D. in computer science from the University of Murcia.

Chapter 4

What private information are you disclosing? A privacy-preserving system supervised by yourself

Title:	What private information are you disclosing? A privacy-preserving system supervised by yourself
Authors:	Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez
Conference:	Proceedings of the 6th International Symposium on Cyberspace Safety and Security
Publisher:	IEEE
Year:	2014
Month:	August
DOI:	10.1109/HPCC.2014.199
State:	Published

2014 IEEE International Conference on High Performance Computing and Communications (HPCC), 2014 IEEE 6th International Symposium on Cyberspace Safety and Security (CSS) and 2014 IEEE 11th International Conference on Embedded Software and Systems (ICESS)

What private information are you disclosing? A privacy-preserving system supervised by yourself

Alberto Huertas Celdrán*, Manuel Gil Pérez*, Félix J. García Clemente†, and Gregorio Martínez Pérez*

* Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071 Murcia, Spain
Email: alberto.huertas@um.es, mgilperez@um.es, gregorio@um.es

† Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, 30071 Murcia, Spain
Email: fgarcia@um.es

***Abstract*—Preserving the privacy of users' information is an essential requirement in information management systems. The emergence of context-aware services makes protecting the users' information an even greater challenge. Addressing this challenge requires an automatic mechanism that allows users to control their information at real-time. In this context, we propose a middleware called Context-Aware PRIvacy-preserving system Supervised by users (CAPRIS), which provides users with groups of policies that form profiles to protect their privacy in the environment in which they are located. CAPRIS lets users to control and supervise at real-time the information they are revealing to other users who use context-aware services. Semantic technologies play a key role in our solution. We use ontologies for shaping the space and context, and semantic rules defining the privacy policies that form the context-aware profiles. Some experiments measuring the throughput and scalability of CAPRIS confirm that our solution improves other related works.**

I. Introduction

How to preserve the privacy of users' information is an open challenge that has been extensively studied in multiple application domains [1]. The emergence of mobile devices, as the main access point to share information, has increased the number of applications offering value-added services. In this sense, the location in a given environment about objects, devices, and people provides meaningful information to offer context-aware services [2]. They can range from a simple location in a map to a list of products or services provided by the elements that are part of the environment.

A very important issue in context-aware systems based on location is the management of the users' privacy. A great majority of context-aware solutions relies on static privacy policies defined at set-up time, which are not easy enough to manage by users. Moreover, these policies are not suitable for context-aware environments, as users should be able to supervise (grant or deny) the access to their information at real-time depending on their context and situation. Therefore, in our opinion, privacy-preserving and context-aware solutions should allow users to control what information they want to release, who can access them, and in which contexts and situations they want to disclose such information.

Due to the lack of systems supplying the previous tasks, we present here a privacy-preserving system on context-aware environments that lets users grant or deny the access to their information at real-time. Our main contribution in this paper is a middleware called CAPRIS (*Context-Aware PRIvacy-preserving system Supervised by users*) that considers the context in which users are located to provide them with groups of policies that protect their privacy in such context. These policies allow users to decide at real-time what, where, when, how, to whom, and at which level of precision they want to release their information. Our solution does not require that users manage their privacy, but they just have to choose the most appropriate group of policies suggested by CAPRIS in accordance with their preferences. Furthermore, users can modify or create new policies depending on their own interests. Using CAPRIS, users may preserve:

- The *space* in which they are located with different levels of granularity (e.g., building, floor, or area)
- Their *personal information* with different levels of precision (e.g., name, age, or address)
- The *activity* they are undertaking at any given time (e.g., working, meeting, or taking a rest)
- The information oriented to the *context* in which they are located (e.g., the user's medical record within a hospital context)

The space and context-aware information in our solution is managed by using semantic web techniques. This provides a common infrastructure that makes possible to represent, process, and share information between independent systems more easily. In this sense, we use a collection of ontologies to shape the space and context-aware information, semantic rules to define the users' policies, and semantic reasoning to infer new knowledge. Thus, *ontology* is a key factor in our solution in order to develop a context-aware system. All this information is provided by location systems and middlewares, which are independent to our system. This independence allows CAPRIS to choose between these systems depending on the characteristics of the environment.

The remainder of this paper is organized as follows. In Section II, we review the related work regarding other privacy-preserving solutions. Privacy policies are presented in Section III, whereas Section IV shows how our system models the space and context-aware information. A motivating example is presented in Section V, considering the policies and the ontologies explained in the previous two sections. Section VI presents the proposed architecture, including a sequence diagram that shows the interaction between CAPRIS and users. Section VII reports some experimental results to illustrate the performance of our solution. Finally, conclusions and future works are drawn in Section VIII.

978-1-4799-6123-8/14 $31.00 © 2014 IEEE
DOI 10.1109/HPCC.2014.199

IEEE computer society

II. Related work

An important requirement of any information management system is to protect the information against unauthorized access. During the last decades, many works have been done on the topic of controlling users' privacy.

A. Role-based models to protect users' privacy

Several works let users to define privacy policies to control the access to their information, such as the one presented in [3] where a Role-Based Access Control (RBAC) model was presented to manage roles that control the access to the users' information. Afterward, an RBAC-based model was presented in [4] with further features to establish role hierarchies, where roles might include permissions of others. Hierarchies are a natural means for structuring roles, whereby seniors' roles acquired the permissions of their juniors.

B. Privacy through protecting location information

The increase of mobile devices such as smartphones has meant the emergence of services based on location (LBS), and consequently a new challenge regarding users' privacy. Researches in user-oriented privacy policies mainly focused on location. For example, [5] shows how the RBAC model can be extended to incorporate the concept of location, proposing the Location-aware Role-Based Access Control (LRBAC) model that considers the users' location to determine if a user has access (or not) to a given object.

Another related work was presented in [6], where users could apply general policies to control the distribution of their information. Users might define policies by considering several elements such as *statement*, *limit time*, *limit location*, *validator*, and *quality of service*. On the other hand, a method was presented in [7] to combine policies and cryptographic primitives with the aim of protecting location information. This proposed two kinds of policies, *User* and *Room*, with which to restrict the granularity of the returned location and limit the time interval during which the access is granted.

C. Privacy in context-aware computing environments

Knowing the users' location, the next stage to provide useful services consists on considering the context in which users are located. Services can provide relevant information to any user by exploiting the information about people, devices, and objects around. In this sense, a recent survey on context-aware systems analyzes a large number of solutions that manage policies to preserve users' privacy [8].

Besides such works, SeCoMan [9] allows developing context-aware applications to preserve the user's privacy in an Internet of Things (IoT) paradigm. This solution proposes a taxonomy of policies where *operational*, *authorization*, and *location* policies can be defined to manage the contextual and users' information, as well as protecting the users' location. Instead, PRECISE [10] is an extension of SeCoMan oriented to the cloud that provides context-aware recommendations preserving the user's location and identity. In contrast, CAPRIS is oriented to all context-aware services and to protect not only the users' location, but also the users' activities and their personal and context-aware information.

There also exist other works that allows preserving the users' information. For example, CoBrA [11] is a context-aware architecture where distributed agents share information with each other. The spaces composed of smart agents, devices, and sensors are defined by using ontologies. CoBrA allows users to protect their privacy by using policies, indicating the personal information that they reveal to others. The policies defined in CoBrA take into account the users' location and the context in which they are located, whereas CAPRIS allows users to define richer policies than CoBrA considering other aspects, such as the requester, time, and activity. Furthermore, in CoBrA, users cannot define policies to manage their location privacy, which is a key factor in context-aware systems.

CoPS [12] is another privacy-preserving and context-aware proposal. It allows users to decide to whom, when, and at which precision they want to share their location and context-aware information. It uses optimistic and pessimistic approaches to classify the policies into different hierarchical levels. CoPS solves the drawback of CoBrA in managing the users' location privacy, although the former does not consider privacy policies about personal information. Instead, the CAPRIS's users can define polices to specific users, or to a group of them, depending on their location, roles, context, and time. Thus, our solution covers the users' location and the contextual privacy of CoPS, besides also preserving the users' personal information and their activities.

Finally, PPCS [13] is a policy-based framework that allows users to protect their information with different levels of privacy in environments with mobile devices. PPCS takes into account the shortcomings of the two previous systems. Its access control considers the dynamic information inferred from the context and the static information about the owner, in order to share or not his/her location, activity, and personal and contextual information. Specifically, PPCS protects users' information and their location by allowing them to decide the granularity at which they want to share their information, to whom, the place, or the time during which they want to reveal such information. This solution is the most related to ours. Yet, both PPCS and the previous works require that users manage their privacy by defining their own policies. In contrast, CAPRIS offers context-aware and privacy profiles that preserve the users' information. Therefore, to the best of our knowledge, there is no solution that preserves the users' information without requiring the user management.

III. Preserving the users' privacy

Privacy is an essential aspect in information management systems. Our solution preserves the users' information by using policies, which form profiles. A *profile* in CAPRIS is formed by a given set of policies that protect the privacy of users' information in a given context. Furthermore, a policy can in turn belong to a given profile or be shared between different profiles belonging to a same context. So, when a user changes of context, e.g., he/she is moving from one location to another, CAPRIS provides him/her with several profiles about the new context. Such a user can then choose one of these profiles, or even modify them according to their interests. Therefore, profiles are predefined in CAPRIS by the service administrators who perfectly know the workflow within the context where profiles will be used.

Fig. 1 shows two contexts with several profiles, which have some shared policies, and a user located in *Context_A* who has chosen *Profile_A*.

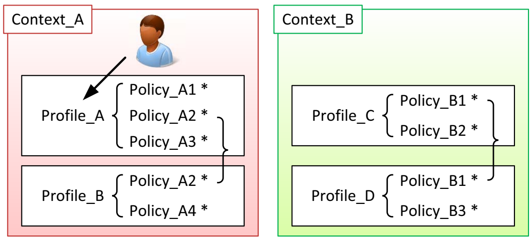


Fig. 1. Context-aware profiles with several policies for each

The policies that form the profiles managed by CAPRIS are composed of the following elements: *Type* is the kind of policy; *maker* is who defines the policy (policies can be suggested by CAPRIS or defined by users); *target* is the user or group of them whose information is managed by the policy; *requester* is the user or group of them who request the target's information; and *result* is the relationship that determines the access to the information. In our proposal, we also include new elements to make policies richer and powerful:

- *What* is the information revealed by the target
- *Where* represents the space of the context in which the policy is applied
- *When* defines the date and time during which the policy is applied
- *How* is the activity done by the target or requester

In the context of CAPRIS, the maker can define different policies with the same target regarding a certain requester about what, where, when, or how elements depending on the type of policy. CAPRIS defines two phases to share the users' information: the *disclosure phase* in charge of indicating what users' information is shared and the *reveal phase* deciding where, when, and how the users' information is shared. Both are covered below by *disclosure* and *reveal* policies.

A. Disclosure policies

By default, CAPRIS hides the users' information to other users. Disclosure policies (see below) are used to indicate *what* information the target wants to share with a given requester. In the *What* element, users establish if they want to share their location, activity, personal information, or data related to the context where they are. Note that these policies do not indicate where, when, or how the information is revealed.

$Type \wedge Maker \wedge \text{Target} \wedge \text{Requester} \wedge \textbf{What} \rightarrow Result$

B. Reveal policies

Reveal policies are defined to indicate where, when, how, and to whom (requester) the target's information, established in the disclosure policies, can be shared. Note that the *When* and *How* elements are optional in this policy indicating that the policy will be applied at any time and anyway.

$Type \wedge Maker \wedge \text{Target} \wedge \text{Requester} \wedge \textbf{Where} \wedge [\textbf{When}] \wedge [\textbf{How}] \rightarrow Result$

This kind of policy (reveal) is related to the previous one (disclosure) through the target and requester elements. In this sense, this policy is in charge of activating the disclosure policies when users are located in certain places or contexts.

It is worth mentioning that we do not consider in CAPRIS policies conflict if users do not decide to disclose or reveal their personal data. It is not possible to have overlapping of policies: one granting the access, and another denying it.

IV. Modeling the Space and Context-Aware Information

Our solution manages a collection of ontologies to shape the space and context-aware information. This collection is composed of an ontology called *CAPRIS* that models the users' location and the common information for all contexts, as well as a set of ontologies for modeling the context-aware services. Fig. 2 depicts the CAPRIS ontology and the ontologies of the two context-aware services that will be subsequently used in the scenario presented in Section V.

A. The CAPRIS ontology

The CAPRIS ontology is categorized into two topics: user and location. *User* is the top-level class of the user topic, which refers to persons who use the system to obtain context-aware information in a privacy-preserving way. Users can have several *Role* and *Activity*, which can be used to receive more personalized information. The *PersonalInformation* element models the users' information, common for all contexts. As a proof of concept, this class has four predefined subclasses: *Name*, *Age*, *Telephone*, and *Address*.

In order to model the users' location, a hierarchical model has been defined in the CAPRIS ontology. *Space* is the top-level class in this model, having five predefined subclasses, namely (from low to high accuracy): *Building*, *Floor*, *Area*, *Section*, and *Position*.

As shown in Fig. 2, the entities of the CAPRIS ontology are related each other by properties, where some of them have been defined to establish new relationships through policies. For example, the disclosure policies use the *hasInformationDisclosure* property to link *PersonalInformation* with *User* and the *hasLocationDisclosure* property to allow users to decide when disclosing their location. Instead, the reveal policies are generated by using the *hasRevealing* property between two users to share parts of their information.

B. The Hospital and Pharmacy ontologies

To model a given hospital context, we defined the Hospital ontology based on the work proposed in [14]. In this ontology, users can be *HospitalStaff* or *Patient*. The hospital staff can be *Doctor*, *Nurse*, or *LabStaff*, whereas patients can have several *MedicalRecord* and *Treatment*. Both elements have a group of predefined subclasses covering some medical aspects, such as *Neurology*, *Dermatology*, or *Psychiatry*.

With regard to the Pharmacy ontology, users can be either *PharmacyStaff* or *Customer*.

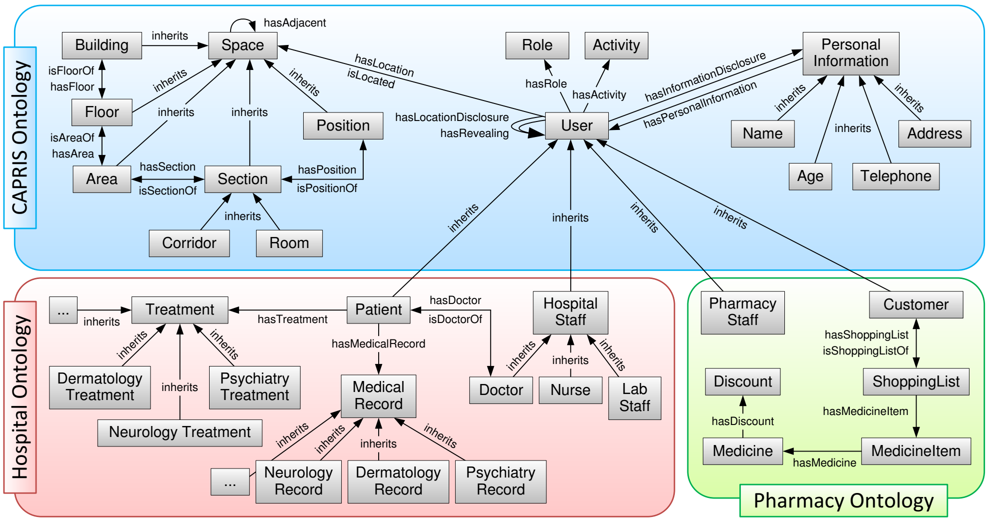


Fig. 2. The CAPRIS ontology, common for all contexts, and the hospital and pharmacy ontologies for shaping the information of both contexts

The top-level of the Pharmacy ontology is *Medicine*, which represents an article of the pharmacy. As observed, in order to add new ontologies to our system, the new ontology only has to inherit from *User* of the CAPRIS ontology. In the pharmacy context, *Medicine* can have *Discount*, and some may belong to a *ShoppingList* containing one or more *MedicineItem*.

V. A Motivating Example

This section presents an example to show how CAPRIS protects the users' information in context-aware environments. Suppose an inter-domain scenario composed of a hospital and a pharmacy. Both contexts have deployed the appropriate physical infrastructure to know the location of their users, as well as specific services to provide context-aware information. Instead, the CAPRIS middleware can be logically deployed anywhere on the Internet (e.g., in the cloud).

Fig. 3 graphically shows the scenario commented earlier.

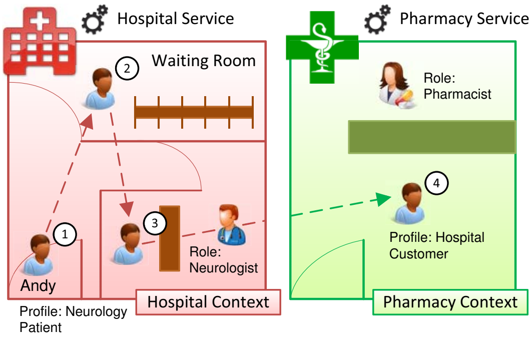


Fig. 3. Scenario composed of a hospital and a pharmacy

CAPRIS uses the user's location, provided by the service's underlying infrastructure, to infer whether he/she is located at the hospital or the pharmacy, and it offers context-aware profiles to preserve his/her information in such a context.

In our example scenario, a user named *Andy* visits the hospital to have a check-up with his neurologist. Since he is concerned about what information can be revealed to the hospital staff and other patients, both elements belonging to the Hospital ontology defined in Section IV, he uses our system to protect his information. When Andy accesses CAPRIS through his mobile device notes that he is already registered in the CAPRIS middleware. CAPRIS detects that Andy is located at the hospital after considering the information provided by the underlying hospital's location infrastructure. Then, CAPRIS offers him some context-aware profiles such as *Dermatology patient*, *Neurology patient*, or *Psychiatry patient*.

It is worth mentioning that these profiles are suggested by the hospital service to CAPRIS, and CAPRIS is in charge of offering it to users. Andy chooses the *Neurology patient* profile (point 1 in Fig. 3), which contains the policies that form his privacy preferences.

A. Neurology patient profile

The first policy of the *Neurology patient* profile consists on a *disclosure* policy related to the users' information. This policy (defined below) indicates that the patient, Andy in our example, disclose his neurology record to doctors owning the *Neurologist* role. Note that the context-aware elements such as *Patient*, *Doctor*, *NeurologyRecord*, and their properties are shaped in the Hospital ontology defined in Section IV.

It is important to remember that in our system the users' information is hidden by default. Therefore, CAPRIS allows users to have as many disclosure policies as they want to share each piece of their information.

$$Patient(?Target) \wedge NeurologyRecord(?Record) \wedge Doctor(?Requester) \wedge hasMedicalRecord(?Target, ?Record) \wedge hasDoctor(?Requester, ?Target) \wedge hasRole(?Requester, \#Neurologist) \rightarrow hasInformationDisclosure(?Record, ?Requester)$$

The second policy is another disclosure policy related to the users' location. This establishes that the patient (Andy) release to the hospital staff their location with a granularity of *Room*. With this policy, the hospital staff is able to know that Andy is in the waiting room at a certain time (point 2, Fig. 3). The location information is shaped in a hierarchical way by the CAPRIS ontology (Fig. 2).

$$Patient(?Target) \wedge HospitalStaff(?Requester) \wedge Room(?Location) \wedge hasLocation(?Target, ?Location) \rightarrow hasLocationDisclosure(?Target, ?Requester)$$

The last policy is a *reveal* policy. This declares in our example that, just when a patient like Andy is located in the hospital building, he/she will reveal the information established by his/her previous disclosure policies to the hospital staff whose activity is *Working*. Therefore, just when Andy is located at the hospital, his neurologist can access his neurology record (established by the first disclosure policy), and the hospital staff can access his location with a granularity of room (established by the second disclosure policy).

$$Patient(?Target) \wedge HospitalStaff(?Requester) \wedge isLocated(?Target, \#Hospital) \wedge hasActivity(?Requester, \#Working) \rightarrow hasRevealing(?Target, ?Requester)$$

Once Andy has chosen the *Neurology patient* profile, his neurologist uses the hospital service (he is already registered in the service) to know when Andy is located in the waiting room (point 2, Fig. 3).

When Andy is with his neurologist (point 3, Fig. 3), Andy receives in his mobile device a request from CAPRIS, asking him if grants to his neurologist to access his neurological record. Andy accepts it, and his doctor is able to access Andy's neurology record using the hospital service. It is important to notice that, to reveal information, Andy has to be located at the hospital, select the *Neurology patient* profile, and accept the requests to access his information. Finally, the neurologist uses the hospital service to update the Andy's neurology treatment with a new prescription. Note that the use case presented here is just an example of our proposal. But, and according to the policies defined earlier, the requester (neurologist) and the target (Andy) may be in distant places.

After a while, Andy leaves the hospital and goes to the pharmacy to buy the medicines prescribed by his doctor. When he is there, he selects a new context-aware profile. CAPRIS detects the new context and offers him some context-aware profiles such as *Normal customer* and *Hospital customer*. Andy chooses the *Hospital customer* profile (point 4, Fig. 3).

B. *Hospital customer profile*

The first policy of the *Hospital customer* profile consists on a *disclosure* policy related to the users' information. This policy establishes that the patient (Andy) disclose his new treatment to the pharmacy staff having a *Pharmacist* role. In this policy, it is important to notice that *Treatment* belongs to the Hospital ontology (Fig. 2). In this sense, CAPRIS is able to define inter-domain policies, allowing users to share contextual information between different contexts.

$$Patient(?Target) \wedge Treatment(?Medication) \wedge hasTreatment(?Target, ?Medication) \wedge PharmacyStaff(?Requester) \wedge hasRole(?Requester, \#Pharmacist) \rightarrow hasInformationDisclosure(?Treatment, ?Requester)$$

The last policy is a *reveal* policy. This indicates that, just when the customer (Andy) is located in the pharmacy building, he reveals the information established by previous disclosure policies to the pharmacy staff whose activity is *Working*. The context-aware elements such as *Customer*, *PharmacyStaff*, and their properties are shaped in the Pharmacy ontology defined in Section IV, graphically shown in Fig. 2.

$$Customer(?Target) \wedge PharmacyStaff(?Requester) \wedge isLocated(?Target, \#Pharmacy) \wedge hasActivity(?Requester, \#Working) \rightarrow hasRevealing(?Target, ?Requester)$$

Once Andy has selected the *Hospital customer* profile, the pharmacist uses the pharmacy's service to access the Andy's neurology treatment to give him the medicines prescribed by his neurologist (point 3, Fig. 3). Previously, Andy must be accepted the request for accessing his neurology record.

VI. Architecture

This section describes our architecture to let users control the privacy on their information in context-aware services based on location, such as the Hospital and Pharmacy services explained earlier in Section V.

The CAPRIS middleware is a trusted third party that provides users with profiles to protect their privacy in the context in which they are located, as well as managing the common context for services and the users' information. The space and contextual information is provided by independent location systems and middlewares (concretely, the Hospital's and Pharmacy's infrastructure in the example presented in Section V). Fig. 4 depicts the proposed architecture.

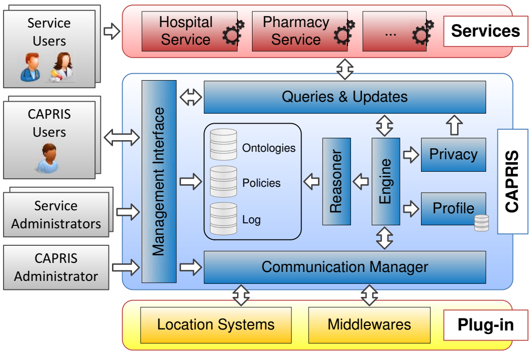
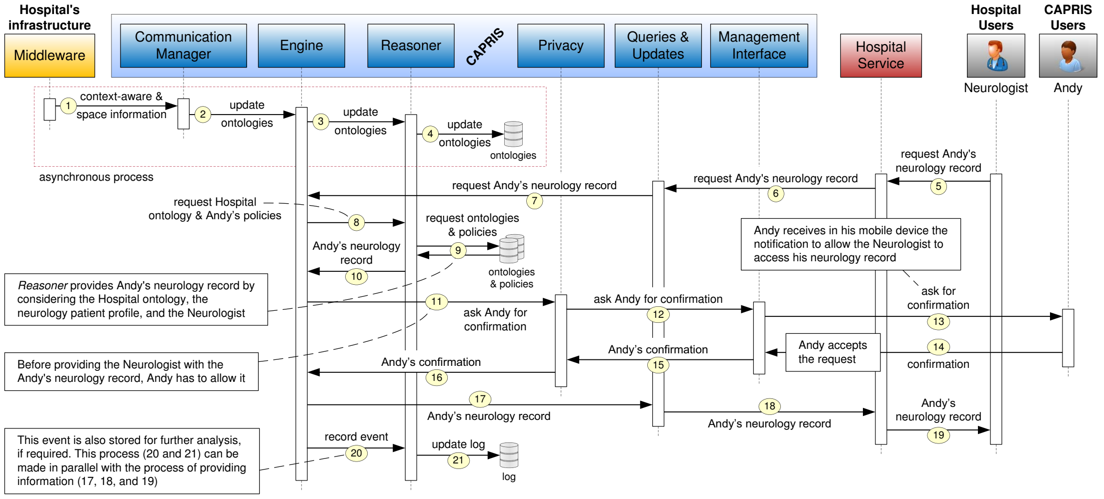


Fig. 4. Overview of the CAPRIS architecture



Reasoner provides Andy's neurology record by considering the Hospital ontology, the neurology patient profile, and the Neurologist

Before providing the Neurologist with the Andy's neurology record, Andy has to allow it

This event is also stored for further analysis, if required. This process (20 and 21) can be made in parallel with the process of providing information (17, 18, and 19)

Fig. 5. Diagram of sequence of CAPRIS

Users have full authority to share the information that they want under their total responsibility, so that although CAPRIS is a centralized trusted party, no one can access to other users' information in a malicious way without their consent.

A. Actors

In our solution, we have defined four kinds of actors. The *CAPRIS Administrator* is in charge of managing the ontology of CAPRIS, registering the context-aware services, defining and managing the context-aware profiles, and indicating to *Communication Manager* the location systems and middlewares to receive the space and context-aware information. On the other hand, *Service Administrators* (Hospital and Pharmacy services' administrators) are in charge of managing contextual information, e.g., the Hospital and Pharmacy ontologies, and the context-aware and privacy-preserving profiles, so that each service has its own administrator(s).

CAPRIS Users is another type of actor who wants to use CAPRIS to protect their information through context-aware and privacy profiles, like Andy in Section V. Finally, *Service Users* (the neurologist and the pharmacist in such an example) use the context-aware services to obtain information related to their contexts and users; for example, the neurologist to ask for the Andy's neurology record, as defined in Section V.

B. Layers of the CAPRIS architecture

The three layers composing our system architecture provide the required resources to allow actors to use and manage the system. The Hospital and the Pharmacy services are located at the *Services* layer, which provide users with information aware to both domains. As an example, the Hospital service can provide its users with information about the space (waiting room), the patients' location (where Andy is), the patients' personal information (Andy's age), and the patients' contextual information (neurology record of Andy).

To manage the space and context-aware information, CAPRIS defines the ontologies shown in Section IV, the semantic rules of Section III defining the privacy policies to protect the user's information, and the semantic reasoners to infer new knowledge considering the previous information.

CAPRIS includes various components to perform these tasks (Fig. 4). *Queries & Updates* maintains two processes to: (i) provide services with predefined queries about the space, users' information, and contextual information; and (ii) allow a service to modify the users' information related to its context (e.g., the patient's medical record). In both processes, *Privacy* is in charge of ensuring that the information owner allows the access to his/her information in real-time. When *Engine* receives a request, query or update, *Reasoner* will generate an ontological model with the ontologies and the users' privacy policies. Considering the inferred information, *Engine* knows if the user's information can be revealed to the requester. If so, *Privacy* requests confirmation to the owner; otherwise, the owner is not requested and the information is not revealed. Note that events will be recorded in the *Log* database.

Engine also detects when users change the context. Using the user's location, CAPRIS is able to know the context where the user is, and thus *Profile* will suggest him/her some context-aware profiles accordingly. On the other hand, *Management Interface* is responsible for supporting administrative tasks of the three actors of CAPRIS, including policy management, registration of the context-aware services, and the management of the *Communication Manager* component. Finally, *Communication Manager* is in charge of collecting the space and context-aware information from the *Plug-in* layer about the elements that form part of the environment, their location, and further information depending on the environment. This is composed of different plug-ins that interact with *Middlewares*, which in turn communicate with sensors or other devices to receive context information, and with *Location Systems* to obtain information about the space.

C. Sequence Diagram

The interactions between the CAPRIS components are formed by six main blocks of steps. Fig. 5 shows the sequence diagram, commented below, when Andy is at the hospital and his neurologist wants to access the Andy's neurology record (point 3, Fig. 3). These six main blocks are:

- Getting the contextual and space information from the independent middlewares and storing it in the ontologies managed by CAPRIS (steps 1-4)
- Receiving the request from the Andy's neurologist by using the Hospital service to access Andy's neurology record (steps 5-9)
- Managing the context-aware information allocated in the ontologies and using the *Neurology patient* profile to decide if Andy should be asked for allowing his neurologist to access his neurology record (step 10)
- Asking Andy for confirmation to share his information and managing the response (steps 11-16)
- Providing the Andy's neurologist with the Andy's neurology record (steps 17-19)
- Recording the event in the *Log* database for further analysis, if any (steps 20 and 21)

VII. Experimental results on CAPRIS

We deployed a prototype of CAPRIS to provide some profiles that protect the privacy of the users' information in the hospital and pharmacy environments. Thus, we also implemented the Hospital and Pharmacy services. By using them, we show in this section some experimental results to measure the throughput and scalability of CAPRIS.

A. CAPRIS prototype

In CAPRIS, policies forming privacy profiles are expressed in SWRL (Semantic Web Rule Language) [15]. It includes a type of axiom, called Horn clause logic, of the form $if...\ then...$, this being the most used in semantic web.

CAPRIS, *Hospital*, and *Pharmacy* ontologies are defined in OWL 2 (Web Ontology Language) [16], which were generated with the Protégé tool[1]. We chose OWL 2 rather than other languages like RDF, RDFS, or DAML+OIL because OWL 2 is more expressive than the rest. It was specifically designed as an ontology language, this being the main ontology language used in semantic web nowadays. It is worth noting that both OWL 2 and SWRL are two standard languages recommended by the W3C consortium.

The *Reasoner* component, described in Section VI, uses the Jena API[2]. With this application programming interface, *Reasoner* can generate ontological models with the information shaped in the ontologies, as well as policies for the Pellet reasoner to infer new knowledge [17].

Finally, the *Engine* component is in charge of translating the queries performed by the users into SPARQL queries [18], which are applied to the inferred model of CAPRIS to get the corresponding result.

B. CAPRIS performance

Experiments are presented below to measure the throughput and scalability of CAPRIS, dealing with questions such as:

(a) Is the computing *time of reasoning* acceptable?

(b) How it *scales* with different amounts of individuals and policies?

(c) How the *query time* varies in the previous premises?

As experimental setting, the conducted tests were carried out in a dedicated PC with an Intel Core i7-3770 3.40 GHz, 16 GB of RAM, and an Ubuntu 12.04 LTS as operative system. The results have been obtained by executing the experiments 100 times, and after computing their arithmetic mean.

With regard to the (a) reasoning time and (b) its scalability, we measured the CAPRIS performance by making several executions with different complexity. The complexity is related to the statements hold in the knowledge base, which depends on the number of individuals present in the ontologies and the number of semantic rules forming the policies. Individuals contained in the ontologies are referred as *population*. For the experiments, the population was randomly generated in a driver way to create a realistic scenario. For instance, *User* is the 10% of the population; *Space* the 75%; context-aware elements, such as *MedicalRecord* or *Medicine*, the 10%; and others like *PersonalInformation*, *Role*, or *Activity*, the 5%.

To evaluate the *reasoner scalability*, we defined an initial population of 30,000 individuals, which were increased with 30,000 in each step. Table I shows the relationships between the individuals and the statements generated by *Reasoner*, which indicates the complexity of our ontologies. As observed, the statements after the reasoning process was proportionally increased according to the number of individuals.

Population	**0**	**1**	**2**	**3**	**4**
Individuals	30,000	60,000	90,000	120,000	150,000
Statements	243,562	487,153	734,253	971,654	1,203,428

TABLE I. Individuals and statements per population

In order to check the *reasoning time*, we used the previous populations and different numbers of policies per user. Fig. 6 depicts how the reasoning time (y-axis) varies depending on each population of Table I and the number of policies (x-axis).

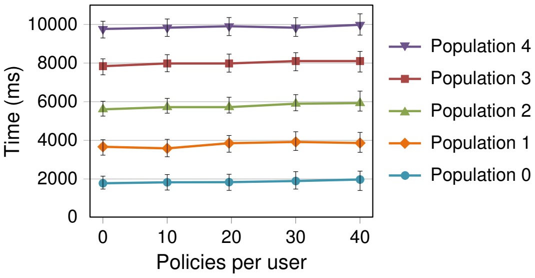


Fig. 6. Reasoning time for different populations and policies per user

[1] http://protege.stanford.edu

[2] http://jena.apache.org/documentation/ontology

The previous experiment answers the questions (a) and (b). It demonstrates that semantic rules forming policies do not have an important impact on the reasoning time, and when the number of statements is linearly increased, the reasoning time also increases linearly.

To answer question (c), we measured the *query time* when a user asks for information about another user. This was shown in Section VI-C when the Andy's neurologist wanted to access the Andy's neurology record. Fig. 7 shows that the response time (y-axis) for queries is mainly influenced by the population (x-axis). This is because there are more statements in the ontologies. Furthermore, policies do not have a great impact on the query time, as policies are related with users and they are the 10% of the individuals contained in each population.

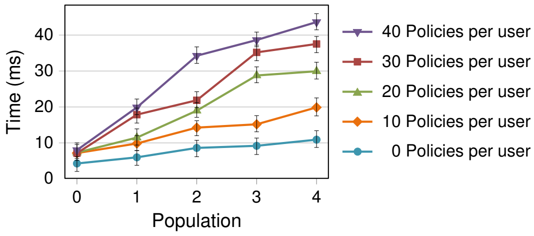


Fig. 7. Query time variation for different populations and profiles

In this section, we have demonstrated that policies forming profiles do not have a significant impact in the performance of CAPRIS. Furthermore, it can be efficiently used to protect the privacy of user's information with no scalability problem.

VIII. Conclusion and Future Work

We have presented in this paper a solution with which protects the users' information in context-aware environments, no requiring further privacy management by users. Privacy-preserving and context-aware profiles are provided by our system considering the location in which users are located. Our main element is a middleware called CAPRIS *(Context-Aware PRIvacy-preserving system Supervised by users)*. CAPRIS manages the context, preserves the users' information by providing context-aware profiles, and is independent to the location systems and middlewares that provide the space and context-aware information. Using CAPRIS, users supervise at real-time what, where, when, how, to whom, and at which level of precision they want to reveal their information. They can preserve the space where they are with different levels of granularity, their personal information with different levels of precision, the activity they are undertaking at any given time, and the information oriented to the context where they are.

As next steps of this research, we plan to provide more intelligence to CAPRIS. In order to offer fine-grained context-aware profiles, we plan to consider other aspects besides the users' location, such as the users' behavior patterns for example. Another aspect to be taken into account is the context of people located close to the user (e.g., if a user is close to friends using a University context, his/her context is probably the University too). Finally, we also plan to study a migration of our architecture to a decentralized environment, by making use of federation mechanisms, for example.

Acknowledgment

This work has been partially funded with support from the Spanish MICINN (project RECLAMO, *Virtual and Collaborative Honeynets based on Trust Management and Autonomous Systems applied to Intrusion Management*, with code TIN2011-28287-C02-02, and project *Mejora de Arquitectura de Servidores, Servicios y Aplicaciones*, with code TIN2012-38341-C04-03) and the European Commission (FEDER/ERDF). Thanks also to the Funding Program for Research Groups of Excellence granted by the Séneca Foundation with code 04552/GERM/06.

References

[1] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surveys*, vol. 42, no. 4, pp. 14:1–14:53, Jun. 2010.

[2] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proc. 1st Workshop Mobile Comput. Syst. Applicat.*, Dec. 1994, pp. 85–90.

[3] 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, Feb. 1996.

[4] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: Towards a unified standard," in *Proc. 5th ACM Workshop Role-based Access Control*, Jul. 2000, pp. 47–63.

[5] I. Ray, M. Kumar, and L. Yu, "LRBAC: A location-aware role-based access control model," in *Proc. 2nd Int'l Conf. Info. Syst. Security*, Dec. 2006, pp. 147–161.

[6] G. Myles, A. Friday, and N. Davies, "Preserving privacy in environments with location-based applications," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 56–64, Jan. 2003.

[7] U. Hengartner and P. Steenkiste, "Protecting access to people location information," in *Proc. 1st Int'l Conf. Security in Pervasive Comput.*, Mar. 2004, pp. 25–38.

[8] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 2014.

[9] A. Huertas Celdrán, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, "SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications," *IEEE Syst. J.*, to appear.

[10] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez, "PRECISE: Privacy-aware recommender based on context information for cloud service environments," *IEEE Commun. Mag.*, vol. 52, no. 8, Aug. 2014.

[11] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *Knowl. Eng. Rev.*, vol. 18, no. 03, pp. 197–207, Sep. 2003.

[12] V. Sacramento, M. Endler, and F. N. Nascimento, "A privacy service for context-aware mobile computing," in *Proc. 1st Int'l Conf. Security and Privacy for Emerging Areas in Commun. Net.*, Sep. 2005, pp. 182–193.

[13] P. Jagtap, A. Joshi, T. Finin, and L. Zavala, "Preserving privacy in context-aware systems," in *Proc. 5th IEEE Int'l Conf. Semantic Computing*, Sep. 2011, pp. 149–153.

[14] H. J. Ko and W. Kang, "Enhanced access control with semantic context hierarchy tree for ubiquitous computing," *Int. J. Comput. Sci. Net. Security*, vol. 8, no. 10, pp. 114–120, Oct. 2008.

[15] W3C Member Submission, "SWRL: A semantic web rule language combining OWL and RuleML," May 2004.

[16] W3C Recommendation, "OWL 2 web ontology language: Structural specification and functional-style syntax (2nd ed.)," Dec. 2012.

[17] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics: Sci., Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.

[18] W3C Recommendation, "SPARQL query language for RDF," Jan. 2008.

Chapter 5

MASTERY: A multicontext-aware system that preserves the users' privacy

Title:	MASTERY: A multicontext-aware system that preserves the users' privacy
Authors:	Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez
Conference:	IEEE/IFIP Network Operations and Management Symposium
Publisher:	IEEE
Year:	2016
Month:	April
DOI:	10.1109/NOMS.2016.7502853
State:	Published

MASTERY: A multicontext-aware system that preserves the users' privacy

Alberto Huertas Celdrán*, Manuel Gil Pérez*, Félix J. García Clemente†, and Gregorio Martínez Pérez*
* Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071 Murcia, Spain
† Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, 30071 Murcia, Spain
Email: {alberto.huertas,mgilperez,fgarcia,gregorio}@um.es

***Abstract*—Users' privacy is a critical challenge for any information management system. The proliferation of mobile devices has promoted the use of context-aware solutions, making the protection of the users' information an even greater challenge. Addressing this requires a given mechanism that allows users to manage and control their personal information. In this sense, this paper proposes a privacy-preserving and context-aware system named MASTERY (*Multicontext-Aware System That prEserves the useRs' privacY*) that manages the privacy of the users' information in intra- and inter-context scenarios. MASTERY is a trusted third party that suggests to users a pool of privacy policies (profiles) aware to the context in which they are located, who can modify them according to their interests. These policies protect the privacy of the users' information being accessed from others without their consent. The information about users and contexts is managed by using semantic web techniques. This provides a common infrastructure that makes possible to represent, process, and share information between independent systems easily.**

I. Introduction

Technology advances have increased using the information management systems over the last decades, which makes us have to face privacy management on the users' information. We can find in the literature quite a number of solutions based on privacy policies to solve this challenge partially, where users can control *what* information is revealed to *whom* [1]. Yet, mobile devices has meant the emergence of new location-based services (LBS), since the location in a given environment about objects and people can provide useful information with which to develop such context-aware services [2]. But these services have introduced new challenges related to the management of the users' privacy in the case of considering their location.

The users of context-aware systems should be capable of managing the privacy of their personal information, locations, and information related to the environment or context in which they are located (intra-context scenario). This is an even more complex process when users move between several contexts, and there is an exchange of information between them (inter-context scenario). This raises new challenging questions about:

i) How users would have to manage and protect the information they want to share.

ii) Who should manage the privacy with respect to the information of the contexts.

iii) What information should be shared between different and independent contexts.

iv) Who can access the context-aware information.

978-1-5090-0223-8/16/$31.00 © 2016 IEEE

To conduct the previous tasks, the main contribution of this paper is a trusted third party named MASTERY (Multicontext-Aware System That prEserves the useRs privacY) that manages the privacy of the users' information in multicontext scenarios (inter and intra-context scenarios), incorporating the user consent to reveal his/her personal information. To this end, MASTERY suggests to the users several sets of privacy preserving and context-aware policies, called *profiles*. The users of MASTERY then choose the most suitable profile according to their interests in the surrounded context. After that, the users will be able to modify the selected profile adding, deleting, or modifying some of the policies that shape it. Intra- and inter-context policies form the profiles that allow the users to protect their location, personal information, activities, and context-aware information. This information is managed in MASTERY by using semantic web techniques, which provide us a common infrastructure that makes possible to share information between independent systems. In this sense, ontologies shape the users and the context-aware information; semantic rules define the privacy policies that form the context-aware profiles; and semantic reasoning infers new knowledge and decides *what*, *where*, *when*, *how*, and to *whom* the information is revealed.

The remainder of this paper is organized as follows. In Section II, we analyze the main related works with respect to other privacy-preserving solutions. A motivating example is presented in Section III, which will be used through the paper to introduce all concepts related to our proposal. The privacy-related policies are presented in Section IV, whereas Section V details the workflow of MASTERY. In Section VI, we show the technologies used by MASTERY in order to manage the privacy of the users' information. Finally, conclusions and future works are drawn in Section VII.

II. Related work

Several works proposed in the last decade let users define privacy-preserving policies with which they could control their personal information. For example, the authors of [3] proposed a Location-aware Role-Based Access Control (LRBAC) model that considered the users' location so as to allow or deny the access to their own information. LocServ [4] was another privacy-preserving policy system to protect the users' location considering the place where they were, the service at which the information was revealed, and the date when the information was shared. Another privacy-preserving system was SeCoMan [5], which proposed operational, authorization, and location policies with which to protect the users' location in the context in which the users were located. PRECISE [6] was an extension of SeCoMan, but oriented to cloud computing, which provided context-aware recommendations preserving the users' location and identity. In addition to all the previous features, MASTERY (the privacy-preserving and context-aware system presented in this paper) also protects the information related to the context in which the users were located.

The previous systems should protect not only the location of the users, but also the information regarding the contexts of users on the move. In that sense, CoBrA [7] protected the privacy of its users by using rules, with which infer whether the users had the right permissions to share and/or receive information in spaces composed of smart agents, devices, and sensors. CoPS [8] was another system that organized the policies into three different hierarchical levels, considering two different approaches to accept or deny the exchange of information. Another solution that protected the contextual information was PPCS [9], a semantic and policy-based system to protect the users' location, their activities, the sensor data, and the context of the users by making use of privacy policies. Dynamic information observed or inferred from the context, along with static information about the owner, was also taken into account to make access control decisions.

Despite the progress made by previous solutions, none of them allowed users to manage their own policies. In this sense, CAPRIS [10] was a context aware system that suggested privacy-preserving profiles considering the context in which the users were located. CAPRIS allowed users to supervise what, where, when, how, and to whom they wanted to reveal their personal information, activity, and information oriented to the context in which they were located. Controlling the privacy of the users' information between independent contexts is a requirement missing by the solutions mentioned earlier. In this sense, a privacy-preserving and multicontext health care oriented solution was proposed in [11]. This solution considered different entities such as users, roles, data, actions, purpose, obligations, and conditions in order to protect the privacy of the users' information in inter-context scenarios. Although this solution covered the inter-context privacy, it lacked a context-aware middleware in charge of discovering the context in which users were located.

Considering the drawbacks of the solutions commented earlier, MASTERY discovers the user's context and provides a multicontext policy-based mechanism in charge of managing the users' information in the context in which the users were located, as well as between the different visited contexts.

III. A MOTIVATING EXAMPLE

This section shows an inter-context scenario with which to illustrate the privacy-related concerns that a user, named Andy, can find with respect to his private information. In Fig. 1, we graphically show this scenario, which is composed of three independent contexts distributed in two countries. CountryA has an Embassy of CountryB, CountryB has a hospital and, both contexts share an international Health Insurance Company. The exchange of information between actors is shown with a red line, and the Andy's movements with a black line.

Andy is a citizen of CountryA who plans to visit CountryB. Before leaving CountryA, Andy requests the visa to the Embassy of CountryB (which is located in CountryA), providing some information such as the days he will stay in CountryB, the places that he wants to visit, the purpose of the trip, personal information, etc. Once the visa is granted by CountryB, the information of the Andy's trip (date, itinerary, etc.) is stored by the CountryB's Embassy. After having the visa, Andy wants to have a health insurance valid in CountryB. Therefore, he goes to an international Health Insurance Company and purchases a Health Insurance.

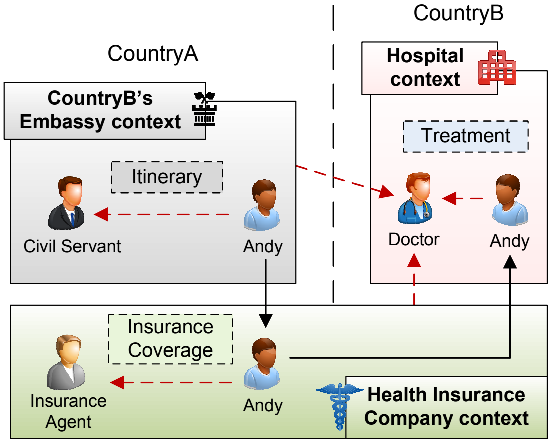


Fig. 1. Scenario with privacy concerns on the information of a given user

Few weeks later, when Andy is already visiting CountryB he has a heavy fever and decides to go to the hospital for a medical check-up with a Doctor. Once Andy is in the hospital, the Doctor needs to know if he has a Health Insurance valid for CountryB. In that sense, the Doctor accesses the information about the Andy's Health Insurance and checks its coverage. Here we have the first concern of privacy. When the Doctor accesses the Andy's Health Insurance, he can see private information about Andy like the Andy's bank account.

Once Andy is in the consulting room, the Doctor wants to know the places visited by Andy during the previous days. In order to know this information, the Doctor accesses Andy's Itinerary information, which belongs to the CountryB's Embassy. Here we have the second concern on privacy because the Doctor must not access information that belongs to another context. Furthermore, when he accesses the Itinerary he can see private information, such as the cities to be visited by Andy in the next days or the purpose of the trip.

Seeing Andy's Itinerary, the Doctor realizes that one week ago Andy stayed in a city with a high risk to get an infection. In this sense, the Doctor vaccines Andy and generates a new Treatment for him with the prescribed medicine. Once Andy comes back to CountryA, if the Health Insurance Company accesses the Andy's Treatment given in the Hospital context, there will be the third privacy concern.

Throughout this use case, we have introduced a number of privacy concerns related to the disclosure of private information between independent contexts. In order to avoid this privacy loss, we propose a management solution with which to protect the users' information.

IV. Preserving the users' privacy

Preserving the users' information is one of the key aspects for context-aware systems. The context is a concept combining the environment in which users are located, the information of nearby people and its objects, as well as any change in the previous terms. We think that context-aware solutions should preserve the users' location, personal information, activity, and information related to the environment in which they are located (e.g., medical records in a hospital context, as introduced in Section III). In this sense, privacy policies must allow the users to manage when, where, how, and to whom reveal information.

Our system MASTERY hides the information of the users by default, but allowing them the possibility of revealing such information through policies when necessary. Specifically, MASTERY provides the users with a group of context-aware and privacy-preserving policies called *profiles*. These profiles are specific to the context in which the users are located, aimed to protect the privacy of their personal information. When a user moves from a context to another, MASTERY provides him/her with several profiles about the new context, although the profile selected by the user can be modified at will by adding, modifying, or deleting its policies according to his/her personal interests.

A. Structure of the policies

The policies that are part of any given profile managed by MASTERY are composed of the following elements:

- *Id*: the policy identifier.
- *Type*: the kind of policy. We have defined four kinds of policies in MASTERY, which are explained below.
- *Maker*: the user or service administrator who defines the policy. Note that a given context can have one or more services, which can have administrators who manage the contextual information.
- *Target*: the user whose information is being managed by the policy.
- *Requester*: the user, or group of them, who request the information about the target.
- *Result*: the relationship that determines the access to the information.

Besides the above elements, we have also included some others to make policies richer and powerful; namely:

- *What*: the sensitive information revealed by the target such as location, personal information, activity, or even information related to a given context.
- *Where*: the place of the context in which the policy must be applied.
- *When*: the date and time during which the policy must be applied.
- *How*: the activity done by the target or the requester.

We classify the policies into two groups, since MASTERY allows the users to control their own information in intra- and inter-context scenarios. First, the *intra-context policies* are in charge of protecting the information in a given context and, secondly, the *inter-context policies* are in charge of protecting the information between different contexts. As an example of these policies, considering again the use case of Section III, the inter-policies would be used to control the Coverage of the Andy's Health Insurance and Itinerary in independent contexts. Both groups of intra- and inter-context policies are, in turn, composed of two different policies. First, the *disclosure policies* are in charge of indicating what information of the users can be shared. Secondly, the *reveal policies* with which the users can decide where, when, and how their information can be shared. These policies are explained in detail in Section IV-B and Section IV-C, respectively.

B. The intra-context policies

The *intra-disclosure policies* protect the users' information in a given context. They indicate *what* information the *target* wants to share with a given *requester*. The unique restriction of the intra-disclosure policies is that the *maker*, *requester*, and *what* fields must belong to the same context. It is important to note that intra-disclosure policies do not indicate where, when, or how the information is revealed.

On the other hand, the *intra-reveal policies* indicate where, when, and how the target's information can be shared. Note that the when and how elements are optional for this policy, which indicate that the policy will be applied at any time and anyway, respectively. This kind of intra-reveal policy is related to the previous intra-disclosure one, which are linked through the target and requester elements. In this sense, this policy activates the disclosure policies when the users are in certain places or contexts.

C. The inter-context policies

The *inter-disclosure policies* are aimed at preserving the exchange of information between different contexts. When a user of a given context wants access to information belonging to another context, it is necessary that the information's context owner leaves it accessible to be exchanged. In these policies, the *maker* must belong to the context of the information and the *what* field must contain information about the maker's context.

In the use case of Section III, we detected an information exchange between different contexts: the Doctor accessing the Insurance Coverage of Andy and the Itinerary of Andy. Therefore, in order to solve the first concern on privacy, the next policy indicates that Andy discloses the Coverage of his Health Insurance to the Hospital Staff. In this policy, the Insurance Agent decides who can access this information with an inter-disclosure policy.

> *Id*: **InsuranceInterD** ∧ *Type*: **InterDisclosure**[Hospital] ∧
> *Maker*: **InsuranceAgent** ∧ *Target*: **Andy** ∧
> *Requester*: **HospitalStaff** ∧ *What*: **InsuranceCoverage**
> → *Result*: **Disclosure**

Regarding the second privacy concern, the Civil Servant defines the inter-disclosure policy shown below indicating that Andy discloses his Itinerary to the Hospital Staff.

Id: **EmbassyInterD** ∧ *Type*: **InterDisclosure**[Hospital] ∧
Maker: **CivilServant** ∧ *Target*: **Andy** ∧
Requester: **HospitalStaff** ∧ *What*: **Itinerary**
→ *Result*: **Disclosure**

Finally, the third privacy concern found in Section III is solved by the next policy, which discloses the Andy's Treatment given by the Doctor of the Hospital context to the Health Insurance Staff.

Id: **HospitalInterD** ∧
Type: **InterDisclosure**[HealthInsuranceCompany] ∧
Maker: **Doctor** ∧ *Target*: **Andy** ∧
Requester: **HealthInsuranceStaff** ∧ *What*: **Treatment**
→ *Result*: **Disclosure**

On the other hand, the inter-reveal policies are very similar to the intra-reveal policies, although the former are totally oriented to operate between contexts. Such inter-reveal policies indicate where, when, and how the information (previously established by the inter-disclosure policies) is revealed. In the use case of Section III, in order to allow the Doctor to access the Coverage of the Andy's Health Insurance, the Hospital Administrator needs to define an inter-reveal policy to reveal the Coverage of the Foreign Patient (Andy) to the Doctor.

Id: **1HospitalInterR** ∧
Type: **InterReveal**[HealthInsuranceCompany] ∧
Maker: **HospitalAdmin** ∧ *Target*: **ForeignPatient** ∧
Requester: **Doctor** ∧ *Where*: **ConsultingRoom**
→ *Result*: **Reveal**

Following the use case of Section III, in order to allow the Doctor to access the Itinerary of Andy, we need the inter-reveal policy that reveals the information allowed by the policy EmbassyInterD. This policy defines that when a Foreign Patient (like Andy) is in the hospital, his/her Itinerary is revealed to the Doctor. Although this policy only defines information about the Hospital context, the information to be revealed belongs to the CountryB's Embassy context.

Id: **2HospitalInterR** ∧
Type: **InterReveal**[CountryAEmbassy] ∧
Maker: **HospitalAdmin** ∧ *Target*: **ForeignPatient** ∧
Requester: **Doctor** ∧ *Where*: **Hospital**
→ *Result*: **Reveal**

Finally, if Andy comes back to CountryA and the agent of the Health Insurance Company accesses the information of the Andy's Treatment, the administrator of this context has to define the next inter-reveal policy.

Id: **InsuranceInterR** ∧ *Type*: **InterReveal**[Hospital] ∧
Maker: **InsuranceAdmin** ∧ *Target*: **ForeignPatient** ∧
Requester: **InsuranceAgent** ∧
Where: **HealthInsuranceCompany**
→ *Result*: **Reveal**

V. Users' privacy management by MASTERY

This section shows how MASTERY manages the privacy of the users' information by using the use case of Section III. When Andy is in the Embassy of CountryB to get the visa of CountryB, as Andy is concerned about what of his information can be revealed, he uses MASTERY to protect his information. Once Andy is in the embassy, MASTERY detects him in accordance with the information provided by the embassy's location infrastructure. Then, MASTERY offers him several context-aware profiles, such as Visitant or Citizen. These profiles are suggested by the CountryB's Embassy Administrator to MASTERY, and MASTERY is in charge of suggesting them to Andy. Andy chooses the Visitant profile, containing several policies, and MASTERY stores and activates that profile for the CountryB's Embassy context. When Andy is with the Civil Servant, the last one asks Andy for the trip information (purpose, itinerary, date, etc.) and MASTERY stores this information. Finally, the Civil Servant defines the policy EmbassyInterD (presented in Section IV-C) to provide the Itinerary information to the Hospital context.

When Andy receives the visa, before leaving CountryA, Andy changes again of context and goes to the international Health Insurance Company in order to get a Health Insurance valid for CountryB. When he is in the insurance company, MASTERY provides him with several profiles and he selects the most appropriate for him. Once Andy decides the coverage of his insurance, he pays it and the Insurance Agent stores the information using MASTERY. Finally, the Insurance Agent defines and stores in MASTERY the policy InsuranceInterD (explained in Section IV-C). This policy provides the coverage of the Andy's Health Insurance to the Hospital context.

Few weeks later, when Andy is in CountryB, he changes of context for going to the hospital because he has fever. Fig. 2 shows the sequence diagram when Andy is in the Hospital context and the Doctor wants to access the Andy's information. When Andy is there, MASTERY detects him thanks to the hospital's location infrastructure. Then, the Hospital Administrator suggests to MASTERY several context-aware profiles, which are offered by MASTERY to Andy (steps 1, 2, and 3 in Fig. 2). Andy chooses the Foreign Patient profile (step 4) that contains policies to protect his personal information in the Hospital context and MASTERY stores the profile (step 5). As the Doctor needs to access Andy's Itinerary and Insurance Coverage, the administrator of the Hospital modifies the Andy's Foreign profile in order to add the policies 1HospitalInterR and 2HospitalInterR (defined in Section IV-C) that allows the access to this information (step 6). It is worth noting that for revealing the Andy's Itinerary is necessary that the Civil Servant defines the inter-disclosure policy EmbassyInterD and the Hospital defines the inter-reveal police 2HospitalInterR. Without one of these policies, the Andy's Itinerary is not revealed to the Doctor.

For the Health Insurance Coverage, it is necessary exactly the same process. Once the profile is updated by MASTERY (step 7), the Doctor uses the Hospital application to ask MASTERY for the Andy's Health Insurance Coverage (step 8). The engine component manages the Andy's profiles and detects that the Andy's Foreign Patient profile and the Health Insurance context allows the Doctor to know the Coverage of the Andy's Health Insurance (step 9). Then, MASTERY

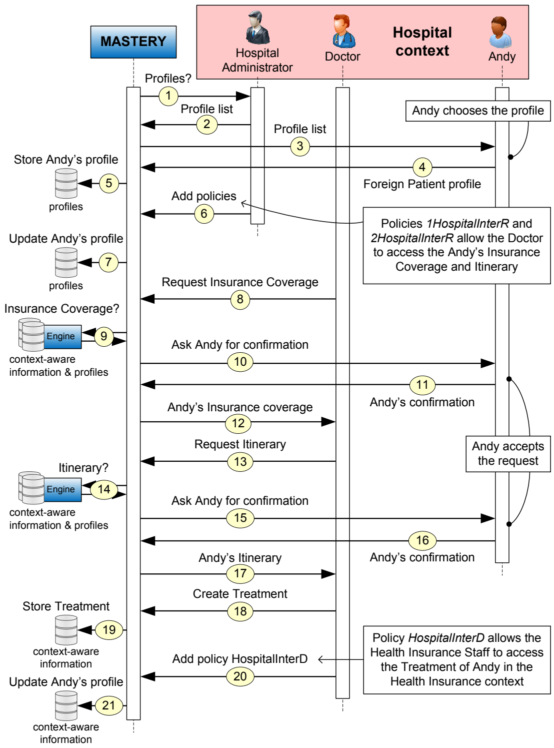

MASTERY
Hospital Administrator
Doctor
Hospital context
Andy
Profiles?
1
Profile list
2
Profile list
3
Andy chooses the profile
4
Foreign Patient profile
Store Andy's profile
5
profiles
Add policies
6
Policies 1HospitalInterR and 2HospitalInterR allow the Doctor to access the Andy's Insurance Coverage and Itinerary
Update Andy's profile
7
profiles
Request Insurance Coverage
8
Insurance Coverage?
Engine
9
context-aware information & profiles
Ask Andy for confirmation
10
11
Andy's confirmation
Andy's Insurance coverage
12
Request Itinerary
13
Andy accepts the request
Itinerary?
Engine
14
context-aware information & profiles
Ask Andy for confirmation
15
16
Andy's confirmation
Andy's Itinerary
17
Create Treatment
18
Store Treatment
19
context-aware information
Policy HospitalInterD allows the Health Insurance Staff to access the Treatment of Andy in the Health Insurance context
Add policy HospitalInterD
20
Update Andy's profile
21
context-aware information

Fig. 2. Sequence diagram of MASTERY in the Hospital context

asks Andy for confirmation (step 10) and, when it receives the acknowledgment response (step 11), MASTERY provides the Doctor with the Coverage of the Health Insurance (step 12). In order to obtain the Itinerary (steps from 13 to 17), the process is the same as the one commented previously. Finally, the Doctor generates the Andy's treatment with the prescribed medicine (steps 18 and 19), and defines the policy HospitalInterD, which allows the Health Insurance context to access the Andy's Treatment.

It is worth noting that to ensure the MASTERY's usability, the users just interact with the system to choose the preferred profiles and consent the release of their information.

VI. Technology deployment in MASTERY

This section shows the technologies used by MASTERY to manage the privacy of the users' information. Specifically, the information about users and contexts is shaped with ontologies defined in OWL 2 (Web Ontology Language) [12], which have been generated with Protégé [13]. We have chosen OWL 2 rather than other languages like RDF, RDFS, or DAML+OIL because OWL 2 is more expressive than the rest. It was specifically designed as an ontology language, it is an open standard, and it is the main ontology language used nowadays in semantic web. Specifically, the ontologies allow MASTERY to shape concepts such as the place where the users are (e.g., the Consulting Room in the Hospital context), the users' roles (e.g., Doctor in the Hospital context), or the context-aware information (e.g., the Itinerary or Insurance Coverage of Andy). On the other hand, the policies that are part of the users' privacy profiles are expressed in SWRL (Semantic Web Rule Language) [14]. This language includes a type of axiom, called Horn clause logic, of the form if... then..., and it is the most used in semantic web.

Fig. 3 shows the process followed by MASTERY when it receives a request to share information. In order to decide what information is revealed, where, when, how, and to whom, the MASTERY's engine has a *Reasoner* module that uses Pellet [15]. We think that Pellet is the appropriate reasoner because it supports semantic rules expressed in the SWRL language. The Reasoner module receives the ontological models generated by the *Interpreter* component and returns the inferred models with new knowledge. The Interpreter uses the Jena API [16] to generate the ontological models with the information shaped in the ontologies and policies. Jena is a free and open source Java framework, which supports OWL and Pellet. The *Engine* component translates the queries performed by the requester into SPARQL queries [17], which are applied to the inferred model to get the corresponding result. Finally, the *Confirmation Decision Point* module asks the target before revealing his/her information.

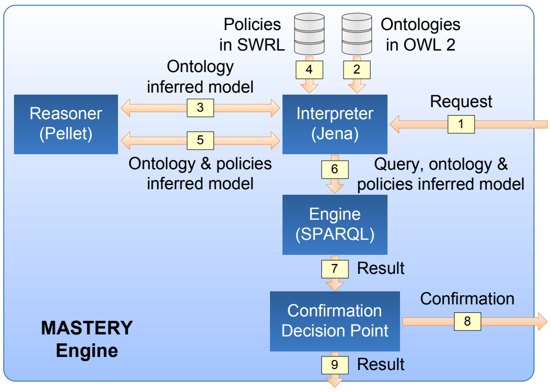
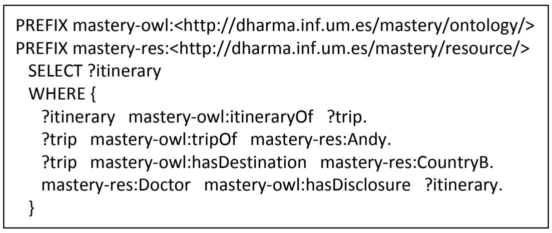


Fig. 3. Reasoning and query processes

When a given user makes a request (step 1 Fig. 3), e.g., when the Doctor wants to access Andy's Itinerary, the Interpreter generates an ontological model from the context-aware ontologies (step 2). This model contains certain information about the context in which the target is located, the hospital, and the users allocated there (Andy and the Doctor). The model is sent to the Reasoner to obtain the inferred model with the new information inferred from the ontologies (step 3). Once the Interpreter receives the inferred model, it is updated with the information provided by the Reasoner (steps 4 and 5). According to the disclosure and reveal policies of the Andy's profiles (policies EmbassyInterD and 2HospitalInterR), this model infers the decision of sharing (or not sharing) Andy's Itinerary. If there is a conflict in the step 5 with the policies that form the active profiles, the Reasoner gives higher priority to the policies in the following order: the ones defined by the users, the ones defined by the administrators, and the ones defined by other users of the service. Once this process is complete, the Interpreter invokes the Engine with the decision of sharing the requested information (step 6). In case that the information cannot be revealed, the process finishes in the step 5. The Engine component applies the SPARQL shown in Fig. 4 to the inferred model and obtains Andy's Itinerary.



```
PREFIX mastery-owl:<http://dharma.inf.um.es/mastery/ontology/>
PREFIX mastery-res:<http://dharma.inf.um.es/mastery/resource/>
  SELECT ?itinerary
  WHERE {
    ?itinerary  mastery-owl:itineraryOf  ?trip.
    ?trip  mastery-owl:tripOf  mastery-res:Andy.
    ?trip  mastery-owl:hasDestination  mastery-res:CountryB.
    mastery-res:Doctor  mastery-owl:hasDisclosure  ?itinerary.
  }
```

Fig. 4. SPARQL query to obtain the Andy's Itinerary

Specifically, the query first obtains the Andy's trips and filters out the trips which are not related to CountryB. Once the CountryB's trip is obtained, the Engine gets the itinerary and checks if the Doctor has authorization to obtain the information. To know if the Doctor has authorization, the Engine checks if the Doctor and the Andy's itinerary are related throw the hasDisclosure property. If so, the Confirmation Decision Point component finally receives the result from the Engine (step 7) and asks Andy for confirmation to share his Itinerary (step 8). If the confirmation is provided by Andy, the result is sent to the Doctor (step 9).

VII. Conclusion and future work

We have proposed in this paper a trusted privacy-preserving and context-aware solution that manages the privacy of the users' information in intra- and inter-context scenarios. Considering the contexts in which the users are located, our system called MASTERY (Multicontext-Aware System That prEserves the useRs' privacY) provides several context-aware profiles for protecting the users' information. These profiles are formed by a pool of privacy policies so as to protect the users' location, personal information, the activities they are doing at any given time, and the information oriented to the context in which they are located. The users of MASTERY can modify the profiles at will by adding, deleting, and modifying their policies to control what, where, when, how, and to whom the users want to reveal their information.

As next steps of this research, we plan to provide more intelligence to MASTERY in order to offer customized profiles, considering the actions performed by the users on previous visits to the context. Finally, we also plan the integration of MASTERY with other systems running in the contexts in which our solution is not deployed.

Acknowledgment

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2014, the Spanish MICINN (project DHARMA, Dynamic Heterogeneous Threats Risk Management and Assessment, with code TIN2014-59023-C2-1-R), and the European Commission (FEDER/ERDF).

References

[1] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao, "Understanding and capturing people's privacy policies in a mobile social networking application," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 401–412, Aug. 2009.

[2] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proceedings of the 1st Workshop on Mobile Computing Systems and Applications*, Dec. 1994, pp. 85–90.

[3] I. Ray, M. Kumar, and L. Yu, "LRBAC: A location-aware role-based access control model," in *Proceedings of the 2nd International Conference on Information Systems Security*, Dec. 2006, pp. 147–161.

[4] G. Myles, A. Friday, and N. Davies, "Preserving privacy in environments with location-based applications," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 56–64, Jan. 2003.

[5] A. Huertas Celdrán, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, "SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications," *IEEE Systems Journal*, To appear.

[6] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez, "PRECISE: Privacy-aware recommender based on context information for cloud service environments," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 90–96, Aug. 2014.

[7] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, vol. 18, no. 03, pp. 197–207, Sep. 2003.

[8] V. Sacramento, M. Endler, and F. N. Nascimento, "A privacy service for context-aware mobile computing," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks*, Sep. 2005, pp. 182–193.

[9] P. Jagtap, A. Joshi, T. Finin, and L. Zavala, "Preserving privacy in context-aware systems," in *Proceedings of the 5th IEEE International Conference on Semantic Computing*, Sep. 2011, pp. 149–153.

[10] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez, "What private information are you disclosing? a privacy-preserving system supervised by yourself," in *Proceedings of the 6th International Symposium on Cyberspace Safety and Security*, Aug. 2014, pp. 1221–1228.

[11] L. D. Martino, Q. Ni, D. Lin, and E. Bertino, "Multi-domain and privacy-aware role based access control in eHealth," in *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, Jan. 2008, pp. 131–134.

[12] W3C Recommendation, "OWL 2 web ontology language: Structural specification and functional-style syntax (2nd ed.)," Dec. 2012.

[13] Stanford Center for Biomedical Informatics Research, "Protégé: A free, open source ontology editor and knowledge-base framework," [Online]. Available: http://protege.stanford.edu.

[14] W3C Member Submission, "SWRL: A semantic web rule language combining OWL and RuleML," May 2004.

[15] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.

[16] The Apache Software Foundation, "The Apache Jena2 ontology API," [Online]. Available: http://jena.apache.org/documentation/ontology.

[17] W3C Recommendation, "SPARQL query language for RDF," Jan. 2008.

Chapter 6

Policy-based management for green mobile networks through Software-Defined Networking

Title:	Policy-based management for green mobile networks through Software-Defined Networking
Authors:	Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez
Journal:	Mobile Networks and Applications
Impact factor:	1.538 Q2 (2015)
Publisher:	Springer
Year:	2016
Month:	December
DOI:	10.1007/s11036-016-0783-8
State:	Published Online on 05 December 2016

Mobile Netw Appl
DOI 10.1007/s11036-016-0783-8

Policy-Based Management for Green Mobile Networks Through Software-Defined Networking

Alberto Huertas Celdrán[1] · Manuel Gil Pérez[1] · Félix J. García Clemente[2] · Gregorio Martínez Pérez[1]

Abstract Traditional networks are characterized by wasting considerable amount of energy that could be reduced drastically. The challenge of energy saving should be managed efficiently, where the mobility of users and services are nominated to play a significant role as well as the use of the Software Defined Networking (SDN) paradigm. Besides the network management supported by the SDN paradigm, we highlight the management of the network infrastructure at run-time, considering aspects like the energy efficiency. In this paper, we present an energy-aware and policy-based system oriented to the SDN paradigm, which allows managing the network infrastructure dynamically at run-time and on demand through policies. With these policies, any network using our solution will be able to reduce energy consumption by switching on/off its resources when they are inefficient, and creating virtualized network resources like proxies to reduce the network traffic. The experiments conducted demonstrate how the energy consumption is reduced when enforcing the proposed policies, considering aspects such as the number of base stations, their cell sizes, and the number of active devices in a given time, among other.

Keywords Energy-aware policies · Software defined networking · Users' mobility · Virtualization

1 Introduction

Energy saving is a critical task in the design and effective management of any computer network. The energy-aware challenge has been a cornerstone in traditional networks, in which the whole infrastructure needs to consume energy to provide users with services. All these services are invariant over time, because the network does not take into account aspects such as the current load traffic, the number of active devices, the date and time, or the patterns of the users' behavior. This fact implies the inefficient use of the network infrastructure and therefore the waste of energy.

The *Software Defined Networking* (SDN) paradigm has recently emerged with the main goal of easing network management [1, 2]. This paradigm allows network administrators to configure and manage their network status at run-time and on demand, where the energy efficiency should be considered in addition to other areas of application such as the ones studied in [3, 4]. Despite the number of resources and facilities provided by the SDN paradigm, the mobility provided by current devices and services has hindered the management of the network infrastructure efficiently. Nowadays, mobile networks are not used at their full capacity, except in given peak times. This entails a high waste of energy in situations with low traffic, while the infrastructure remains optimized for maximum traffic load.

✉ Alberto Huertas Celdrán
alberto.huertas@um.es

Manuel Gil Pérez
mgilperez@um.es

Félix J. García Clemente
fgarcia@um.es

Gregorio Martínez Pérez
gregorio@um.es

[1] Departamento Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071, Murcia, Spain

[2] Departamento Ingeniería y Tecnología de Computadores, University of Murcia, 30071, Murcia, Spain

Published online: 05 December 2016

Important improvements in terms of energy saving can be achieved by managing the network resources at runtime, considering the mobility of users and services. In that sense, this paper at hand presents a mobility-aware policy-based system in charge of reducing the energy consumption in networks oriented to the SDN paradigm. The policies defined in our solution allow the SDN paradigm to switch on/off network resources when they are consuming energy in an inefficient way, as well as create virtualized network resources like proxies to reduce the network traffic generated by users consuming services close to the network infrastructure.

The remainder of the paper is structured as follows. Section 2 discusses the related work of other energy-aware solutions. A motivating example is presented in Section 3, which will be used through the paper at hand to introduce all concepts related to our proposal. The energy-aware policies and how they manage the energy consumption in our use case are presented in Section 4. Section 5 shows the ontology to shape the information as well as the technologies for the proposed architecture. Section 6 reports some experimental results to illustrate the performance of our proposal. Finally, conclusions and future work are drawn in Section 7.

2 Related work

A recent survey can be found in the current literature, in which a deep comparison of a number of green mobile network approaches was performed [5]. In this paper, the authors recognized that more than the 50 % of the energy consumed by mobile networks was produced in base stations (BS). In order to reduce this consumption, the analyzed solutions were classified into five different categories: improvement on the hardware components, sleep mode techniques, optimization in the radio transmission process, network planning and deployment, and adoption of renewable energy resources.

From the *energy efficiency* perspective, the solution presented in [6] quantifies power consumption of mobile communication systems. It establishes that there is a high potential to reduce the energy consumption when improving the energy efficiency of the BSs at low traffic load. Another proposal to enhance efficiency of power amplifier for wireless BSs was proposed in [7]. On the other hand, the second category was focused on *sleep mode* techniques, which turn on/off network resources during non-peak traffic hour selectively. The variation of the traffic patterns over time, in order to decide when BSs should sleep or not, was considered in [8]. Another solution was proposed in [9], where a control mechanism enables small cells to switch off all components while not serving active connections. In order to speed up the decision process of switching on/off BSs, a transfer actor-critic algorithm (TACT) was proposed in [10], making use of historical data from neighbor regions.

The *cell zooming* solution proposed in [11] provided a level of flexibility higher than the previous sleep mode proposals. This solution was capable of adjusting cell sizes according to different aspects such as the traffic load, the users' requirements, or the channel conditions. It allowed balancing the traffic load by zooming in/out the cell, in order to reduce/increase the coverage area to avoid congestion. In [12], it was studied the effect of the cell size on the energy consumed by BSs that make use of current technologies. Its authors recognized that the optimal cell size from an energy perspective depends on several factors such as the technology of the BS, the data rates, and the traffic demands. They also proposed a schema to adjust the cell sizes dynamically for saving energy. On the other hand, the concept of area power consumption was introduced in [13], in order to evaluate the impact of deployment strategies on power consumption of mobile radio networks. They considered different numbers of micro BSs, in addition to the conventional macro sizes. Their outcomes indicated that it had a moderate effect on power consumption by using micro BSs in scenarios with full traffic load.

Regarding the *radio transmission* process, a number of approaches have been proposed to efficiently make use of resources in time, frequency, and spatial domains. The multiple network access interfaces of the mobile devices were used in [14], with which they can cooperate in sending data packets to the BS. Another solution oriented to cognitive radio transmission was proposed in [15], where a multi-hop cognitive cellular network architecture facilitates the ever exploding data transmissions in cellular networks. In this same research area, a comparative study of the energy consumption of several wireless network access points can be found in [16].

Deploying low power-consuming small cells (micro, pico, and femto cells) in reduced areas with dense traffic was proposed in [17]. It analyzed the energy efficiency in small cell networks, using a random spatial network model in which BSs and users were modeled as two independent spatial Poisson Point Processes (PPP).

Finally, the last category identified in [5] included several approaches adopting *renewable energy resources*. Among them, in [18] it was proposed an energy cooperation model between cellular BSs with hybrid energy sources, limited storage, and a connecting power line.

So far, we have seen solutions oriented to reduce the energy consumption in BSs. However, data centers are also consuming huge amount of energy for computing. Many designs in data centers rely on virtualization to eliminate the hardware constraints and make the computation more flexible and efficient. In this context, a mechanism able to transfer virtualized resources from one physical machine to another was proposed in [19], without interrupting its services. In other solutions, as the one proposed in [20], an event-driven network control framework that used high-level policies was presented to configure and manage the network state, which are translated into a set of forwarding rules to be managed by the controller.

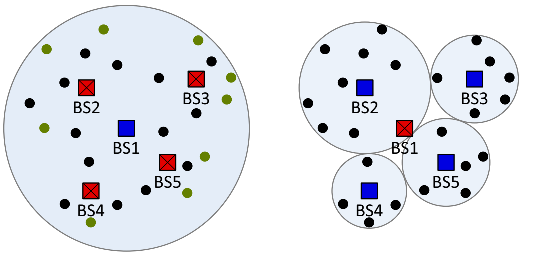
As commented on earlier, the previous solutions use different metrics and techniques in order to save energy or reduce the energy consumption. In [21], it was presented a survey where a high number of green proposals show researches, experiments, deployments, and evaluations of green metrics for mobile networks. These metrics can be categorized into two types: *equipment-level* and *facility-level* metrics. Equipment-level metrics are focused on the lower-level energy efficiency. They consider individual pieces of mobile networks such as, for example, Energy Consumption Rating (ECR), Consumer Consumption Rations (CCR), and Telecommunications Energy Efficiency Ratio (TEER). Instead, facility-level metrics are related to high-level energy efficiency. They are focused in a macro point of view, such as data center metrics, including Power Usage Effectiveness (PUE), Data Center Infrastructure Efficiency (DCiE), and Data Center Productivity (DCP).

Despite the previous solutions improved energy efficiency with novel techniques and metrics, the decision of applying them was conducted by administrators. In that sense, our solution is able to decide at run-time and on demand when the appropriate techniques should be applied by using our energy-aware policies.

3 Use case in mobile scenarios

In this section, we present a use case to illustrate how our solution manages the energy efficiency at run-time in networks providing services to devices with a high mobility oriented to the SDN paradigm. The proposed use case is composed of five BSs distributed along a given area. Among them, one (BS1) is providing devices with different services, whereas the rest are asleep (BS2, BS3, BS4, and BS5) for saving energy. Figure 1a shows a given time when new devices, marked as green (lighter) points, appear in the BS1's cover area, and start consuming services and move away from BS1. Its cell size is then automatically increased to continue providing services. This situation generates an energy-aware concern, because BS1 needs to consume more energy in increasing its cell size for providing the services.

In order to fix this problem, our solution switches on the asleep BSs located close to the mobile devices and switches off BS1, as shown in Fig. 1b. Despite of switching on new BSs, they consume less energy than BS1. This is because users are close to them and their coverage cells are smaller than the one needed by BS1. (It is worthy to note that this assertion in energy consumption is later demonstrated in Section 6.) The sizes of the coverage cells are computed by the network infrastructure dynamically, using for example the load balancing technique of [11]. This technique automatically increases or decreases the size of BSs' cells by considering the energy efficiency. It is worth noting that if users go back closer to BS1, or even the number of active devices consuming services decreases, our solution will switch on BS1 and switch off the rest of BSs.



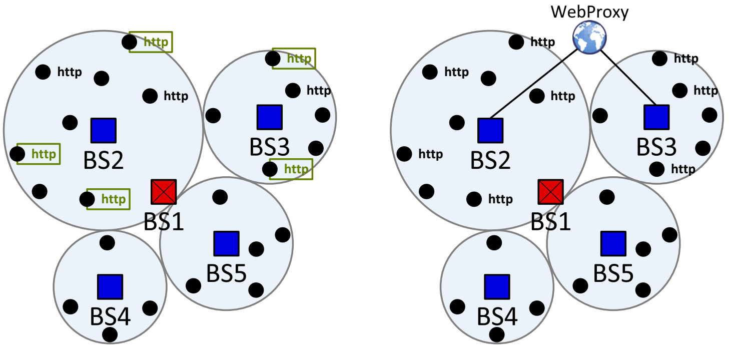
(a) Initial scenario (b) Switched ON/OFF scenario

Fig. 1 Base stations (BS) are switched on/off to consume energy in an efficient way. *Blue* (not strikethrough) squares represent BSs switched on, *red* (strikethrough) squares switched off, *black* (darker) circles are devices consuming services, and *green* (lighter) circles newcomers

Following the use case, Fig. 2a shows the moment when several active devices located at the cover areas of BS2 and BS3 start visiting certain websites, thereby consuming http services. This situation generates a new energy-aware concern, because the network infrastructure has to transmit and receive packages. In order to reduce this energy consumption, our solution is capable of deploying virtualized resources in the network. A *virtualized web proxy* is an example of virtualized infrastructure that allows temporary storage (caching) of information. For example, virtualized web proxies can cache web pages to reduce network traffic in future http requests, as it will not be necessary to retrieve again that information from the web service in future requests. Reducing network traffic will entail a reduction in energy consumption, both for the devices and for the network infrastructure [22]. Within our use case, Fig. 2b shows how a virtualized web proxy is created close to BS2 and BS3, in order to provide active devices of BS2 and BS3 with the http service. The reverse process (dismantling the virtualized proxy) is performed by our solution when users stop consuming that service.

4 Energy-aware and management-oriented policies

Our solution allows the SDN paradigm to manage at run-time the energy consumption of the network infrastructure



(a) Initial scenario (b) Managed scenario

Fig. 2 Virtualized web proxies are created/dismantled to reduce energy consumption. *Blue* (not strikethrough) squares represent BSs switched on, *red* (strikethrough) squares switched off, *black* (not boxed) text makes reference to devices consuming http services, and *green* (boxed) text to devices starting to consume http services

by using policies. Among the different existing policies, we make use of energy-aware and management-oriented policies. The network administrator is in charge of defining the set of policies that will decide the list of potential actions to be taken by the SDN components, in accordance with the energy consumption, the users' mobility, and the network statistics.

Our policies are comprised by the following elements: the *type* of policy (e.g., switching, virtualization, etc.); the network *resource*, whose information is currently being managed (e.g., base station, switch, service, etc.); the *metric* with which the network state can be evaluated (e.g., performance indicators like PI_{urban} [23] or ABf, which indicate the average power consumption in peak hour and the Average of Bytes per flow, respectively); the *location* or region where the policy will be enforced (e.g., geographic position, area, etc.); the *date* or the period of time at which the policy will be applied (e.g., hour, timestamp, etc.); and the *result* or set of actions to be carried out over the network once the policy is applied (e.g., switch, create a proxy, etc.).

The previous set of elements defining a given policy can be represented as follows.

$$\boxed{Type \wedge Resource \wedge Metric \wedge Location \wedge Date \rightarrow Result}$$

4.1 Policies to ensure the energy efficiency

We introduce below the two kinds of policies required to manage the energy consumption of the previous use case. It is worthy to note that other policies could also be defined, since the proposed solution is extensible.

4.1.1 Switching policies

These policies allow the SDN paradigm to switch on/off the network resources located at specific locations. By using this kind of policies, the network administrator can indicate that when a given network *resource* is making an inefficient use of the energy (e.g., the PI_{urban} *metric* is below a given threshold, which was defined by the network administrator beforehand), the *result* is to switch on or off the network elements whose *location* is close to the inefficient ones.

In addition, this kind of policies can also be applied in a proactive way, in case of knowing the patterns of energy consumption or the users' mobility. In switching policies, the *metric*, *location*, and *date* parameters are optional. It is important to note that once the network resources are switched on/off, the network traffic should be balanced in the network infrastructure.

4.1.2 Virtualization policies

These policies allow creating or dismantling the virtualized network resources to optimize the energy consumption of the network infrastructure as well as the devices when users consume certain services.

The virtualization policies can save energy in many different ways. For example, virtualized infrastructure can be created to balance the load traffic and optimize the use of the resources, or even in charge of reducing the network traffic and thus the time to consume services. By using these policies, the network administrator can indicate that when a given network (*resource*) is making an inefficient use of the energy (e.g., PI_{urban} *metric* is below another threshold), the *result* is to create a given number of virtualized resources whose *location* is close to the inefficient ones. As before, the *metric*, *location*, and *date* parameters are optional in this kind of policies.

4.2 How to reduce energy consumption in mobile scenarios

How our solution can be used to reduce the consumed energy is explained below, in which we use the use case presented in Section 3.

In order to switch on/off the network resources, as depicted in Fig. 1b, our solution defines two generic *switching policies*. They are shown below as an example. These policies indicate that when the *PIUrban* value of any BS is within the *AlarmA* range values (the range of this alarm is set by the network administrator), the network should switch on the BSs located at the same area that the inefficient one and switch off the latter. It is important to note that, despite these examples of policies use the metric and location parameters, they are optional in our policies.

The policy shown below is in charge of switching on the BSs located close to the inefficient one, as shown in Fig. 1 as an example of a proof of concept.

```
Type(#Switching) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBS(?area,?neighborBSs) ∧
integer[PIUrban in #AlarmA] hasPIUrban(?bs)
→ hasStatus(?neighborBSs,#ON)
```

Once the previous policy is enforced, the next action is to implement the following policy in order to switch off the inefficient BS.

```
Type(#Switching) ∧ BaseStation(?bs) ∧
integer[PIUrban in #AlarmA] hasPIUrban(?bs)
→ hasStatus(?bs,#OFF)
```

In our use case, these two policies switch on BS2, BS3, BS4, and BS5 and switch off BS1. It is important to note that the cell sizes of the new active BSs are calculated by the network infrastructure, by making use of balancing techniques like the one proposed in [11].

Following the use case proposed in Section 3, we have defined two *virtualization policies* to reduce the energy consumption in the network infrastructure and in the users' devices. The first virtualization policy is in charge of creating and associating virtualized web proxies to reduce the traffic of the http service in BSs located close to the inefficient ones. This inefficiency is measured in our policy by using the values of the PI_{urban} and ABf metrics. Specifically, when the values of these two metrics are within the *AlarmB* and *AlarmC* range values, a web proxy will be created (if not any) and it will be associated to the BS.

```
Type(#Virtualization) ∧ BaseStation(?bs) ∧
WebProxy(?webP) ∧
integer[PIUrban in #AlarmB] hasPIUrban(?bs) ∧
integer[ABf in #AlarmC] hasABf(#HttpService)
→ associate(?bs,?webP)
```

Following the use case, the previous policy creates a virtualized web proxy associated to BS2 and BS3, in order to reduce the http traffic generated by the users located in the cover areas of these BSs.

Finally, the web proxy is disassociated once users stop consuming the http service, as well as being dismantled if it is not used anymore. This is shown below by the next policy.

```
Type(#Virtualization) ∧ BaseStation(?bs) ∧
WebProxy(?webP) ∧
integer[PIUrban in #AlarmD] hasPIUrban(?bs) ∧
integer[ABf in #AlarmE] hasABf(#HttpService)
→ disassociate(?bs,?webP)
```

5 Policy deployment infrastructure

This section shows how our solution models the network information through ontologies. Moreover, we also propose an architecture with which to manage the ontology proposed in this section as well as the policies defined in Section 4, with the aim of saving energy in network resources and users' devices.

5.1 The mobile network ontology

We describe here the main ontology that is managed by our solution, which models all the concepts introduced along the paper. With this ontology, we provide a set of primitives with which to describe a collection of the elements that belong to the mobile network topic and the relationships among them. This ontology is shown in Fig. 3, which is focused on shaping the information of the base stations and the devices. All that information is shaped in OWL 2 (Web Ontology Language) [24], using Protégé [25] for its generation.

The mobile network ontology is categorized into two different, but related topics: base station and device. The top-level class in the former topic is *BaseStation*, which is one of the components where we reduce the energy consumption in mobile networks. A base station is a kind of the network *InfrastructureElement*, which is composed of several *Components*, and it can in turn use different *Technologies* for its design. As a proof of concept, we have modeled four of the main components that compose any kind of base station.

It is worthy to note that all elements commented so far, and the ones introduced below, have been modeled by following the Common Information Model (CIM) language defined by the DMTF as a standard [26]. This language provides a common definition of management information for systems, networks, applications, and services. Due to that, all the elements herein defined inherit from the *ManagedElement* class, defined in CIM as the main element for a given model.

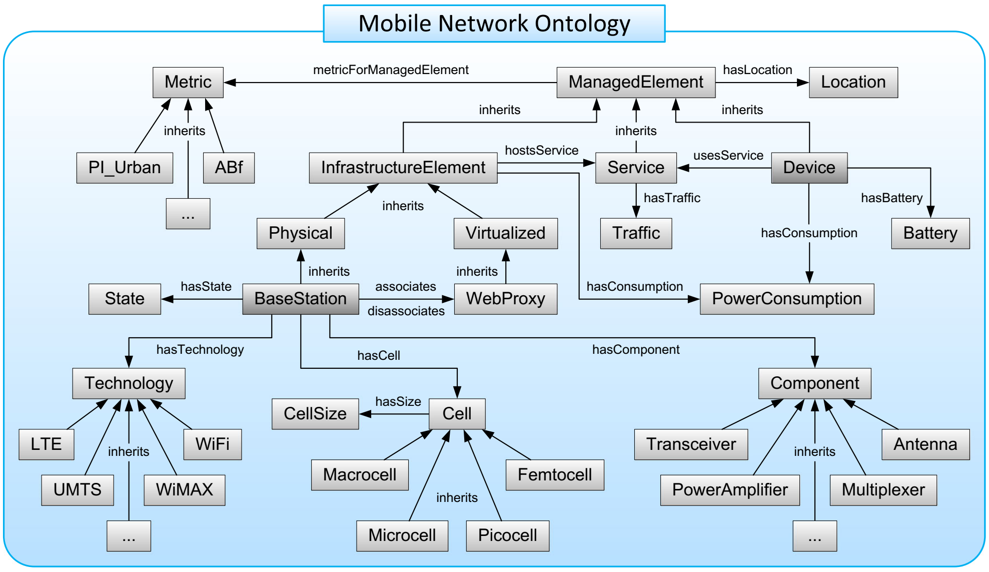


Fig. 3 Elements of the proposed ontology for shaping a mobile network

The Component class has four predefined subclasses: *Antenna* makes reference to the structure for sending and receiving electromagnetic waves; *PowerAmplifier* in charge of amplifying the signal for any transmission; *Multiplexer* modeling ways of separating sending and receiving signals; and *Transceiver*, which does transmission and reception of signals. Nowadays, we can find several technologies of base stations, such as *LTE*, *UMTS*, *WiMAX*, *WiFi*, etc. Considering the size of the *Cells*, named *CellSize* in Fig. 3, base stations can be *MacroCell*, *MicroCell*, *PicoCell*, and *FemtoCell*. Moreover, base stations are located at specific positions that are modeled by the *Location* class. Finally, the energy consumed by base stations is considered by the *PowerConsumption* class.

The another topic in our mobile network ontology is the device, where the *Device* class is the main element. This is the other point in which our solution is focused on for saving energy through policies. All devices are connected to base stations with the aim of consuming the *Services* provided by the network. Furthermore, as the base stations, devices are located at specific position and consume energy. These two aspects are modeled by *Location* and *PowerConsumption*, respectively, which have been commented earlier. From the energy saving perspective, we have modeled *Battery* as an example.

All entities defined earlier for the mobile network ontology are related each other by properties, where a portion of them is used to establish new relationships through policies. For example, the *BaseStation* class use the *hasTechnology*, *hasCell*, and *hasComponent* properties for defining how a given base station is implemented internally. With respect to the another category of our mobile network ontology, the *Device* class is linked to *Service* and *Battery* via the *usesService* and *hasBattery* properties, respectively, in order to know the services that are being used by each device as well as its corresponding battery for energy issues. Note that all these elements are connected to *PowerConsumption*, through the *hasConsumption* property, so as to know the energy consumed by each base station and by each device.

5.2 Deployment of our energy-aware and policy-based system

This section shows the architecture of our energy-aware solution in charge of managing mobile networks oriented to the SDN paradigm at run-time. Figure 4 shows this architecture, where the *SDN plane* contains the layers of the SDN paradigm and the *SDN management plane* depicts the components shaping our solution.

From bottom to top, the first component of our proposal is the *Collector*. This is in charge of joining the infrastructure location, gathered from an independent *Location Middleware*, with the network configuration and statistics received from the *Controller*. It is important to know that the network configuration contains energy-aware information, such as power consumption or cell sizes. All this information is obtained by our OpenDaylight controller [27], using southbound protocols like OpenFlow or SNMP.

When the *Interpreter* component receives the joined information from the *Collector*, the former stores all this

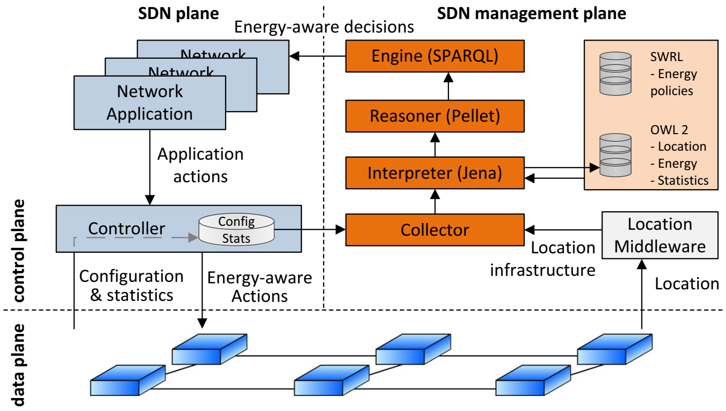


Fig. 4 Architecture of the proposed mobility-aware and policy-based system for energy saving

information by using the ontology detailed earlier. Once it is stored, the *Interpreter* uses the Jena API to generate ontological models with our ontology and the energy-aware policies defined in SWRL [28]. On the other hand, the *Reasoner* component is making use of Pellet [29], which receives the ontological models being generated by the *Interpreter* component and returns the inferred models with new knowledge.

Finally, the *Engine* is in charge of translating the decision(s) performed into SPARQL queries [30], which are applied to the inferred model. Therefore, the energy-aware results of the SPARQL queries are sent to the SDN applications, which will enforce the corresponding energy-aware actions over the network.

6 Experiment results

In this section, we conducted some experiments with the aim of demonstrating the usefulness of our solution. We modeled the energy consumption that is needed for offering a service to a given area A. This requires to consider the consumption of the base stations as well as the consumption derived from the active devices. To this end, we have used Eq. 1, proposed in [12], in order to obtain the energy consumption of the network.

$$P_{total} = c_1 \cdot N_{bs} + c_2 \cdot N_d \cdot d^e \qquad (1)$$

where N_{bs} is the number of base stations; N_d represents the number of active devices; d is the cell size of the base station; c_1 and c_2 are related to the power consumption of the base stations, which depend on the technology used by those base stations; and e is the associated path-loss exponent. Moreover, it is important to take into account that if we want to cover an area A, the cell size of the base station (d) depends on the number of base stations: $N_{bs} = \frac{A}{d^2}$.

In order to calculate the power consumption of the network, we consider that the distance between the active devices and the base stations is d. In other words, we consider the worst case for the devices' location into the cells.

Considering the previous analysis, we are going to perform some experiments in order to see how the parameters of Eq. 1 affect to the network consumption. Specifically, Fig. 5 shows the variation of the power consumption for different fixed consumptions (c_1) when we increase the number of BSs (N_{bs}).

We can see with this experiment that, when the fixed consumption (c_1) decreases, it is better to have many base stations with small cells than a few of them with big cells. In Fig. 5, we can observe this situation when the fixed consumption is 2000W, whose optimum number of base

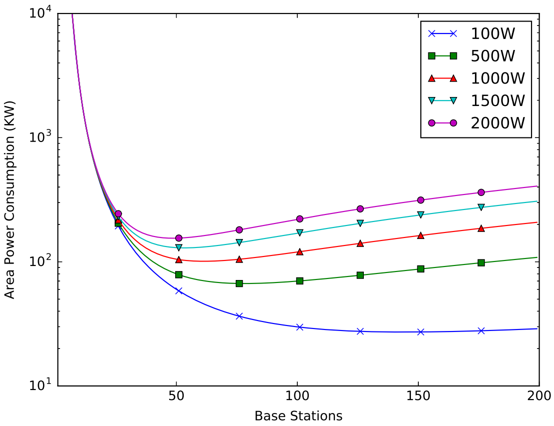


Fig. 5 Area power consumption modifying the number of base stations that cover a given area of 100 sq Km. Different curves show the fixed BS power consumption (c_1) when $c_2 = 0.5mW$ and $N_d = 10,000$

stations is close to 50, while this optimum number is close to 150 when the fixed consumption decreases to 100W.

Our next experiment is depicted in Fig. 6, where it is shown the variation of power consumption regarding the number of active devices and base stations.

Figure 6 demonstrates that when the number of active devices increases, several small cells are more efficient from an energy saving perspective than having a few of them with big cells. For example, if our network has 500 active devices, the optimum number of base stations is close to 20. However, in case the number of active devices increases to 50,000, the optimum number of base stations is between 100 and 150.

Based on the consumption energy model, and as it has been shown in the previous experiments, the number of active devices and the consumption of the base stations determine an optimum number of these base stations offering their services in each time. Our solution with a number of well-defined policies can switch on/off the base stations automatically in an optimum number of them. In Fig. 7, we show the specific points or moments (black circles) when our solution changes the network configuration to save energy.

In this figure, we can see how our solution changes the network configuration by using the policies defined in Section 4 to reduce energy consumption. PI_{urban} establishes the moment when the policies will change the network configuration. In this sense, Fig. 7 shows that when the PI_{urban} value is bigger than 2.5KW (64 active devices) and smaller than 5KW (758 active devices) our policies change the network configuration to 2 base stations. If the power consumption is increased between 5KW (758 active devices) and 11KW (3,693 active devices), it is better to have four base stations. Finally, if the energy consumption is higher than 11KW (we need over 3,693 active devices) the most suitable configuration consists on having 8 base stations.

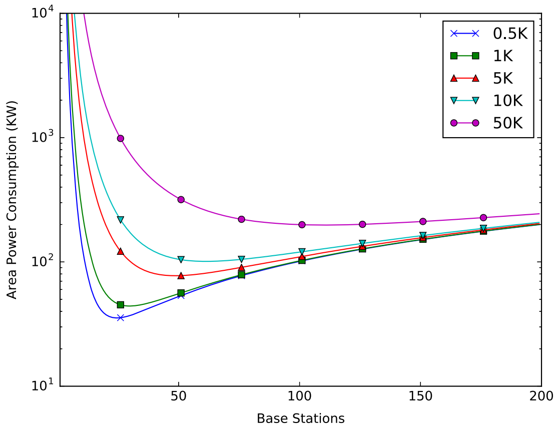


Fig. 6 Area power consumption modifying the number of base stations that cover a given area of 100 sq Km. Different curves show the number of active devices (N_d) when $c_1 = 1000W$ and $c_2 = 0.5mW$

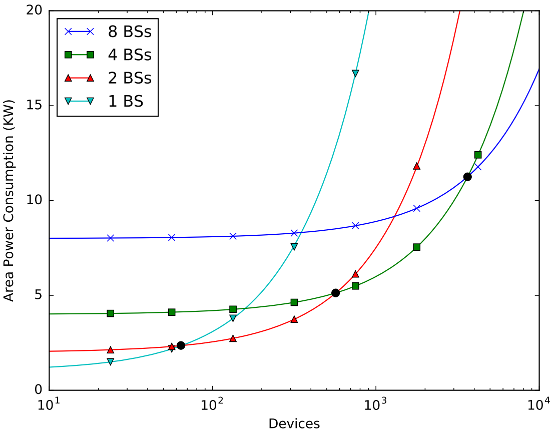


Fig. 7 Area power consumption modifying the number of active devices in a given area of 4 sq Km. Different curves represent the number of base stations when $c_1 = 1000W$ and $c_2 = 0.5mW$

In this section, we have demonstrated the usefulness of our solution, which is capable of changing dynamically and at run time the network configuration in order to consume energy in an optimal way. To this end, we have considered the fixed power consumption, the number of active devices, and the number of base stations and their cell sizes.

6.1 Decisions time

The Reasoner and Engine components are important parts of our solution. They are in charge of maintaining updated the information of the network infrastructure and deciding when our solution has to change the network configuration. In order to check the time required to make decisions at real time for reducing the energy consumption of the network, we conducted several experiments altering the complexity of our ontology.

We have increased the number of statements hold in the knowledge base, which depends on the number of individuals present in the ontology and the number of policies. Increasing the number of individuals and semantic rules, we will provoke an increment on the number of statements, and thus on the complexity of the executions. The conducted tests were carried out in a dedicated PC with an Intel Core i5-3317-U 3.40 GHz, 6 GB of RAM, and a Windows 10 as operative system. The results shown in this section have been obtained by executing the experiments 100 times and computing their arithmetic mean.

Table 1 Individual distribution of population

Element	Amount	Percentage
Devices	9,750	65 %
Location	4,500	30 %
Services	300	2 %
Others	300	2 %
Base Stations	150	1 %
Total	15,000	100 %

The number of individuals contained in our ontology is referred as *population*. This was randomly generated for the experiments, but in a controlled way in order to achieve the desired distribution for simulating a scenario as real as possible. Table 1 depicts the number of elements used in our environment and the percentages obtained for them.

In order to establish the population, we defined an initial population of 3,000 individuals, which is increased with other 3,000 individuals in each step. In Table 2 it is depicted the relationships between the individuals and the statements generated by the Reasoner, in order to show the complexity of our ontology. Table 2 shows that the number of statements is proportionally increased according to the number of individuals.

Figure 8 depicts the time in milliseconds (ms) used by the Reasoner to maintain updated the ontology with the network information and make decisions to change its configuration, considering the policies (*Reasoning time*) and the time needed to notify the decisions to the network applications in charge of switching on/off the base stations (*Query time*).

The last experiment shows that, having 9,750 devices and 150 base stations, our solution needs 570 ms to update the ontology with the network information and take the decision of changing the network configuration (or not). We consider that 570 ms is an acceptable time considering the number of active devices and base stations managed by our solution.

In this section, we have demonstrated that our solution can support a very large number of individuals or statements; when this number is linearly increased, the reasoning time also increases linearly. Furthermore, it is important to notice that the reasoning process could be made in an offline-mode depending on the scenario' necessities. Regarding the query time, we have demonstrated that queries do not have a significant impact in the performance of our solution.

Table 2 Individuals and statements per population

Population	1	2	3	4	5
Individuals	3,000	6,000	9,000	12,000	15,000
Statements	11,770	23,062	34,330	45,622	56,890

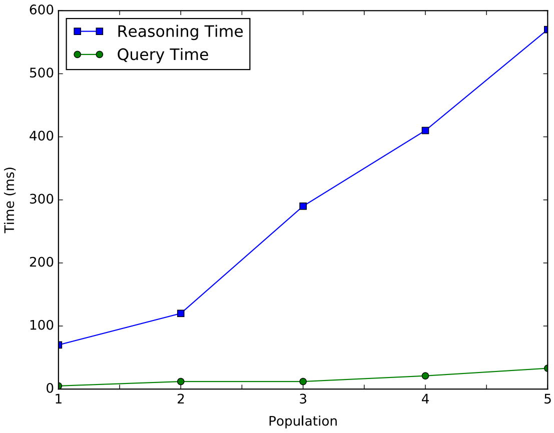


Fig. 8 Time needed by our solution to make the decision of modifying the network configuration

7 Conclusions and future work

A mobility-aware policy-based system oriented to the SDN paradigm has presented in this paper, with the aim of managing the energy efficiency of any network infrastructure at run-time by using policies. These policies allow the SDN paradigm to switch on/off network devices when they consume energy in an inefficient fashion, and also virtualize network resources such as proxies. This latter allows reducing traffic of specific services consumed by users who move close to the network infrastructure.

As future work, we plan to consider the location of devices and the users' behavior patterns in order to make decisions of switching or visualizing the network infrastructure. Having information about users' behavior allows our solution to predict users' movements and manage network resources in an energy saving fashion. In addition, we also plan the protection of the sensitive information. To this end, we could extend our solution to provide privacy-preserving policies, where users could indicate the granularity at which they want to release their location and behavior patterns.

Acknowledgments This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2014, the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (*Framework for Self-Organized Network Management in Virtualized and Software Defined Networks*), the Spanish MINECO (project DHARMA, *Dynamic Heterogeneous Threats Risk Management and Assessment*, with code TIN2014-59023-C2-1-R; and project

TecMASAS, *Techniques to Improve the Architecture of Servers, Applications and Services*, with code TIN2015-66972-C5-3-R), and the European Commission (FEDER/ERDF).

References

1. Horvath R, Nedbal D, Stieninger M (2015) A literature review on challenges and effects of software defined networking. Procedia Comput Sci 64:552–561
2. Huertas Celdrán A, Gil Pérez M, García Clemente FJ, Martínez Pérez G. Enabling highly dynamic mobile scenarios with software defined networking. IEEE Communications Magazine, Feature Topics Issue on SDN Use Cases for Service Provider Networks, In Press
3. Jimenez JM, Romero O, Rego A, Dilendra A, Lloret J (2015) Study of multimedia delivery over software defined networks. Netw Protoc Algorithm 7(4):37–62
4. Molina E, Jacob E, Astarloa A (2016) Using OpenFlow to control redundant paths in wireless networks. Netw Protoc Algorithm 8(1):90–103
5. Jingjin W, Yujing Z, Zukerman M, Yung EKN (2015) Energy-efficient base-stations sleep-mode techniques in green cellular networks A survey. IEEE Commun Surv Tutor 17(2):803–826
6. Auer G, Giannini V, Desset C, Godor I, Skillermark P, Olsson M, Imran MA, Sabella D, Gonzalez MJ, Blume O, Fehske A (2011) How much energy is needed to run a wireless network? IEEE Wirel Commun 18(5):40–49
7. Yun W, Staudinger J, Miller M (2012) High efficiency linear GaAs MMIC amplifier for wireless base station and Femto cell applications. In: IEEE Topical Conference on Power Amplifiers for Wireless and Radio Applications, pp 49–52
8. Marsan MA, Chiaraviglio L, Ciullo D, Meo M (2009) Optimal energy savings in cellular access networks. In: IEEE International Conference on Communications Workshops, pp 1–5
9. Claussen H, Ashraf I, Ho LTW (2010) Dynamic idle mode procedures for femtocells. Bell Labs Tech J 15(2):95–116
10. Rongpeng L, Zhifeng Z, Xianfu C, Palicot J, Honggang Z (2014) TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks. IEEE Trans Wirel Commun 13(4):2000–2011
11. Zhisheng N, Yiqun W, Jie G, Zexi Y (2010) Cell zooming for cost-efficient green cellular networks. IEEE Commun Mag 48(11):74–79
12. Bhaumik S, Narlikar G, Chattopadhyay S, Kanugovi S (2010) Breathe to stay cool: Adjusting cell sizes to reduce energy consumption. In: First ACM SIGCOMM Workshop on Green Networking, pp 41–46
13. Richter F, Fehske AJ, Fettweis GP (2009) Energy efficiency aspects of base station deployment strategies for cellular networks. In: IEEE Vehicular Technology Conference Fall, pp 1–5
14. Yulong Z, Jia Z, Rui Z (2013) Exploiting network cooperation in green wireless communication. IEEE Trans Commun 61(3):999–1010
15. Ming L, Pan L, Xiaoxia H, Yuguang F, Glisic S (2015) Energy consumption optimization for multihop cognitive cellular networks. IEEE Trans Mob Comput 14(2):358–372
16. Andrade S, Ruiz E, Granell E, Lloret J (2013) Energy consumption of wireless network access points. In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 113, pp 81–91
17. Chang L, Jun Z, Letaief KB (2013) Energy efficiency analysis of small cell networks. In: International Conference on Communications, pp 4404–4408
18. Chia Y-K, Sun S, Zhang R (2013) Energy cooperation in cellular networks with renewable powered base stations. In: IEEE Wireless Communications and Networking Conference, pp 2542–2547
19. Kejiang Y, Dawei H, Xiaohong J, Huajun C, Shuang W (2010) Virtual machine based energy-efficient data center architecture for cloud computing: A performance perspective. In: Conference on Cyber, Physical and Social Computing, pp 171–178
20. Hyojoon K, Feamster N (2013) Improving network management with software defined networking. IEEE Commun Mag 51(2):114–119
21. Wang X, Vasilakos AV, Chen M, Liu Y, Kwon TT (2012) A survey of green mobile networks Opportunities and challenges. Mob Netw Appl 17(1):4–20
22. Huaping S, Kumar M, Das SK, Wang Z (2004) Energy-efficient caching and prefetching with data consistency in mobile distributed systems. In: International Parallel and Distributed Processing Symposium 2004, p 67
23. Aligrudic A, Pejanovic-Djurisic M (2014) Energy efficiency metrics for heterogenous wireless cellular networks. In: 2014 Wireless Telecommunications Symposium, pp 1–4
24. Motik B, Patel-Schneider PF, Parsia B (2012) OWL 2 web ontology language: Structural specification and functional-style syntax, 2nd edn. W3C Recommendation
25. Stanford Center for Biomedical Informatics Research. The Protégé tool: A free, open source ontology editor and knowledge-base framework. Available at http://protege.stanford.edu
26. Distributed Management Task Force, Inc. The CIM standard: Common Information Model. Available at http://www.dmtf.org/standards/cim
27. Linux Foundation. OpenDaylight: Open source SDN platform. Available at http://www.opendaylight.org
28. Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) SWRL: A semantic web rule language combining OWL and RuleML, W3C Member Submission
29. Sirin E, Parsia B, Cuenca Grau B, Kalyanpur A, Katz Y (2007) Pellet: A practical OWL-DL reasoner. Web Semant Sci Serv Agents World Wide Web 5(2):51–53
30. Prud'hommeaux E, Seaborne A (eds) (2008) SPARQL query language for RDF, W3C Recommendation

Chapter 7

Enabling highly dynamic mobile scenarios with Software Defined Networking

Title:	Enabling highly dynamic mobile scenarios with Software Defined Networking
Authors:	Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, Gregorio Martínez Pérez
Journal:	IEEE Communications Magazine
Impact factor:	5.125 Q1 (2016)
Publisher:	IEEE
State:	Accepted

Enabling highly dynamic mobile scenarios with Software Defined Networking

Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, and
Gregorio Martínez Pérez, *Member, IEEE*

***Abstract*—Mobile devices have promoted the users' mobility and, therefore, the necessity of providing services that accomplish the users' requirements at any place and time. With this, location becomes a key aspect to provide the dynamism required by solutions like the provisioning of reasonable mobile services by service provider networks. In that sense, the Software Defined Networking (SDN) paradigm arose to evolve from current static networks, where they are manually configured by administrators, towards dynamic networks able to manage by their own at run-time and on demand. Solutions managing the SDN resources by using policies have been proposed, but they do not consider one of the main aspects to network dynamism, i.e. the mobility. This article presents a mobility-aware and policy-based on demand control network solution oriented to the SDN paradigm. This is in charge of managing at run-time the service and/or system state with high-level policies, which consider the mobility of users and services, the network statistics, and the infrastructure location. In this context, we define different use cases with the concerns that end-users find when they are in very crowded places, and the solutions provided by our solution through policies: balancing the network traffic between the infrastructure located close to the overloaded one; creating or dismantling geolocated virtual network infrastructure when the existing one is not enough, or is misused to accomplish the end-user demand; and restricting specific network traffic in critical scenarios, like in sport events where crowd consume services with a large bandwidth.**

***Index Terms*—Software Defined Networking, dynamic scenarios, mobility, management-oriented policies.**

I. Introduction

The recent technology advancements in mobile devices and networks have encouraged users' mobility, thus being location one of the most important aspects for knowing where devices, resources, or people are. Location information can provide useful evidence with which to develop new proposals and solutions. For example, the European Commission is making great efforts, funding the Horizon 2020 Programme to define new use cases where mobility and dynamism are key aspects. Under the 5G-PPP initiative, the EU project METIS-II (*Mobile and wireless communications Enablers for the Twenty-twenty Information Society*) [1] is proposing several use cases that highlight the provisioning of reasonable mobile broadband by service provider networks, with high levels of service experience in crowded areas (e.g., stadiums or shopping malls) and even with end-users on the move (e.g., in cars or trains). Other initiatives are being conducted in parallel in other countries or continents, such as 4G Americas [2], where leading telecommunications service providers and manufacturers are fostering the advancement of the LTE mobile broadband technology and its evolution beyond to 5G, or the IMT-2020 (5G) Promotion Group [3] including main operators, vendors, research institutes in China.

Managing the dynamism displayed by the previous proposals requires a deep change from the current networks, where service provider administrators usually configure the network depending on triggered events, towards Self-Organized Networks (SON) [4], which are able to monitor, manage, and configure by their own at run-time and depending on different factors, among which location of users receiving a service is a critical one. This diversity requires that service provider networks collect and analyze large quantities of data, thereby increasing the network management complexity.

In order to ease the network management arose the Software Defined Networking (SDN) paradigm [5]. SDN is a paradigm where a central software program, called *controller*, is the brain of the network to manage its behavior, thereby making network devices become simple packet forwarding elements. This paradigm focuses on the separation of the *control plane* (where the controller is) from the *data plane* (where the forwarding devices are); the definition of a logically centralized controller; the use of open interfaces between the control and data planes; and the programmability of the network by applications. These features provide several benefits, such as the ease to change the network configuration through software rather than typing commands in network devices. Nowadays, we can find several solutions focused on deciding how the SDN resources have to be managed at run-time.

For example, NetGraph [6] provides a scalable graph library and its interfaces with the controller to support network management functions, such as run-time monitoring and diagnostics. Another example is Procera [7], an event-driven network control framework that uses high-level policies to manage and configure the network state. This solution enables dynamic policies, which are translated into a set of forwarding rules to manage the network state by the controller. Following

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2014, the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (*Framework for Self-Organized Network Management in Virtualized and Software Defined Networks*), the Spanish MICINN (project DHARMA, *Dynamic Heterogeneous Threats Risk Management and Assessment*, with code TIN2014-59023-C2-1-R), and the European Commission (FEDER/ERDF).

A. Huertas Celdrán, M. Gil Pérez, and G. Martínez Pérez are with the Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071 Murcia, Spain (e-mail: alberto.huertas@um.es; mgilperez@um.es; gregorio@um.es)

F. J. García Clemente is with the Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, 30071 Murcia, Spain (e-mail: fgarcia@um.es)

the policy-oriented approach, we find OpenSec [8]. Opensec is an OpenFlow-based framework that allows the network operators to describe security policies using human-readable language to implement them across the network.

Up until this point, we have seen that there are solutions allowing the SDN controller to manage the network resources at run-time, using policies defined by service provider network administrators beforehand. Yet, these solutions do not consider one of the main aspects that provide network dynamism, i.e. the mobility. We think that it is a must to consider users' mobility and the location of the network resources so as to manage and configure the SDN state in a more accurate way. In that sense, this paper presents a mobility-aware and policy-based on demand control network solution oriented to the SDN paradigm. Specifically, our solution is in charge of managing the SDN resources at run-time, using high-level policies that consider the mobility of users and services, the network statistics, and the infrastructure location. These policies are oriented to guarantee end-users experience in very crowded places (e.g., stadiums, shopping malls, or unexpected traffic jams). To this end, the policies decide when the SDN should balance the network traffic between the infrastructure located close to the congested one; when the SDN should create or dismantle physical or virtual infrastructure in case of the congested one is not enough to accomplish the end-user demand; and when the SDN should restrict or limit specific services or network traffic in critical situations produced by large crowds using services at specific areas.

II. Use cases in a dynamic mobile scenario

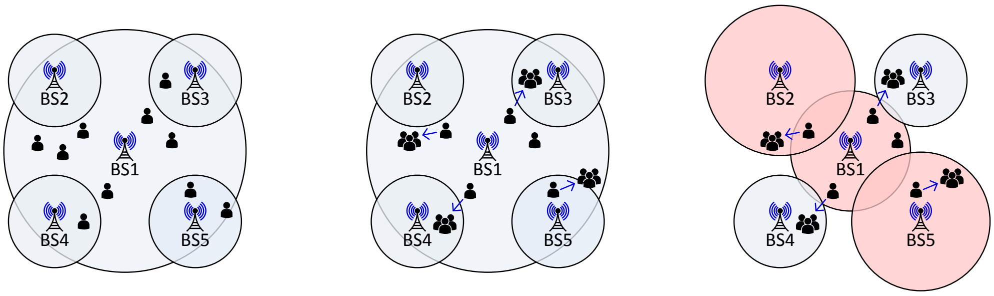
This section shows a dynamic mobile scenario composed of four different use cases, with which to illustrate the service provisioning concerns that end-users can find when they are in a very crowded place (e.g., open air festivals, traffic jams, stadiums, or public events with lots of people). The first use case shows a concern when the network provides low quality services, even having enough resources to accomplish the end-users requirements. The second use case considers that the network does not have enough resources and provides low quality services, whereas in the third use case the network does not have enough resources and it is not able to provide services. In the fourth use case, the network misuses its resources to provide services. In Section III-B, we will explain in detail how our solution manages these concerns to ensure end-users experience.

A use case showing the first concern is shown in Fig. 1a, where a central base station (BS1) and four secondaries (BS2, BS3, BS4, and BS5) are located along a specific area. When large crowds are formed, and end-users move across the networking area, the BS1 is overloaded. Fig. 1b shows this situation. BS1 is congested because it is providing services to a lot of users, and BS2 and BS5 just to a few. To solve it, our solution allows the load balancing at run-time between the base stations located close to the congested one (BS1). In that sense, Fig. 1c shows how the zoom cell size load balancing technique [9] decreases the BS1 cell size and increases BS2 and BS5 cell sizes to ensure end-users experience. It is worth noting that when the crowd moves inside or outside the area, our system dynamically balances the load traffic increasing or decreasing the size of the base stations cells. An example of this situation could be an open festival with a central base station covering the whole festival, and four base stations close to the concert stages. Once the concerts start, the crowd moves to the concert stages and overload the central base station (e.g., sharing photos and videos through social networks).

Regarding the second concern, produced when the network does not have enough resources and provides low quality services, Fig. 2a shows a use case where a base station (BS1) and four generic hardware elements (HW) with 3G/4G antennas are located along a specific area. In this context, Fig. 2b shows the moment when a mobile crowd is formed and the BS1 cannot accomplish the end-users requirements. To manage this situation, our proposal allows creating virtual base stations (BS2, BS3, BS4, and BS5) at run-time by using at will the generic hardware elements. Fig. 2c, depicts the situation managed by our solution. The created virtual base stations are providing services once the network traffic is balanced. It is worthy to note that once the crowd is gone our proposal dismantles the virtual base stations, and the generic hardware will be available to the service provider network. This situation is shown in Fig. 4, which is explained in detail at the end of this section. An example of this second use case, could be a motorway with a base station and four generic hardware elements located along its area. Due to weather conditions, a traffic jam is formed and the base station cannot accomplish the requirements of the crowd, even knowing the atmospheric forecast. To solve it, our solution decides to create four virtual base stations from the existing generic hardware and balances the traffic between them.

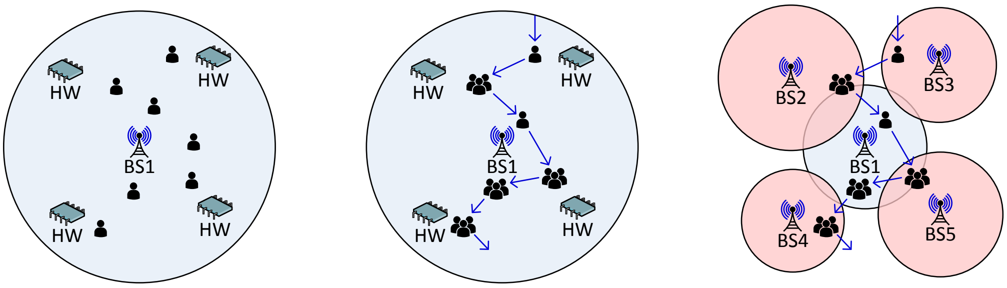
The use cases commented earlier may become critical situations when the network does not have more available resources to accomplish the crowd necessities. In this sense, Fig. 3a shows a new use case where the whole available network infrastructure (all the base stations) is already deployed in a certain area to ensure the end-users experience. Fig. 3b depicts how this situation could become critical causing the network cannot provide services when more users come and consume services that require a large bandwidth like, for example, 4K Ultra High Definition (UHD) video. To solve it, Fig. 3c shows the scenario, where our solution decides that all the base stations reduces the quality of video service from 4K Ultra High Definition (UHD) to High Definition (HD), and limits the bit rate to decrease the network congestion. As in the previous use cases, the reverse process (restrictions are removed) is performed when crowd conditions disappear. An example of this use case could be the Super Bowl event, where the whole network infrastructure is deployed and balanced along the stadium. At the celebration, the crowd massively makes use of the network to send 4K-UHD videos, thus causing the base stations cannot accomplish the demand.

Up until now, we have seen several concerns generated when large crowds are formed. However, it is important to consider the reverse process, when the crowds are gone and the resources are not used in an efficient way, wasting energy resources. In that sense, the fourth concern arisen when



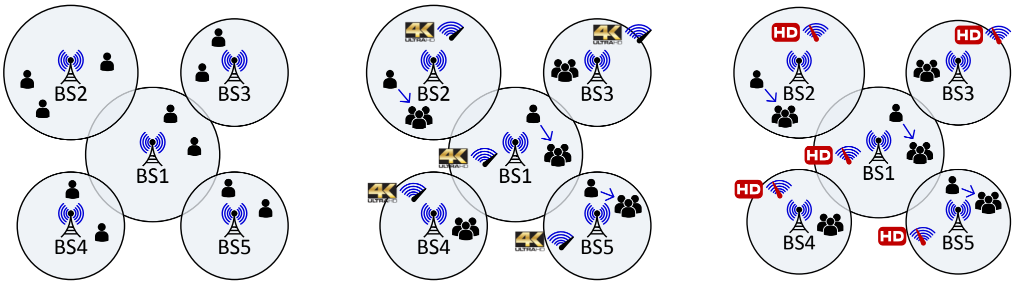
(a) Initial scenario (b) Concerned scenario (c) Managed scenario

Fig. 1. Network with enough resources providing low quality services in a crowded scenario.



(a) Initial scenario (b) Concerned scenario (c) Managed scenario

Fig. 2. Network without enough resources providing low quality services in a crowded scenario.



(a) Initial scenario (b) Concerned scenario (c) Managed scenario

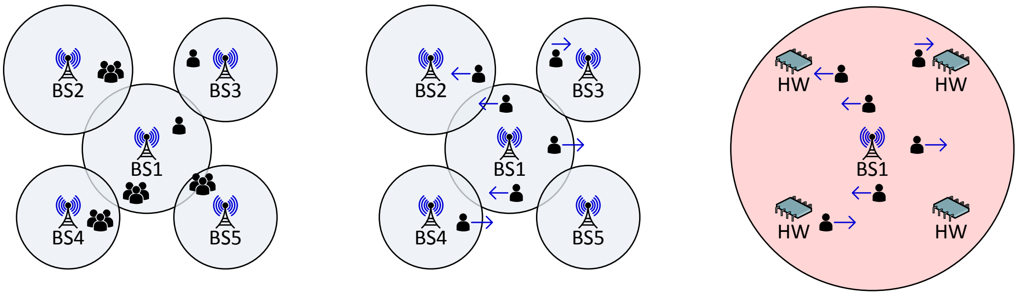
Fig. 3. Network without more resources unable to provide services in a crowded scenario.

the network uses unnecessary resources to provide services. Fig. 4a shows a use case where a physical base station (BS1) and four virtual base stations (from BS2 to BS5) are providing services along a specific crowded area. Fig. 4b shows the moment when the crowd starts leaving the area and all base stations continue providing services to a few users. In order to prevent the misuse of resources, our proposal allows dismantling the virtual base stations (BS2, BS3, BS4, and BS5) at run-time. Fig. 4c depicts this situation. The virtual base stations are dismantled and BS1 provides services after increasing its cell size through a load balancing.

Following with the traffic jam example, the jam begins clearing up when the weather conditions improve, and the virtual network infrastructure previously created is not necessary. In that sense, our solution decides to dismantle the four virtual base stations and balance their traffic to BS1 by increasing its cell size to cover the whole motorway area.

III. SDN MANAGEMENT POLICIES

The policy-based management lets the simplification and automation of the network administration processes [10]. By using policies, the SDN paradigm can control the network state at run-time and on demand in order to guarantee the end-user experience. Among the different sets of policies, we emphasize here the use of mobility-aware management-oriented policies, defined by the service provider network



(a) Initial scenario (b) Concerned scenario (c) Managed scenario

Fig. 4. Network misusing resources to provide services.

TABLE I
ELEMENTS THAT COMPOSE THE BASE OF OUR MOBILITY-AWARE MANAGEMENT POLICIES

Element	Values	Description
Type	Load balancing, Infrastructure, Restriction	Indicate the kind of policy
Resource	Base station, Switch, Service, Intrusion Detection System, etc.	Network element whose information is being managed
Metric	Average of Bytes per flow (ABf), Average Number of Packets per Flow (ANPPF), Average of Duration per flow (ADf), etc.	Define the term that encompasses the different parameters that can be used to evaluate the network state
Location	Geographic position, Area, etc.	Position or region where the policy will be enforced
Date	Date, Hour, Timestamp, etc.	Moment or period of time at which the policy will be applied
Result	Balance, Create, Dismantle, Disable, Limit	Action performed over the network when the policy is applied

administrator to decide the actions made by the SDN according the network infrastructure statistics and location, and the mobility of users and services. In our solution, the schema of the rules shaping the policies are composed of the elements shown in TABLE I, being

$$Type \wedge Resource \wedge Metric \wedge Location \wedge Date \rightarrow Result$$

A. Policies to guarantee end-users experience

We introduce below the three kinds of policies required to manage the concerns depicted in the previous use cases, although other sorts of policies could be defined at will because the proposed solution herein presented is extensible.

1) Load balancing policies: These policies are in charge of deciding when, where, and why it is needed a load balancing of the traffic between the network resources, this being a key aspect in the SDN paradigm for managing and forwarding at run-time the packets passing by the network, considering their location, the date, and the metrics previously defined. These parameters are optional in this kind of policies. It is important to note that we are not proposing a new load balancing solution, but ours is able to use any load balancing solution.

2) Infrastructure policies: These policies allow the SDN paradigm to create or dismantle virtual network resources located at specific locations. As the previous kind of policies, they can be applied in a proactive way in case of knowing when the network needs more infrastructure. As before, the *Date*, *Metric*, and *Location* parameters are also optional.

3) Restriction policies: They manage the network or SDN to guarantee the end-user experience. These policies allow the SDN paradigm to disable or limit the traffic of given network resources or services in case the traffic overload is critic.

B. Managing the dynamic mobile scenario

It is shown below how our solution manages the concerns presented in Section II and how we guarantee end-user experience in very crowded places, when important changes in the population are produced in a short period of time.

Regarding the first concern, when the network has enough resources but it provides low quality services, our solution defines a generic load balancing policy. The policy defined below indicates, for example, that when the ABf value of any base station is within *Yellow* range values (the range of this alarm is set by the service provider administrator depending on the state and characteristics of the scenario), the network should try to balance the traffic load between the base stations located at the same area as the congested ones.

```
Type(#LoadBalancing) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
hasABf(?bs,?abf) ∧ inRange(?abf,#Yellow)
→ balance(?bs,?nearBs)
```

In this policy, *BaseStation* is a possible value of the *Resource* element (defined in our policy schema as shown in TABLE I); *Location* and *locatedBaseStation* are modeled by the *Location* element; *hasABf* is a specific *Metric*; and *balance* is a possible value of the *Result* element. Considering our open air festival scenario, Fig. 1c shows in red the changes made by this policy in the festival area.

To manage the second concern, when the network does not have enough resources and provides low quality services, our solution defines an *Infrastructure* policy. As an example, the policy defined below creates new virtual base stations from generic hardware located close to the congested one when ANPPF of any base station is within *Orange* range values (this alarm is also defined by the service provider administrator, whose range of values is higher than *Yellow* range).

```
Type(#Infrastructure) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedResources(?area,?resource) ∧
hasANPPF(?bs,?anppf) ∧ inRange(?anppf,#Orange)
→ create(?resource,#BaseStation)
```

In this policy, *BaseStation* is a value of the *Resource* element; *Location* and *locatedResources* are shaped by the *Location* element; *hasANPPF* is a kind of *Metric*; and *create* makes reference to a possible value of the *Result* element. Following the traffic jam scenario, Fig. 2c depicts in red the virtual base stations (BS2, BS3, BS4, and BS5) created from the existing generic hardware. Furthermore, it is necessary a new load balancing policy once the virtual base stations are created, in order to balance the network traffic between them.

Regarding the third concern, when the network does not have more resources and it cannot provide services, our solution avoids this situation with two *Restriction* policies. The first one is in charge of disabling the 4K-UHD video traffic of the base stations located at the congested area. It is important to note that a disable action does not filter the video service, but disables a specific quality and the service is provided with lower quality. Below we can find this policy.

```
Type(#Restriction) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
Service(?nearBs,?service) ∧
hasABf(?bs,?abf) ∧ inRange(?abf,#Red)
→ disable(?service,#4K-UHDVideo)
```

The second *Restriction* policy limits the bit rate of the services provided by the base stations located in the congested area.

```
Type(#Restriction) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
Service(?nearBs,?service) ∧
hasABf(?bs,?abf) ∧ inRange(?abf,#Red)
→ limit(?service,#BitRate)
```

In both policies, *BaseStation* and *Service* are values of the *Resource* element; *Location* and *locatedBaseStation* are modeled by the *Location* element; *hasABf* is a kind of *Metric*; and *disable* and *limit* are values of the *Result* element. Fig. 3c depicts the Super Bowl event, where all base stations located at the stadium area decrease the video quality (from 4K-UHD to HD) and limit the bit rate.

Finally, the fourth concern arises when crowd is gone and the network resources are misused. Our solution defines an *Infrastructure* policy that dismantles the misused virtual base stations located close to the underloaded one when the ANPPF value of any base station is less than *Yellow* range values.

```
Type(#Infrastructure) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
hasANPPF(?bs,?anppf) ∧ lessRange(?anppf,#Yellow) ∧
hasANPPF(?nearBs,?nearAnppf) ∧
lessRange(?nearAnppf,#Yellow)
→ dismantle(?nearBs,#BaseStation)
```

As before, *BaseStation* is a value of the *Resource* element; *Location* and *locatedBaseStation* are shaped by the *Location* element; *hasANPPF* is a kind of *Metric*; and *dismantle* corresponds to a value of the *Result* element. Following the traffic jam scenario, Fig. 4c shows the virtual base stations (BS2, BS3, BS4, and BS5) dismantled and converted again in generic hardware (HW). Furthermore, it is necessary a new load balancing policy once the virtual base stations are dismantled, in order to balance the network traffic to BS1.

IV. Architecture

This section describes our mobility-aware architecture for managing networks oriented to the SDN paradigm at run-time and on demand. Fig. 5 shows the proposed architecture, where the *SDN plane* contains the elements forming the layers of the SDN paradigm and the *SDN management plane* depicts the components composing our solution.

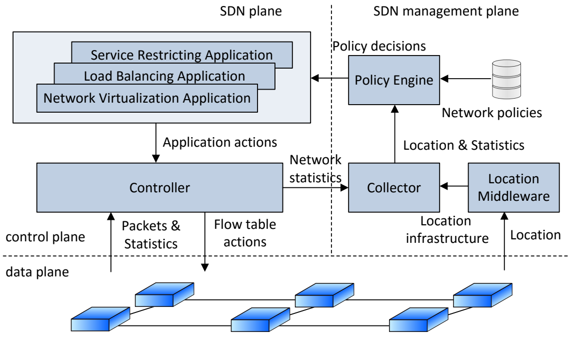


Fig. 5. Architecture of the proposed mobility-aware and policy-based solution

A. SDN plane

One of the main features of SDN is the decoupling of the control from the data plane. In that sense, our proposal has the *data plane* at the bottom layer, where physical and virtual network infrastructure (base stations, switches, routers, etc.) forwards and manipulates packets, not having any control intelligent. The networking logic control is allocated in the *control plane*, in which the Controller component lies on.

To exchange information between control and data planes, our solution makes use of OpenFlow [11]. This is one of the most common southbound SDN interfaces and allows our

Controller to get statistical data about the network traffic, as well as the management of the network infrastructure through software. Nowadays, there are a high number of OpenFlow-capable controllers, such as OpenDaylight, which is used by our solution.

Finally, the application layer is at the top of the SDN stack. This layer contains the applications that use the services provided by the Controller to perform tasks related to the network. Among the existing applications, we highlight three of them used in our solution. The Network Virtualization Application is in charge of managing the virtual network resources by using a well-known open-source software platform called OpenStack Networking (Neutron). Other solutions can be found in the literature such as FlowN [12], which presents an architecture for SDN virtualization. This allows tenants to specify their own address space, topology, and control logic. The second application is the Load Balancing Application, which redistributes the network traffic between the network resources. In this topic, several solutions have been proposed, as the one presented in [9], where load balancing is performed increasing or decreasing the cell size according to the traffic load, user requirements, and network conditions. The last application is the Service Restriction Application, which restricts the network traffic by considering different parameters, such as the bit rate, services, ports, etc.

B. SDN management plane

The main component of our solution is the Policy Engine. This component is in charge of making decisions over the SDN applications, considering network statistics, the infrastructure location information, and the network policies. Among the possible decisions, we highlight three of them. The first one consists on notifying the Load Balancing Application about the need of redirecting the traffic. The second one is focused on deciding if Network Virtualization Application has to create or dismantle virtual resources. The last decision is aimed at knowing if the Service Restriction Application should limit or disable some kind of traffic.

To perform the previous decisions, the Policy Engine uses network policies, defined by the service provider network administrator, and geospatial network statistic information provided by the Collector. This component generates geospatial network statistics, by joining the information received from the Controller and the infrastructure location obtained from the Location Middleware. In order to deploy the Collector, we have several options like, for example, the extended version of IPFIX that includes the location of the network infrastructure to generate network statistics.

Finally, the Location Middleware component obtains the locations of the network infrastructure. This is an independent middleware that provides independence to our solution with regard to the location system used, thus allowing the Location Middleware to choose the best location system or middleware depending on the environment.

V. Conclusion and future work

This paper has presented a mobility-aware solution to manage at run-time networks oriented to the SDN paradigm, considering users' mobility as a key aspect for the service provision. This proposal uses management policies to decide on demand the actions performed by the network, considering the mobility of users and services, the network statistics, and the infrastructure location. These policies ensure the end-user experience in crowded scenarios balancing the network traffic between the infrastructure located close to the congested one, when the SDN has enough resources but it provides low quality services; creating virtual network infrastructure when the SDN does not have enough resources and provides low quality services; and restricting specific network traffic, when the SDN does not have more resources and it is unable to provide services.

As next steps of research, we plan to validate our solution in a 5G advanced self-organizing network, as this has an important intelligence component oriented to the SDN paradigm. This scenario is proposed in the EU project for 5G called Selfnet, which is included in the 5G-PPP initiative and where the authors of this paper are currently working.

References

[1] European Commission, "The EU project METIS-II," [Online]. Available: https://metis-ii.5g-ppp.eu.

[2] 4G Americas, "The voice of 5G for the Americas," [Online]. Available: http://www.4gamericas.org.

[3] "The Chinese IMT-2020 (5G) Promotion Group," [Online]. Available: http://www.imt-2020.cn/en.

[4] H. Yang, X. Meng, and S. Lu, "Self-organized network-layer security in mobile ad hoc networks," in Proceedings of the 1st ACM Workshop on Wireless Security, Aug. 2002, pp. 11–20.

[5] R. Horvath, D. Nedbal, and M. Stieninger, "A literature review on challenges and effects of software defined networking," Procedia Computer Science, vol. 64, pp. 552–561, 2015.

[6] R. Raghavendra, J. Lobo, and K.-W. Lee, "Dynamic graph query primitives for SDN-based cloud network management," in Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks, Aug. 2012, pp. 97–102.

[7] K. Hyojoon and N. Feamster, "Improving network management with software defined networking," IEEE Communications Magazine, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[8] A. Lara and B. Ramamurthy, "OpenSec: Policy-based security using Software-Defined Networking," IEEE Transactions on Network and Service Management, vol. 13, no. 1, pp. 30–42, Mar. 2016.

[9] N. Zhisheng, W. Yiqun, G. Jie, and Y. Zexi, "Cell zooming for cost-efficient green cellular networks," IEEE Communications Magazine, vol. 48, no. 11, pp. 74–79, Nov. 2010.

[10] D. C. Verma, "Simplifying network administration using policy-based management," IEEE Network, vol. 16, no. 2, pp. 20–26, Mar. 2002.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[12] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," IEEE Internet Computing, vol. 17, no. 2, pp. 20–27, Mar. 2013.

Bibliografía

[1] OSI. *Information processing systems-Open system inteconnection-Systems management overview.* ISO 10040, 1991.

[2] Jefatura del Estado. Ley Orgánica de Protección de Datos de Carácter Personal. https://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf.

[3] D. W. Samuel and D. B. Louis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[4] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser. Terminology for policy-based management. IETF Request for Comments 3198, November 2001.

[5] B. Moore. Policy Core Information Model (PCIM) extensions. IETF Request for Comments 3460, January 2003.

[6] S. Godik and T. Moses. OASIS eXtensible Access Control Markup Language (XACML). *OASIS Committee Specification*, 2002.

[7] A. Dardenne, A Van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50, 1993.

[8] F. L. Gandon and N. M. Sadeh. Semantic web technologies to reconcile privacy and context awareness. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):241–260, April 2004.

[9] I. Horrocks. Ontologies and the semantic web. *Communications ACM*, 51(12):58–67, December 2008.

[10] C. Bennewith and R. Wickers. *The mobile paradigm for content development*, pages 101–109. Vieweg+Teubner, 2009.

[11] I. A. Junglas and R. T. Watson. Location-based services. *Communications ACM*, 51(3):65–69, March 2008.

[12] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[13] 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 *Handheld and Ubiquitous Computing*, pages 304–307, September 1999.

[14] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceeding of the 1st Workshop Mobile Computing Systems and Applications*, pages 85–90, December 1994.

[15] N. Ryan, J. Pascoe, and D. Morse. Enhanced reality fieldwork: The context aware archaeological assistant. In *Proceedings of the 25th Anniversary Computer Applications in Archaeology*, pages 85–90, December 1997.

[16] A. K. Dey. Context-aware computing: The CyberDesk project. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 51–54, 1998.

[17] P. Prekop and M. Burnett. Activities, context and ubiquitous computing. *Computer Communications*, 26(11):1168–1176, July 2003.

[18] R. M. Gustavsen. Condor–an application framework for mobility-based context-aware applications. In *Proceedings of the Workshop on Concepts and Models for Ubiquitous Computing*, volume 39, September 2002.

[19] C. Tadj and G. Ngantchaha. Context handling in a pervasive computing system framework. In *Proceedings of the 3rd International Conference on Mobile Technology, Applications and Systems*, October 2006.

[20] S. Dhar and U. Varshney. Challenges and business models for mobile location-based services and advertising. *Communications ACM*, 54(5):121–128, May 2011.

[21] F. Ricci, L. Rokach, and B. Shapira. *Recommender systems: Introduction and Challenges*, pages 1–34. Springer US, 2015.

[22] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. *Collaborative filtering recommender systems*, pages 291–324. Springer Berlin Heidelberg, 2007.

[23] P. Lops, M. de Gemmis, and G. Semeraro. *Content-based recommender systems: State of the art and trends*, pages 73–105. Springer US, 2011.

[24] D. Slamanig and C. Stingl. Privacy aspects of eHealth. In *Proceedings of Conference on Availability, Reliability and Security*, pages 1226–1233, March 2008.

[25] C. Wang. Policy-based network management. In *Proceedings of the International Conference on Communication Technology*, volume 1, pages 101–105, 2000.

[26] A. Huertas Celdrán, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez. SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications. *IEEE Systems Journal*, 10(3):1111–1124, September 2016.

[27] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):28–37, May 2001.

[28] B. Motik, P. F. Patel-Schneider, and B. Parsia (ed.). OWL 2 web ontology language: Structural specification and functional-style syntax (2nd ed.), W3C Recommendation, December 2012.

[29] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML, W3C Member Submission, May 2004.

[30] E. Prud'hommeaux and A. Seaborne (ed.). SPARQL query language for RDF, W3C Recommendation, January 2008.

[31] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez. PRECISE: Privacy-aware recommender based on context information for Cloud service environments. *IEEE Communications Magazine*, 52(8):90–96, August 2014.

[32] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez. What private information are you disclosing? A privacy-preserving system supervised by yourself. In *Proceedings of the 6th International Symposium on Cyberspace Safety and Security*, pages 1221–1228, August 2014.

[33] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez. MASTERY: A multicontext-aware system that preserves the users' privacy. In *IEEE/IFIP Network Operations and Management Symposium*, pages 523–528, April 2016.

[34] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez. Policy-based management for green mobile networks through Software-Defined Networking. *Mobile Networks and Applications*, Published online 05 December 2016.

[35] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez. Enabling highly dynamic mobile scenarios with Software Defined Networking. *IEEE Communications Magazine, Feature Topics Issue on SDN Use Cases for Service Provider Networks*, Accepted, 2017.

[36] R. Boutaba and I. Aib. Policy-based management: A historical perspective. *Journal of Network and Systems Management*, 15(4):447–480, 2007.

[37] P. A. Carter. *Policy-based management*, pages 859–886. Apress, 2015.

[38] D. Florêncio and C. Herley. Where do security policies come from? In *Proceedings of the 6th Symposium on Usable Privacy and Security*, pages 10:1–10:14, 2010.

[39] K. Yang and X. Jia. *DAC-MACS: Effective data access control for multi-authority Cloud storage systems*, pages 59–83. Springer New York, 2014.

[40] B. W. Lampson. Dynamic protection structures. In *Proceedings of the Fall Joint Computer Conference*, pages 27–38, 1969.

[41] B. W. Lampson. Protection. *ACM SIGOPS Operating Systems Review*, 8(1):18–24, January 1974.

[42] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations. Technical report, DTIC Document, 1973.

[43] D. F. Ferraiolo and D. R. Kuhn. Role-based access controls. In *Proceedings of the 15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[44] V. P. Astakhov. *Surface integrity – Definition and importance in functional performance*, pages 1–35. Springer London, 2010.

[45] K. J. Biba. Integrity considerations for secure computer systems. Technical report, DTIC Document, 1977.

[46] M. J. Culnan and P. K. Armstrong. Information privacy concerns, procedural fairness, and impersonal trust: An empirical investigation. *Organization Science*, 10(1):104–115, 1999.

[47] A. I. Antón, E. Bertino, N. Li, and T. Yu. A roadmap for comprehensive online privacy policy management. *Communications ACM*, 50(7):109–116, July 2007.

[48] J. Karat, C. M. Karat, C. Brodie, and J. Feng. Privacy in information technology: Designing to enable privacy policy management in organizations. *International Journal of Human-Computer Studies*, 63(1–2):153–174, 2005.

[49] M. Jafari, R. Safavi-Naini, P. W. L. Fong, and K. Barker. A framework for expressing and enforcing purpose-based privacy policies. *ACM Transaction Information Systesms Security*, 17(1):3:1–3:31, August 2014.

[50] G. Karjoth, M. Schunter, and M. Waidner. *Platform for enterprise privacy practices: Privacy-enabled management of customer data*, pages 69–84. Springer Berlin Heidelberg, 2003.

[51] S. R. Blenner, M. Kollmer, A. J. Rouse, N. Daneshvar, C. Williams, and L. B. Andrews. Privacy policies of Android diabetes apps and sharing of health information. *JAMA*, 315(10):1051–1052, 2016.

[52] R. Ramanath, F. Liu, N. Sadeh, and N. A Smith. Unsupervised alignment of privacy policies using hidden Markov models. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 605–610, June 2014.

[53] J. Gerlach, T. Widjaja, and P. Buxmann. Handle with care: How online social network providers' privacy policies impact users' information sharing behavior. *The Journal of Strategic Information Systems*, 24(1):33–43, 2015.

[54] D. C. Verma. Simplifying network administration using policy-based management. *IEEE Network*, 16(2):20–26, March 2002.

[55] D. C. Verma. *Policy-based networking: Architecture and algorithms.* New Riders Publishing, 2000.

[56] J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, and G. Pavlou. A methodological approach toward the refinement problem in policy-based management systems. *IEEE Communications Magazine*, 44(10):60–68, October 2006.

[57] F. Perich. Policy-based network management for next generation spectrum access control. In *Proceedings of International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 496–506, April 2007.

[58] S. Shin, P. A Porras, V. Yegneswaran, M. W. Fong, G. Gu, and M. Tyson. FRESCO: Modular composable security services for Software-Defined Networks. In *Proceedings of the 20th Annual Network and Distributed System Security Symposium*, 2013.

[59] K. Odagiri, S. Shimizu, N. Ishii, and M. Takizawa. Functional experiment of virtual policy based network management scheme in Cloud environment. In *International Conference on Network-Based Information Systems*, pages 208–214, September 2014.

[60] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking control of the enterprise. In *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 1–12, August 2007.

[61] M. Wichtlhuber, R. Reinecke, and D. Hausheer. An SDN-based CDN/ISP collaboration architecture for managing high-volume flows. *IEEE Transactions on Network and Service Management*, 12(1):48–60, March 2015.

[62] A. Lara and B. Ramamurthy. OpenSec: Policy-based security using Software-Defined Networking. *IEEE Transactions on Network and Service Management*, 13(1):30–42, March 2016.

[63] W. Jingjin, Z. Yujing, M. Zukerman, and E. K. N. Yung. Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey. *IEEE Communications Surveys Tutorials*, 17(2):803–826, 2015.

[64] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske. How much energy is needed to run a wireless network? *IEEE Wireless Communications*, 18(5):40–49, 2011.

[65] W. Yun, J. Staudinger, and M. Miller. High efficiency linear GaAs MMIC amplifier for wireless base station and Femto cell applications. In *IEEE Topical Conference on Power Amplifiers for Wireless and Radio Applications*, pages 49–52, January 2012.

[66] M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo. Optimal energy savings in cellular access networks. In *IEEE International Conference on Communications Workshops*, pages 1–5, June 2009.

[67] H. Claussen, I. Ashraf, and L. T. W. Ho. Dynamic idle mode procedures for femtocells. *Bell Labs Technical Journal*, 15(2):95–116, 2010.

[68] L. Rongpeng, Z. Zhifeng, C. Xianfu, J. Palicot, and Z. Honggang. TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks. *IEEE Transactions on Wireless Communications*, 13(4):2000–2011, 2014.

[69] G. C. Januario, C. H. A. Costa, M. C. Amarai, A. C. Riekstin, T. C. M. B. Carvalho, and C. Meirosu. Evaluation of a policy-based network management system for energy-efficiency. In *IFIP/IEEE International Symposium on Integrated Network Management*, pages 596–602, May 2013.

[70] C. Dsouza, G. J. Ahn, and M. Taguinod. Policy-driven security management for fog computing: Preliminary framework and a case study. In *Conference on Information Reuse and Integration*, pages 16–23, August 2014.

[71] H. Kim and N. Feamster. Improving network management with Software Defined Networking. *IEEE Communications Magazine*, 51(2):114–119, February 2013.

[72] O. Gaddour, A. Koubaa, and M. Abid. Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL. *Ad Hoc Networks*, 33:233–256, 2015.

[73] Q. Zhao, D. Grace, and T. Clarke. Transfer learning and cooperation management: Balancing the quality of service and information exchange overhead in cognitive radio networks. *Transactions on Emerging Telecommunications Technologies*, 26(2):290–301, 2015.

[74] M. Charalambides, P. Flegkas, G. Pavlou, A. K. Bandara, E. C. Lupu, A. Russo, N. Dulav, M. Sloman, and J. Rubio-Loyola. Policy conflict analysis for quality of service management. In *Proceedings of the 6th IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 99–108, June 2005.

[75] M. F. Bari, S. R. Chowdhury, R. Ahmed, and Boutaba R. PolicyCop: An autonomic QoS policy enforcement framework for Software Defined Networks. In *2013 IEEE SDN for Future Networks and Services*, pages 1–7, November 2013.

[76] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[77] K. R. Wood, T. Richardson, F. Bennett, A. Harter, and A. Hopper. Global teleporting with Java: Toward ubiquitous personalized computing. *Computer*, 30(2):53–59, February 1997.

[78] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the Internet of Things: A survey. *IEEE Communnications Surveys Tutorials*, 16(1):414–454, 2014.

[79] B. Guo, L. Sun, and D. Zhang. The architecture design of a cross-domain context management system. In *Proceedings of Conference Pervasive Computing and Communications Workshops*, pages 499–504, April 2010.

[80] A. Badii, M. Crouch, and C. Lallah. A context-awareness framework for intelligent networked embedded systems. In *Proceedings of Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services*, pages 105–110, August 2010.

[81] S. Pietschmann, A. Mitschick, R. Winkler, and K. Meissner. CroCo: Ontology-based, CROss-application COntext management. In *Proceedings of Workshop on Semantic Media Adaptation and Personalization*, pages 88–93, December 2008.

[82] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 270–275, January 2004.

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

[84] D. Ejigu, M. Scuturici, and L. Brunie. CoCA: A COllaborative Context-Aware service platform for pervasive computing. In *Proceedings of Conference Information Technologies*, pages 297–302, April 2007.

[85] R. Yus, E. Mena, S. Ilarri, and A. Illarramendi. SHERLOCK: Semantic management of location-based services in wireless environments. *Pervasive and Mobile Computing*, 15:87–99, 2014.

[86] L. Tang, Z. Yu, H. Wang, X. Zhou, and Z. Duan. Methodology and tools for pervasive application development. *International Journal of Distributed Sensor Networks*, 10(4):1–16, 2014.

[87] B. Bertran, J. Bruneau, D. Cassou, N. Loriant, E. Balland, and C. Consel. DiaSuite: A tool suite to develop sense/compute/control applications. *Science of Computer Programming*, 79:39–51, 2014.

[88] P. Jagtap, A. Joshi, T. Finin, and L. Zavala. Preserving privacy in context-aware systems. In *Proceedings of Conference on Semantic Computing*, pages 149–153, September 2011.

[89] V. Sacramento, M. Endler, and F. N. Nascimento. A privacy service for context-aware mobile computing. In *Proceedings of Conference on Security and Privacy for Emergency Areas in Communication Networks*, pages 182–193, September 2005.