Identity Management System Model in the Internet of Things

Belkacem Athamena [1], Zina Houhamdi [2]

[1] *Business Administration Department, College of Business, Al Ain University, UAE*
[2] *Software Engineering Department, College of Engineering, Al Ain University, UAE*

Abstract **– This paper describes the identity management system (IdMS) by defining system and user requirements. Additionally, it introduces the IdMS concept that approaches the things identity management. Moreover, the paper deeply describes the IdMS features using unified modelling language (UML) diagrams such as class, system, and sequence diagrams to show the main system functionalities. Ultimately, the suggested system is evaluated by comparing it with the existing systems and discussing the fulfilment of user and system requirements.**

Keywords **– Internet of Thing, Identity Management Systems, UML diagrams.**

1. Introduction

We conduct the study of an identity management system (IdMS) in the Internet of Things (IoT) because of the absence of supportable and simple solutions that solve part of the main technology for enablers. Some propositions are implemented; however, they require refinement and normalization [1], [2]. The complete power of the IoT implies going beyond the organization systems and integrating the user in the IoT, where the devices and information provided by the users are supported [3]. Accordingly, a new valuable services model that is user-centered IoT is proposed.

DOI: 10.18421/TEM94-04
https://doi.org/10.18421/TEM94-04

Corresponding author: Belkacem Athamena,
Business Administration Department, College of Business, Al Ain University, UAE.
Email: athamena@gmail.com

Received: 21 July 2020.
Revised: 23 September 2020.
Accepted: 06 October 2020.
Published: 27 November 2020.

The paper focuses on developing a new IdMS that includes things such as users and devices (actuators, sensors, computing devices, smart devices, etc.) in the IoT, and explains the collaboration between things. Finally, the proposed system is validated to test of the fulfillment with regard to the declared user and system requirements. We begin by conducting a survey of existing IdMSs and their communication processes suggested for IoT heterogeneous networks [4] (such as DNS and ONS [5], Cooltown [6], SAML [7], OAuth 2.0 [8], OpenID Connect [9], oneM2M [10], MAGNET [11], MANETs for healthcare applications [12], and identity management in M2M networks [13]). Then, we identify the user and system requirements. We define a use case diagram to explain the interaction between the environment and the system. Then, we propose a new IdMS framework that is user-centered. UML sequence diagrams are used to describe the system connection and authentication processes. System validation is performed by considering technical aspects.

2. Identity Management System Overview

IdMS manages identities individually and their privileges, roles, authorization, and authentication inside or outside the system boundaries to increase performance and security and to decrease delay, cost and repetitive operations. IdMS is a set of procedures organized manually and computerized. The IdMS's aim is the identification and management of system resource utilization and the support of data integrity and privacy. Additionally, IdMS is responsible for generating certificates, managing attributes and roles, controlling accesses and authentication. IdMS includes a collection of decentralized software resources and several network protocols. Furthermore, the interface of IdMS with business components and IdMS procedures conforms to human resources, legislation and ethical procedures of businesses. Figure 1 presents the IdMS architecture [14], where things such as simple devices (for example, sensors) and complex devices (for example, smartphones or mainframes) are shown. Each thing belongs to a specific user-space

and collaborates with other things despite their heterogeneity. Additionally, there are multiple services that require information gathered from internal or external sensors to be used in scenarios such as e-Health, enterprise and private companies. In the center of the IdMS architecture (between the services and things layers), there is a middleware layer that manages the communication between services and things securely [14].

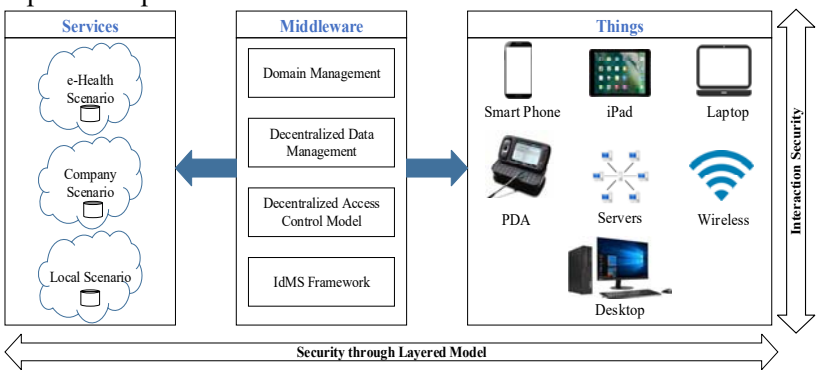

Figure 1. IdMS Framework

The IdMS design mainly balances security, software properties and protocol expenses to work effectively in wireless and mobile systems, therefore increasing overall benefits for the entire business application processing [15].

3. IdMS Requirements

In addition to the IdMS challenges, the IdMS has to satisfy several user requirements. The user is a human who uses the proposed single thing sign-on (STSO) for identification [16]. According to the literature [1], [2], [16], [17], [18], [19], the important requirements that influence the integration of STSO in IdMS are classified into two categories as follows.

3.1. System Requirements

The system requirements are privacy (data protection from unauthorized accesses), reliability (trustworthy communication), extensibility (appropriate APIs to integrate new devices), scalability (ability to grow, to manage the increased demands and to provide interaction between detected identifiers inside different domains) and flexibility (support of different types of devices).

3.2. User Requirements

Considering the technical and personal requirements and expectations of the end-users, there are two major user requirements influencing the STSO:

- End-users: They are part of the IoT ecosystem (it is a set of devices connected by networks that communicate with other devices, services, applications and people). Thus, their role is to inform their requirements, provide feedback and control the operators separately [20]. The integrated IdMS should personalize the users' profile and accordingly provide a set of services.
- Continuous receptive services: The system should fulfill and support the users' requirements independently of location and time. The IdMS aims to provide continuous receptive services, depending on a particular user's environment and running time, by defining communication mechanisms between things in IoT [20].

Our IdMS focuses on the user and his requirements to provide a highly usable and accessible system by the user through identifying the device identifier that is unique and automating the abilities, properties and information exchanged between devices. Note that IdMS allows the compatibility of heterogeneous devices connected by networks and provides the needed interaction and authentication methods to reduce the number of iterations. Figure 2 shows the IdMS configuration. It consists of a set of heterogeneous devices (such as PCs and laptops) used to access several services. The user has to provide his identity to devices to be identified by the service. In Figure 2, the identity is represented by a circle and the service by a rectangle.

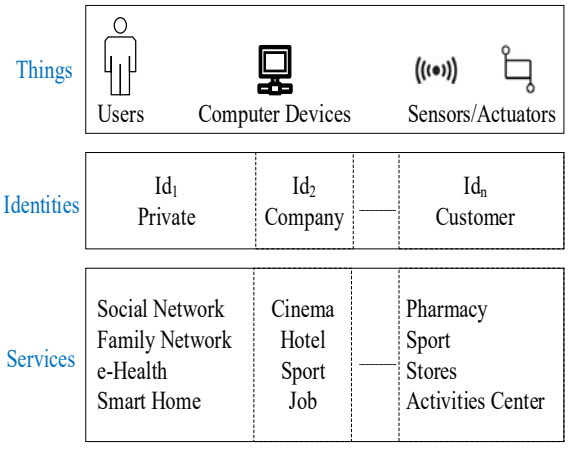

Figure 2. Access Management

4. IdMS Model

The physical framework and logical entities that are explicitly or implicitly involved in the communication are deeply described in this section, along with the IdMS characteristics.

4.1. IdMS Framework

Our suggested IdMS framework follows the federated IdMS model. The IdMS high-level architecture is represented by a system diagram, which is illustrated in Figure 3.

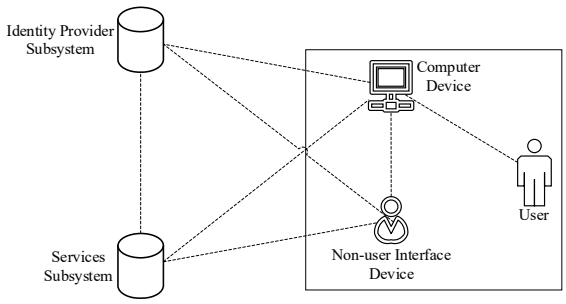

Figure 3. IdMS High-Level Architecture

The IdMS model consists of three subsystems:

1. *Things Subsystem* (TS): The TS represents "things" in the IoT that provide identities to the identity provider subsystem (IdPS) required for identifying the user. TS is composed of three different components:
   - *User*: a human accessing a set of services through device(s). The user's identity is generated by the IdPS.
   - *Computer Device*: resides in the device and serves to provide middleware services for the IdMS. From a security perspective, it prevents any illegal data changes or deterioration.
   - *Nonuser Interface Device*: required to collect reliable information concerning users' requirements, from common and/or private nonuser interface devices. Thus, it assists the service subsystem (SS) in providing continuous receptive services without delay to users.
2. *Identity Provider Subsystem* (IdPS): The IdPS is responsible for storing the entire identity data and authenticating users, devices and services. Furthermore, it manages access to nonuser interface devices. The IdPS can be located locally or remotely. The IdPS builds communications dynamically.
3. *Service Subsystem* (SS): The SS belongs to the IdMS service layer and authenticates users and things for remote and local services.

The user can establish private identifiers for his devices and create identities and provide them to IdPS for management. Concerning remote data where the identity provider is a cloud service, IdPS provides the choice to the IdMS to allow system scalability, global public access, and device mobility between places. In the IdMS model, authentication is necessary for services and users.

4.2. Logical Entities

The acquisition of required information is essential for establishing communication between shared devices. Therefore, there is a need for a method for managing the three types of logical entities: things (user, computer device and nonuser interface device), domain (zone where things act) and service provider (provides needed services to users).

These logical entities are connected by an obvious relationship: a user uses particular nonuser interface device information and operates by means of a specific computer device. These devices define a user-domain relationship that is considered in an access authorization decision. Accordingly, our IdMS uses an identifier representing a user who uses a set of things from a specific domain. The identifier is defined as (user, thing, and domain).

4.3. IdMS Characteristics

The system diagram of the IdMS is shown in Figure 4, which describes the IdMS components.

The IdMS is planned to be implemented as a cloud public service. The management of things identities is carried out by the identity controller and involves nonuser interface devices, IdPS and data processing. The identity controller is different from the identity provider (describes the identity management framework), and it supervises and governs relationships to specific domains. Each thing creates its identity. The user (considered a thing) identity is created and saved in IdPS through shared/personal devices.

Before benefiting from IdMS services, all things' identities are recorded in the IdPS by producing a smart-sheet. The communication is achieved via accessible IP addresses and using an identifier format. When a user/company registers to utilize the IdMS, a domain is created, and its *domId* is obtained. *domId* defines the domain identifier that is used to determine things in a set of domains. Thus, it makes the system scalable and devices mobile across different networks. Then, the things identities are created, and the devices' identity (*devId*) is created by IdPS, and the users' identity (*userId*) is created by the identity controller. The user establishes criteria to manage his identity through his credentials. The

credentials are defined during the creation of a user profile and used to manage settings and authorized personal data. The well-known types of user credentials are the pair (username, password) and biometric data. For computer and nonuser interface devices, identities are made in the IdPS component in the same way as for
users' identities. An object (device or service) is identified by *dtype* (device type), *devId*, and *domId* information.

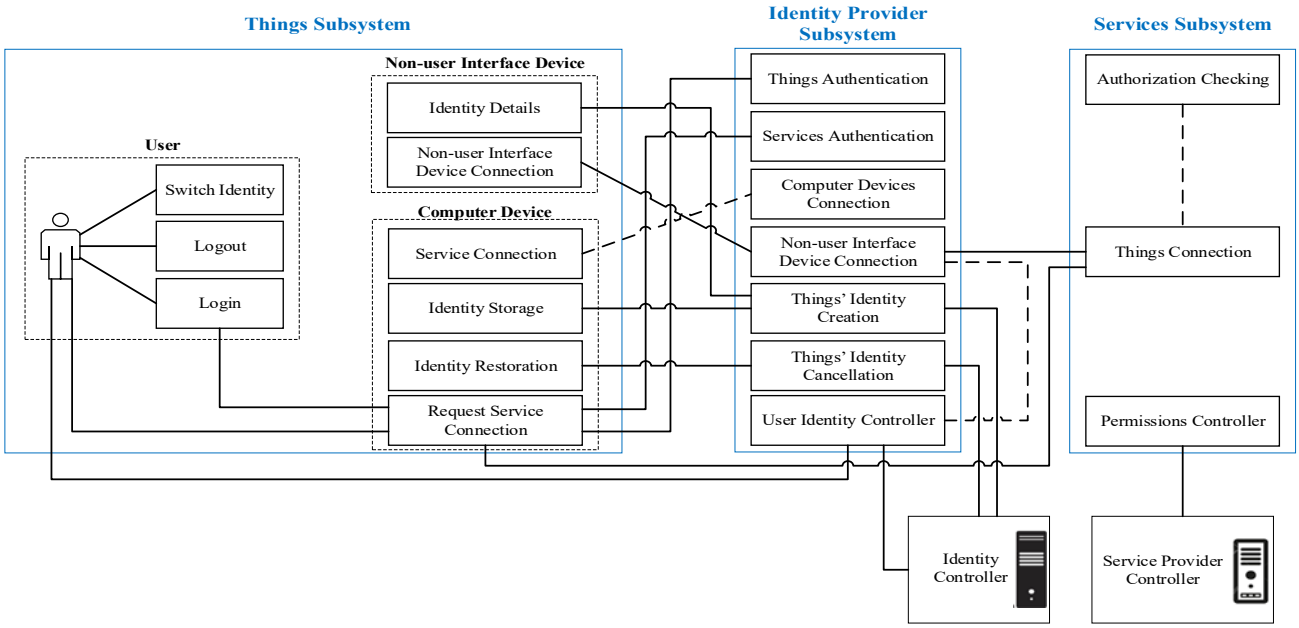

Figure 4. IdMS System Diagram

4.4. IdMS Structure

Figure 5 represents the IdMS UML class diagram that explains the suggested IdMS model.

The IdMS contains three core modules: TS, IdPS and SS.

- The TS module represents the information related to things. The identity agent (IdAgt) class provides authentication facilities to the highest level subsystems (for example, applications and the operating system). The computer device identities are stored in the TS. The TS interacts with the IdPS by providing information about nonuser interface devices, such as domain, identity and security information. The TS also contains an application class that defines the applications usage. Note that it is essential for applications to appropriately use the existing APIs provided by the system by calling the IdAgt and should be consistent with the service identifier claimed for access.
- The IdPS module provides identity information to the SS. It collaborates with the TS. The IdPS data are acquired and exploited only by reliable services (having pre-signed contracts with the IdPS). Reliable services and contracts are not addressed in this paper.
- The SS module, which can be remote or local, performs user device authentication by using identity information (*dtype*, *devId*).

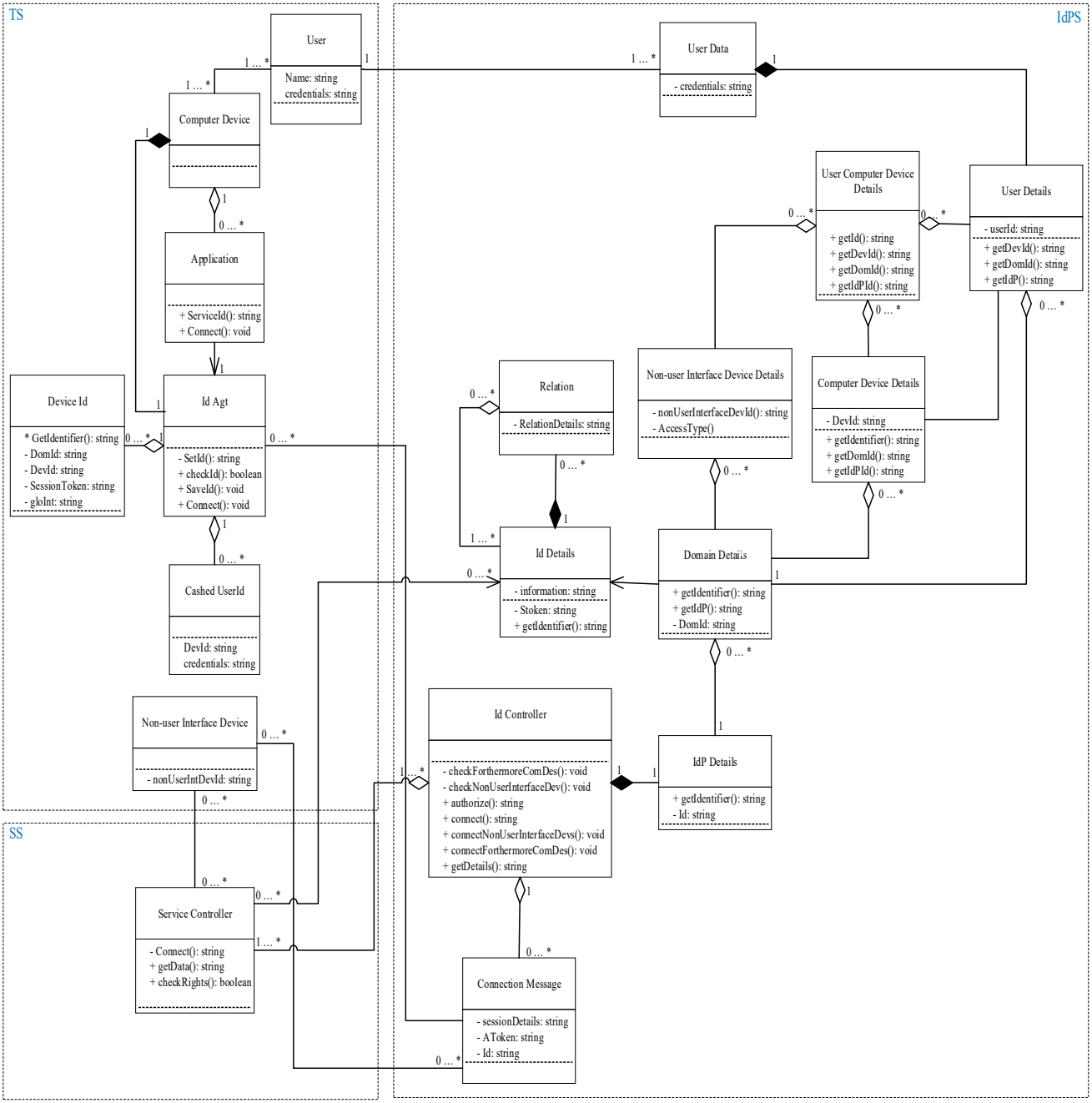

Figure 5. IdMS Class Diagram

5. IdMS Functionalities

The IdMS enables service access by identifying and authenticating things according to the user information. The IdMS functionalities depend on connection establishment and authentication. Section 5.1 comprehensively describes the connection establishment, and section 5.2 clearly explains the IdMS features and things actions.

5.1. IdMS Connection

Figure 6 shows the sequence diagram describing the messages involved in the IdMS connection.

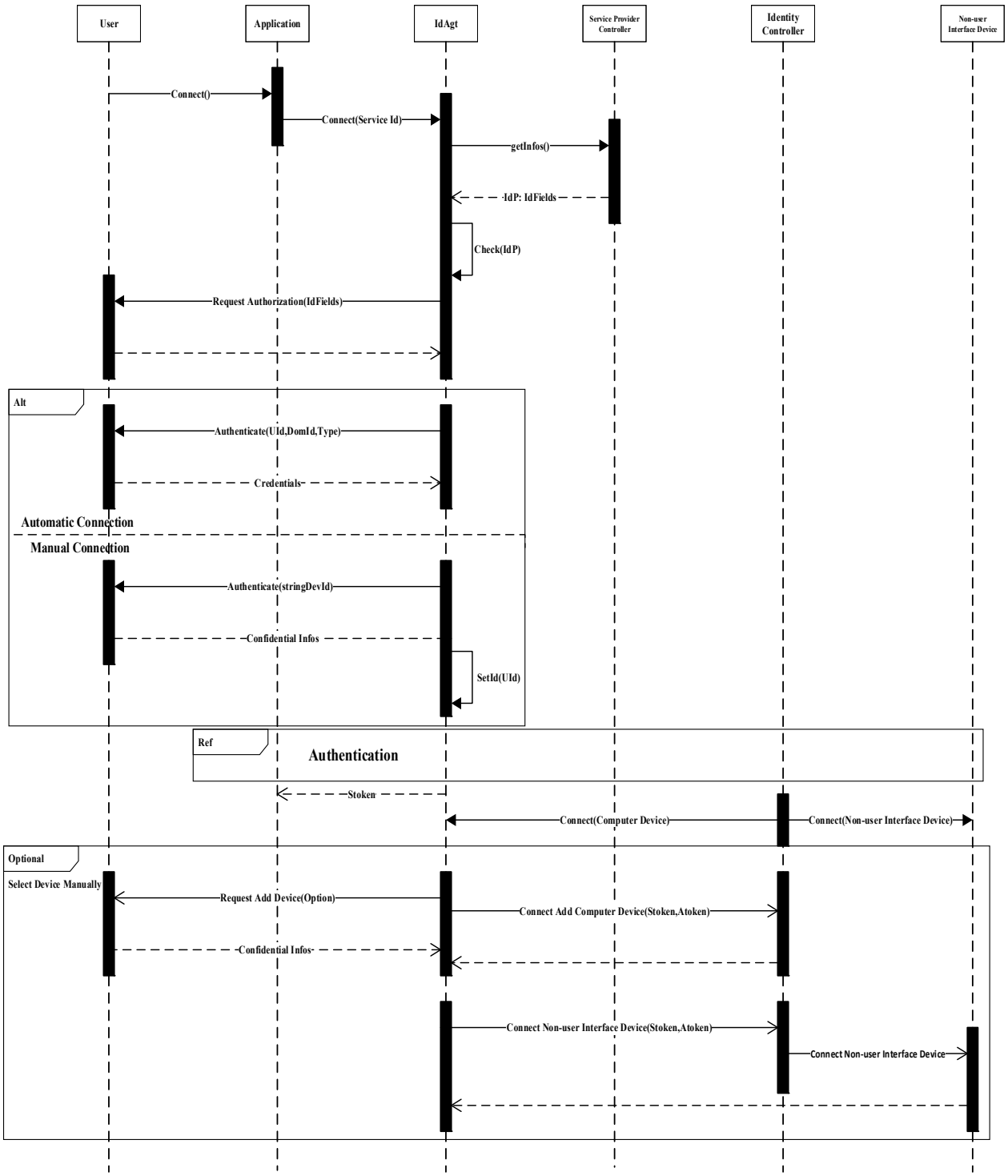

Figure 6. IdMS Connection Sequence Diagram

- The process starts when the user uses a computer device to access a local or remote service through an application.
- At this moment, the Application sends the request to the *IdAgt*, which provides the *serviceId* of the desired service to set up the connection with the associated service controller.
- The *IdAgt* obtains information about the service and identity fields.
- The *IdAgt* obtains information related to the required service from the service controller.
- The *IdAgt* tests whether the obtained *devId* is active and adequate for the service.
- Then, the user is asked to provide its identity fields to *IdAgt*.
- Figure 6 indicates two login options: automatic connection and manual connection. In the automatic connection, all user devices are connected automatically after the identification

of one thing (i.e., computer devices are recognized by the *IdP* based on user preferences), and IdMS is activated.
- However, in the manual connection, the user confirms his identity and selects the device for usage.
- Then, the *IdAgt* or *IdP* checks the thing's credentials.

After successfully completing authentication, the identity controller sets a session token (*Stoken*) and sends it to the *IdAgt*, which forwards the *Stoken* to the application. This later requires the *Stoken* to set up its connection to the service without *IdAgt*. The identity controller sends a request for device connection to the device list (both computer and nonuser interface) according to the actual user profile to connect automatically to other devices after the identification and connection to the first things (user or computer device).

Alternatively, if the user chooses manual connection to particular devices,

- The *IdAgt* sends a request to the user, asking the user to use another device to access the service.
- If the user agrees, he has to select a set of additional computer and nonuser interface devices and then specifies the access rights as an authentication token (*Atoken*), which is sent to the identity controller in a message.

The service controller asks for authentication from the identity controller (see Figure 7), after which

- The identity controller receives a message (containing *Atoken* and *Stoken*) from *IdAgt* requesting connection to the targeted device.
- After performing the authentication successfully, the identity controller broadcasts a connection message to the intended devices.

5.2. IdMS Authentication

Figure 7 presents the sequence diagram for the IdMS authentication process. This later performs some actions to determine and control the access rights to a specific service.

1. The authentication process begins whenever the *IdAgt* sends a message to the service controller requesting access to a particular service to check the authentication. The message contains data concerning the active *devId* or *IdPId*.
2. The service controller verifies and finds which *dtype* (nonuser interface device) needs to be authenticated depending on user preferences and service requirements by analyzing the *devId*/*IdPId*, adding *dtype* (if *dtype* is appropriate) and sending a request to the identity controller to check the device authentication.
3. The identity controller sends a message to *IdAgt* requesting identity information to check the authentication.

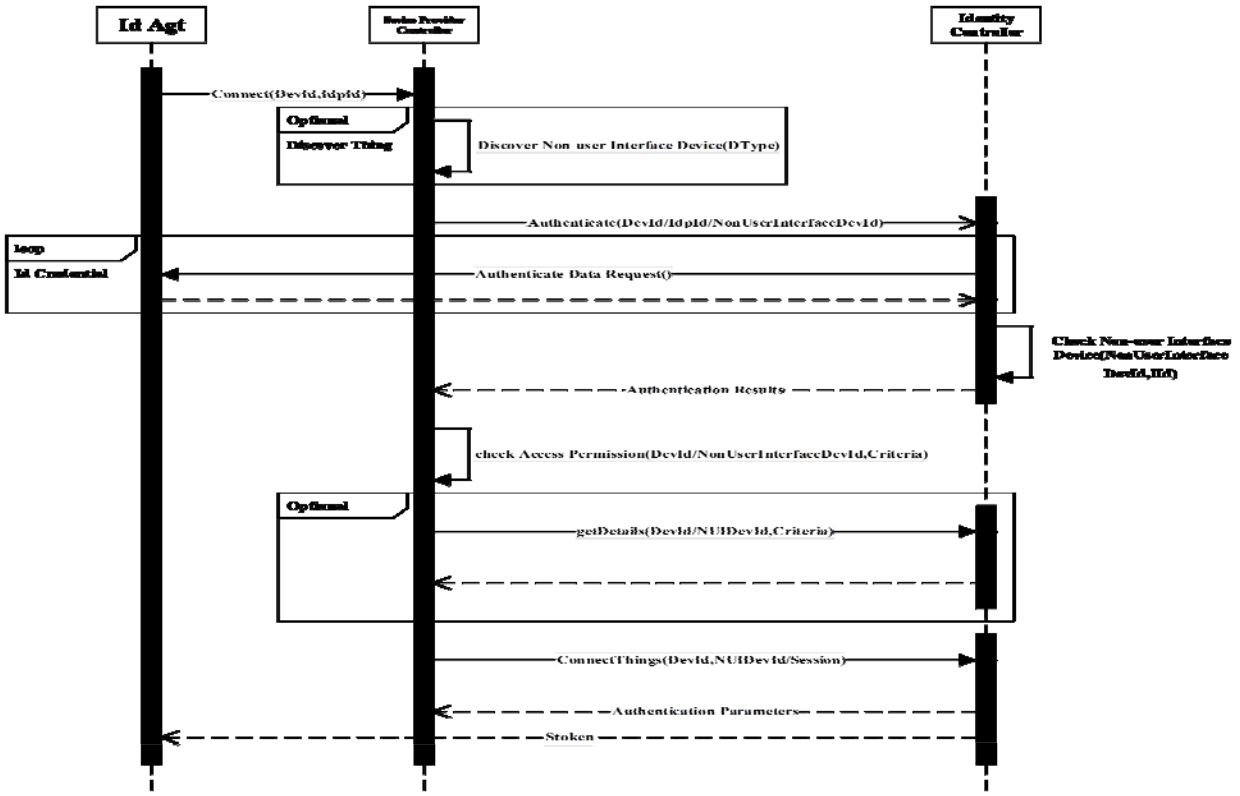

Figure 7. IdMS Authentication Sequence Diagram

4. The identity controller checks whether the received identities information of the thing is authorized to use a specific service. The process stops if the access is unauthorized.
5. (*devId*, *userId*) information is used by the identity controller to determine if the user has access rights to nonuser interface devices. Sometimes, the identity controller requires service controller identification.
6. If the device authentication succeeds, the identity controller notifies the service controller about the authentication result by sending the message *devId*.
7. The service controller performs its proper authentication process. It requires supplementary identity information from the identity controller related to the *devId*.
8. After successfully completing authentication, the service controller requests an identity controller to establish the connection between the service and authenticated devices.
9. Accordingly, the identity controller provides the identity information to devices and services and manages the interaction between them.
10. The service controller guarantees communication security by acquiring security parameters.
11. The device obtains *Stoken*, which contains information about identity data, security parameters, device, and service authentication.

6. Model Validation

In this section, we technically discuss the proposed model to determine if it fulfills all requirements, to compare the IdMS model with standard IdMSs, and finally to analyze our proposed IdMS in reference to seven laws.

6.1. Requirements Fulfillment

The suggested IdMS focuses on users, and its features are necessary to ensure a new approach for automatic identification. The IdMS is examined by considering the IdMS requirements in IoT and user's goals and understanding these requirements. IdMS is flexible since the identification process uses all things in the IoT based on IdMS features. The IdMS allows communication between a network's devices, regardless of whether they are explicitly or implicitly used by the user. Furthermore, the IdMS is scalable and mobile because identity management is cloud-based. Our IdMS supports different protocols to deliver information (possibly in different formats such as XML or JSON). IdMS is extensible since it does not limit the vendor or type of the devices as far as they can communicate. Conceptually, IdMS is considered reliable since it considers the relationships between the logical entities in its data flow process and the authentication process considers the failure and users' preferences and provides a substitute method to log in and manually access services. The IdMS necessitates authentication for any services accordingly, the data confidentiality is considered by IdMS. The identity controller monitors services restrictions, and the service controller controls the access constraints to a service.

6.2. IdMS Evaluation

This section compares the suggested IdMS with reference to IdMSs in IoT.

Device-based system: The comparison between our IdMS and standard device-based systems reveals some similarities due to the considerations of the advantages of these systems when designing the IdMS. Our IdMS is extended to include identity management, data security and authentication, which makes it more complex. The IdMS is similar to the device-based system Cooltown because both allow the thing's identification and data communication between recognized things in the system. Additionally, the IdMS is comparable to ONS since both allow device discovery and identity data management. However, IdMS is end user-oriented and focuses on the devices regardless of their technology, but ONS is an abstract technique for managing devices.

User-centered: The IdMS focuses on the user and integrates the user and device identity. The computer device uses embedded software components to independently transmit messages. Moreover, in the *IdPS*, users' profiles contain data concerning nonuser interface devices. The IdMS facilities are extended to explore things and differentiate users depending on predefined contracts. The IdMS allows the storage and selection of several user identities in the *IdPS*. The management of identities is a completely computerized process, and the complete identity details are delivered upon services authentication, similar to *OpenID*. SAML acts on the web level; however, the IdMS operates on the entire things level in the IoT to enable unique things, similar to OAuth2. Additionally, the authorization technique (for the IdMS, identity data and services) controls the access to resources of the systems. Concerning identity authentication and management, IdMS is closer to *OpenID* Connect. IdMS is similar to MAGNET in user profile customization and service-oriented property personalization. Additionally, Paraktish IdMS is comparable to the IdMS since they depend on the context of things' identities, and the users are not engaged directly as an essential component of IdMS.

Identity seven laws: Our system is analyzed by considering the "seven identity laws" established by Cameron [21], with regards to users' privacy as a main aspect requiring identities access to web services.

1. *Minimal revelation of constrained usage*: IdMS limits the provision of identity details (*userID*) before the authentication of the service. Moreover, IdPS informs the things, users or devices, about required identity data requested by the service. The service controller oversees the services identities, and the identities' sharing between different services is prohibited.
2. *User control and consent*: This is approached by IdMS, which gives a better understanding to users about acceptable identity and identity details while (even before) logging into the system (determined by the IdPS-service provider cooperation).
3. *Justifiable Parties*: Our IdMS meets this requirement since the identity data is accessible only after the successful completion of the authentication procedure.
4. *Directed Identity*: The IdMS supports directed identity because of the use of the partial device identifier (*devId*) and the provision of global ownership/interface in the process of public and private things identification.
5. *Diversity of Technologies and Operators*: There is an absence of studies in the cooperation between different IdPs obstacles and the IdMS to address this requirement. From the users' point of view, they can utilize and take advantage of different IdPs despite their technologies.
6. *User Involvement*: Users are integrated and considered within system components by providing them the possibility of controlling and setting rules and participating directly or indirectly in the communication process. The IdMS meets this requirement because the identification and authentication methods are considered a main component of the system.
7. *Simple and Consistent Experience*: The IdMS guarantees users a simple and unified experience and enables context separation via diverse technologies and operators.

6.3. IdMS Competitive Advantages and Values

- Reduction in manual password usage: a simple approach to be used by the user to access services.
- Identity management relies only on devices (computer devices and nonuser interface devices).
- Automatic identification and provision of IdMS feature (time saving).
- Continuous responsive services independent of location and time.
- User-centered application services.
- Context-based applications.

Accordingly, after implementing and testing our system successfully, the IdMS can be employed in multiple application domains, including but not limited to customized e-Wellness and e-Health applications (hospital and pharmacy), organizations (for example, for accessing purposes), supermarkets (for queue reduction by recognizing users automatically and using their bank information for payment) and smart transportation services.

7. Conclusion

The IdMS faces a set of challenges, such as security mechanisms, communication protocols, device diversity and software properties. This paper defines the IdMS by identifying user and system requirements. Additionally, it introduces IdMS concepts to address things identity management and proposes the usage of STSO identity. The IdMS functionalities, for example, authentication and connection of computer and nonuser interface devices, are deeply described. The validation of the suggested IdMS is addressed to determine the IdMS advantages.

Information communication technology solutions can use the STSO feature to contribute to successful business models that provide business intelligence and suffer from chaotic technologies. Since the proposed IdMS design is open, the IdMS is categorized as future potential for developing and integrating new properties, and accordingly, the IdMS is considered a first step toward evolving the IoT to be the Internet of Everything, including people.

References

[1]. Ibarra-Esquer, J. E., González-Navarro, F. F., Flores-Rios, B. L., Burtseva, L., & Astorga-Vargas, M. A. (2017). Tracking the evolution of the internet of things concept across different application domains. *Sensors*, *17*(6), 1379.

[2]. El-hajj, M., Fadlallah, A., Chamoun, M., & Serhrouchni, A. (2019). A survey of internet of things (IoT) Authentication schemes. *Sensors*, *19*(5), 1141.

[3]. Moreno, M. V., Ramos, J. L. H., & Skarmeta, A. F. (2014, March). User role in IoT-based systems. In *2014 IEEE World Forum on Internet of Things (WF-IoT)* (pp. 141-146). IEEE.

[4]. Someswar, G. M., & Reddy, M. M. (2018). Design And Development Of A Suitable Identity Management Framework In Heterogeneous Internet of Things. *Compusoft*, *7*(3), 2716-2732.

[5]. Hamzic, A., & Olofsson, I. (2016). DNS and the Internet of Things: Outlining the challenges faced by DNS in the Internet of Things. (Dissertation).

[6]. Trnka, M., Cerny, T., & Stickney, N. (2018). Survey of Authentication and Authorization for the Internet of Things. *Security and Communication Networks*.

[7]. Mollaei, N., Shirazi, H., & Pourebrahimi, A. (2019). SAML Standard Optimization for Use on CoAP-Based Web Servers on Internet of Things. *Journal of Soft Computing and Information Technology*, *8*(1), 14-23.

[8]. Polu, S. K. (2018). OAuth based Secured authentication mechanism for IoT Applications. *International Journal of Engineering Development and Research (IJEDR),6*(4), 409-413.

[9]. Ofleh, O. (2018). *Future of Identity and Access Management: The OpenID Connect Protocol* (Doctoral dissertation).

[10]. Wu, C. W., Lin, F. J., Wang, C. H., & Chang, N. (2017, September). OneM2M-based IoT protocol integration. In *2017 IEEE Conference on Standards for Communications and Networking (CSCN)* (pp. 252-257). IEEE.

[11]. Prasad, R. (2010). *My personal adaptive global NET (MAGNET)*. R. Prasad (Ed.). Dordrecht, Heidelberg, London, New York: Springer.

[12]. Chibelushi, C., Eardley, A., & Arabo, A. (2013). Identity management in the Internet of Things: the role of MANETs for healthcare applications. *Computer Science and Information Technology*, *1*(2), 73-81.

[13]. van Thuan, D., Butkus, P., & van Thanh, D. (2014, October). A User Centric Identity Management for Internet of Things. In *2014 International Conference on IT Convergence and Security (ICITCS)* (pp. 1-4).

[14]. Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications*, *88*, 10-28.

[15]. Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 291-319.

[16]. Miraz, M. H., Ali, M., Excell, P. S., & Picking, R. (2015, September). A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT). In *2015 Internet Technologies and Applications (ITA)* (pp. 219-224). IEEE.

[17]. Tabassum, K., Ibrahim, A., & El Rahman, S. A. (2019, April). Security issues and challenges in IoT. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-5). IEEE.

[18]. Verma, P. K., Verma, R., Prakash, A., Agrawal, A., Naik, K., Tripathi, R., ... & Abogharaf, A. (2016). Machine-to-Machine (M2M) communications: A survey. *Journal of Network and Computer Applications*, *66*, 83-105.

[19]. Todorov, D. (2007). *Mechanics of user identification and authentication: Fundamentals of identity management*. CRC Press.

[20]. Tabassum, K., Ibrahim, A., & El Rahman, S. A. (2019, April). Security issues and challenges in IoT. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-5). IEEE.

[21]. Cameron, K. (2005). The laws of identity. *Microsoft Corp*, *5*, 8-11.