Design of a Security Gateway for iKaaS Platform

Seira Hidano[1(✉)], Shinsaku Kiyomoto[1], Yosuke Murakami[2], Panagiotis Vlacheas[3], and Klaus Moessner[4]

[1] KDDI R&D Laboratories, 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan
se-hidano@kddilabs.jp
[2] KDDI Research Institute, Tokyo, Japan
[3] WINGS ICT Solutions, Athens, Greece
[4] University of Surrey, Surrey, UK

Abstract. The iKaaS (intelligent Knowledge-as-a-Service) platform integrates the data on multiple local clouds organically and provides the data to various types of applications as knowledge while taking security and privacy fully into account. However, access control on the iKaaS platform is not without complications because the application may access personal data in different countries from the one where the application exists. We thus design a security gateway that is set at the entrance of each local cloud and can control access while interpreting the differences in regulations and guidelines between countries.

Keywords: Access control · Security policy · Privacy certificate

1 Introduction

The Internet of Things (IoT) paradigm is rapidly gaining momentum in modern wireless telecommunications. IoT devices, such as smart sensors designed to monitor temperature, pressure and other environmental conditions and wearable devices to measure an individual's state of health, generate vast amounts of time sequence data. These data are accumulated on clouds and analyzed for useful information like personal preferences and to predict the environmental conditions surrounding people and the next actions that people may take. The impact will increase if the heterogeneous data stored on multiple clouds can be organically integrated. However, vast quantities of potentially correlated data have not yet been analyzed in correlated contexts for a number of reasons. As the data obtained from IoT devices are mostly sensitive information related to an individual, anxiety concerning security and privacy is an obstacle to the participation of users. A universal data model is also required for the analysis of the heterogeneous big data obtained from various types of sensors. Furthermore, there are legal considerations that complicate matters further. The compatibility of regulations related to personal data should be clearly dealt with. It is expected that with increasing trust, decentralized multi-cloud environments are about to unlock great potential for future data analysis [3,4].

Y. Zhang et al. (Eds.): CloudComp 2015, LNICST 167, pp. 323–333, 2016.
DOI: 10.1007/978-3-319-38904-2_34

The iKaaS (intelligent Knowledge-as-a-Service) platform thus has been proposed as a way to resolve these problems [6]. On this platform, a global cloud is hierarchically built atop multiple local clouds that are set up in different countries. It integrates the data stored on the local clouds organically, and the integrated data are provided to various applications as knowledge. Security and privacy are controlled by a security gateway that is set at the entrance of each local cloud. When using the iKaaS platform, the application can access the data for different countries, conduct various-scale analyses and compare different countries. However, privacy issues have not been sufficiently resolved in the current model. When the application accesses personal data in different countries, the iKaaS platform is required to handle the data in accordance with the regulations and guidelines governing personal data in both the country where the application exists and the country where the local cloud is set up. These regulations and guidelines are complicated and there are differences between countries. For instance, a Japanese act [7] permits the transfer of personal data to the EU while an EU directive [2] does not permit the transfer of the data to Japan. Although there have been few technical studies on security and privacy for decentralized multi-cloud environments, these studies have not focused on privacy issues in relation to cross-border data [5,9]. In order to resolve these issues, there needs to be fundamental review of the architecture.

The main contribution of this paper is to design a security gateway that can interpret the differences in regulations and guidelines between countries and is capable of flexibly controlling the access permissions of the application while taking privacy into consideration. The rest of the paper is organized as follows: Sect. 2 overviews the functional capabilities of the iKaaS platform and our contributions. Section 3 proposes the security and privacy architecture for the iKaaS platform. Section 4 describes the protocol whereby the application accesses the data through the security gateway. Section 5 presents the conclusions of this paper.

2 iKaaS Platform

Intelligent Knowledge-as-a-Service (iKaaS) is a concept model where the data accumulated on multiple local clouds are organically integrated on a global cloud and the data are provided as knowledge taking security and privacy into consideration. Figure 1 shows the architecture of the iKaaS platform. The iKaaS platform encompasses a global cloud, multiple local clouds, IoT devices and third party DBs, which are hierarchically arranged. The multiple local clouds are established in different countries such as in the UK and Japan, and each local cloud has DBs for various types of data. The data are obtained not only from the newly available IoT devices but also from existing DBs designed for other purposes. The various data models are converged to a universal data model before the data are stored in the DBs on the local cloud. This resolves the issue where the models and formats of the data are different between local clouds. The global cloud is considered to be a trust component, and all data are provided to the applications through it.

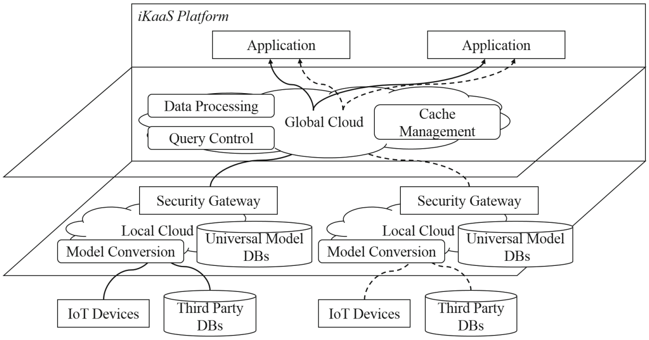

Fig. 1. Architecture of iKaaS platform.

The global cloud has three functions: query control, data processing and cache management. When the application makes a request for knowledge, the global cloud helps the application to generate queries consistent with the objective and transmits the queries to suitable local clouds. The global cloud not only deals with the raw data obtained from the local clouds but also processes the data statistically depending on the request. Massive-scale big data and heterogeneous data are combined and analyzed, and more useful knowledge is produced as a result. Self-Organizing Maps (SOMs) is an indicative technique to process the big data (in terms of disparate data formats and diversity of sources), offering user-friendly or oriented insightful knowledge visualization, for data mining with a high degree of accuracy so as to support decision making [1]. Additionally, the data are stored in a cache DB on the global cloud so that the application can access them more effectively. The global cloud manages the cache data according to the frequency with which the data are updated (because some data, like map information, for example, CityGML [10], are not useful if the data exist as an outdated version). However, the data that IoT devices extract are mostly sensitive information related to an individual, namely, personal data. Personal data should not be disclosed or provided to third parties without the consent of the data owner. There is also the case when the transfer of personal data to third parties is not permitted under the relevant regulations. The security gateway is thus arranged at the entrance of the local cloud and controls the access of the application to the data with privacy and security considerations taken into account.

Our Contributions. The contributions of our work are the following:

- We design a security gateway that can interpret the differences in the rules between the country where the application exists and the country where the local cloud is set up, to control the access permissions of the application taking the above privacy issues into account. The interpretation is realized using a privacy certificate and security and privacy policies, which are defined in Sect. 3.

- We introduce the access control with a token, which allows the process of the above interpretation to be omitted for the same application. This is because while the application may frequently request data at short intervals as the data are continuously transmitted from IoT devices, the interpretation process takes time.
- We provide a method by which the security gateway determines the validity of the application without communicating with it directly since the global platform inevitably intervenes between them. This method is achieved by combining the privacy certificate and the public and private keys of the application.
- We elucidate the concept of cache management because sensitive information could be cached. The process of determining whether the data are to be cached and the deletion of the cache data is conducted based on the cache policy defined in the local cloud. Additionally, the cache data are encrypted with the encryption algorithm recommended in each country.

3 Functions of Security Gateway

Figure 2 shows privacy-conscious architecture centered on the security gateway for the iKaaS platform. Each local cloud has a security gateway at the touch point with the global cloud. The queries from the application and the data on the local cloud are all exchanged through the security gateway. The security gateway has two functional capabilities: one is the access control of the application under the rules governing the handling of personal data both for the country of the application and the country of the local cloud, and the other is privacy control on behalf of the data owners on the local cloud. The privacy certificate issued by the privacy certificate authority (CA) and the security policy are referred to by the security gateway for access control, and the privacy policy is formulated for privacy control. The privacy certificate and the policies are defined in

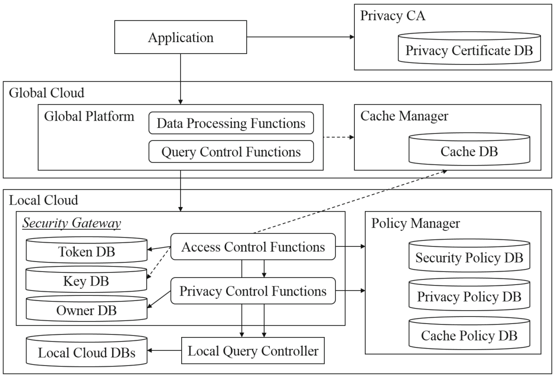

Fig. 2. Privacy-conscious model of iKaaS architecture.

Sects. 3.1, 3.2 and 3.3. The methods for both controls are explained in detail in Sect. 3.4. Furthermore, we elucidate the concept of cache management on the global cloud in Sect. 3.5. The cache manager is set up to control access to the cache data while cooperating with the security gateway. Management of the cache data is conducted taking both privacy and usability into consideration as mentioned in Sect. 2.

3.1 Privacy Certificate

The privacy certificate makes it possible for the security gateway to interpret the rules in the country where the application exists. The privacy CA is built for each country and creates the privacy certificate on the basis of the national regulations that prevail in that country and information concerning the application. It is a requirement that the application is issued a privacy certificate by the privacy CA of the same country before requesting the local cloud for data. The following parameters are listed on the privacy certificate:

- *CA Country:* The name of the country where the privacy CA is established. It also refers to the name of the country where the application exists.
- *Application IP:* The IP address of the application.
- *Application ID:* The type of service that the application provides.
- *LC Countries:* The names of the countries that the application is permitted to access.
- *LC Data IDs:* The identifiers indicate the types of data that the application is permitted to access. The values are nested in each value of *LC Countries.*
- *Expiry:* The expiry date of the privacy certificate.
- *Application PK:* The public key of the application. (The role of this key is mentioned in Sect. 3.4.)
- *Signature:* The signature is generated with the private key of the privacy CA. The public key is distributed to security gateways.

3.2 Security Policy

The security policy is created by the administrators of the local cloud based on several regulations and guidelines through the policy manager. The policy manager is provided by the privacy CA in charge of the country where the local cloud is set up, and the basic policy is formulated in accordance with the national regulations in advance. The administrator configures the security policy on the basis of the local regulations, such as a city, a town or a company, and in accordance with relevant guidelines. The security policy has two tables: the expiry periods of access permissions and the definitions related to data privacy. Tables 1 and 2 are an example of a security policy. The administrators configure the values for each type of data, namely, data ID. Each row of the tables indicates the rules related to a regulation or a guideline (including the basic policy). The expiry periods of access permissions are defined for each country as shown in Table 1. A zero value means that the data of the type indicated by the data ID

Table 1. Expiry periods.

No	Data 1	⋯	Data N
1	UK 0/JP 2mo	⋯	UK 0/JP 0
2	UK 1h/JP 2h	⋯	UK 0/JP 0
⋮	⋮	⋱	⋮

Table 2. Definitions on data privacy.

No	Data 1	⋯	Data N
1	Non-privacy	⋯	Privacy
2	Privacy	⋯	Privacy
⋮	⋮	⋱	⋮

are not permitted to be transferred to the country. This table is configured for each type of application (although Table 1 shows the configuration for a specific application). The value "Privacy" in Table 2 means that the privacy of the data of the type indicated by the data ID should be taken into account and the value "Non-privacy" means that there are no privacy-related concerns regarding the data. Even if the data are defined as "Non-privacy" in Table 2, the expiry periods are set in terms of security or with the frequency with which the local cloud DB is updated.

3.3 Privacy Policy

The status of the consent on the transfer of data to third parties for each data owner is listed on the privacy policy [8]. Personal data should be controlled by the data owner in terms of privacy as mentioned in Sect. 2. The privacy policy makes it possible for the security gateway to provide personal data to the application while preserving the privacy of the data owner. Table 3 is an example of a privacy policy. The status of the consent is defined for each data ID by each data owner. The value "Yes" means that the data owner agrees that the data can be used on the iKaaS platform and the value "No" means that the data owner does not agree. This table is configured for each type of application (although Table 3 shows the configuration for a specific application).

Table 3. Privacy policy.

Owner ID	Data 1	⋯	Data N
1	Yes	⋯	Yes
⋮	⋮	⋱	⋮
K	No	⋯	Yes

Table 4. Cache policy.

No	Data 1	⋯	Data N
1	2mo	⋯	5days
2	2mo	⋯	2wk
⋮	⋮	⋱	⋮

3.4 Access and Privacy Control

The security gateway uses a token to control the access of the application because the process involved in checking the privacy certificate and the policies takes time. When an application requests access to the local cloud DBs, the security gateway generates a token and returns it to the application. The application with the token can request the data any number of times until the token has expired. The security gateway provides two functions for access control: *Issue Token* and *Get Data*.

Issue Token. This function is called when the application obtains the token to access local cloud DBs. The application is then required to specify the data IDs that it wants to have access to and sends the privacy certificate. When issuing the token, the security gateway refers to the privacy certificate and the security policy in order to comply with the rules of both countries.

The values of parameters *LC Countries* and *LC Data IDs* listed on the privacy certificate are checked first. The security gateway uses those values to confirm whether or not the application is permitted to access the data specified by the data IDs under the rules of the country where the application exists. Meanwhile, the security gateway uses the security policy to interpret the rules of the country where the local cloud is set up. The table on the expiry periods for access permissions as shown in Table 1 is used to decide the expiry date of the token. The security gateway searches the corresponding columns of data IDs in the table using the values for the parameter *Application ID* listed on the privacy certificate and the requested data IDs. The security gateway chooses the one with the shortest expiry period for the country name corresponding to the value of the parameter *CA Country* listed on the privacy certificate and derives the expiry date by adding the current time to the chosen period. If the multiple data IDs are specified by the application, this process is carried out for each data ID, which means multiple expiry dates are set for one token. After deciding the expiry dates, the security gateway generates a token. However, if the shortest expiry period equals zero for all the data IDs, the token is not issued to the application.

Next, the security gateway refers to the table on the definitions of data privacy as shown in Table 2 to judge whether or not the privacy of the data owners is to be taken into consideration when the application with the token requests the data. The security gateway refers to the corresponding columns of data IDs in the table using the same process that was used to make the decision regarding the expiry date of the token. If the value "Non-privacy" is set for all rows, the data of the type indicated by the data ID are called non-privacy data, and the security gateway determines that there are no privacy-related concerns regarding the data. Otherwise, the data are called privacy data, and the security gateway takes the privacy of the data owner carefully into consideration when transferring the data. This process is also carried out for each data ID.

The values of the parameters *Application IP* and *Application ID* listed on the privacy certificate, the token, the data ID that the application can have access to, the expiry date of the token and the privacy type (non-privacy data or privacy data) are associated and stored in the token DB. If the application can access multiple data IDs with one token, the set of the values of the data ID, the expiry date and the privacy type are created for each of the data IDs and the multiple sets are all associated with one token. Additionally, the token is transmitted to the application after being encrypted with the value of the parameter *Application PK* listed on the privacy certificate. This is because as the security gateway does not directly communicate with the application on the iKaaS platform, conventional schemes that are used to confirm the validity of the application, such as the SSL client authentication, cannot be applied. In our architecture, the application to which the privacy certificate is issued by the privacy CA only can decrypt the token with its own private key, which prevents unauthorized use of the token by other applications.

Get Data. The application with a token calls this function via the global platform to transmit the query to the local cloud DBs. First, the security gateway verifies the authenticity of the token. This verification is conducted on the basis of a message authentication code (MAC). In other words, the token is treated as a common key. We do not assume that a specific algorithm is used to generate the MAC because the recommended algorithms are different for each country. The security gateway is thus required to inform the application of the algorithm when issuing the token. The privacy of the data owner is safeguarded when the data are transferred to the application. The security gateway checks the privacy type of the data that the application has requested to have access to. If the data is of the non-privacy type, the data is directly returned to the application. However, if there are privacy considerations, the security gateway filters the data on the basis of the privacy policy. As the relation between the ID of the data owner and the attributes is stored in the owner DB, the security gateway extracts the corresponding IDs in the owner DB with the owner attributes specified in the query. The security gateway searches the corresponding rows in the privacy policy with the extracted owner IDs and confirms the consent status on the transfer of the data for the application ID and the data IDs specified in the query. The security gateway only returns the data for which the data owner has set the value "Yes".

3.5 Cache Management

Cache functions are required when the application wants access to data more effectively. When the application obtains the data through the security gateway, the communication cost increases as compared to the case where the DBs are accessed directly. If using the cache manager, the application can access the data in fewer steps, and for that reason it is expected that the access time can be shortened.

The data on the local cloud are directly cached from the security gateway through the cache manager when the token to access the data is requested by an application. The expiry date of the cache data is decided on the basis of the cache policy. Table 4 shows an example of the cache policy. The expiry periods are configured based on various types of requirements. This is because some types of data, like map information, must be kept fully up to date, so the expiry periods should be set taking not only security and privacy into account but also the frequency with which the data need to be updated. The cache policy is formulated in the same manner as the security policy by the security gateway when the expiry date is decided.

Furthermore, it is not desirable to store sensitive information in a third party domain for an extended period of time due to security issues, and consequently there is a requirement for the cache data to be encrypted. However, because the recommended algorithms are different for each country, before transmitting the data to the cache manager, the security gateway encrypts the data using the algorithm recommended in the country of the local cloud. The encryption key is stored in the key DB and returned to the application with the token.

4 Protocol

We define the protocol for security and privacy control on the iKaaS platform. We provide the definition of a query, and then show the sequence of steps involved in issuing a token and the data request. In our architecture, the security gateway provides the functions as web APIs, and HTTPS connections only are allowed. Additionally, the global cloud instantiates a global platform for each application, and the application cannot use the global platform for any other application.

4.1 Query Formats

The security gateway has no function for interpreting the query for local cloud DBs (*LCD-query*). When the application requests the data, the headers (*SGW-headers*) are added by the global platform and the query for the security gateway (*SGW-query*) is created. The types of headers are as follows:

- *Application IP:* The IP address of the application.
- *Application ID:* The type of service that the application provides.
- *LC Data IDs:* The IDs indicate types of data that the application wants to have access to.
- *Owner Attributes:* The attributes that narrow down the data owners, for example, age and gender. This header is required when the attributes are specified as search conditions in the *LCD-query.* The security gateway uses the values to extract the corresponding owner IDs from the owner DB as mentioned in Sect. 3.4.
- *Time Stamp:* The time when the *SGW-query* is generated. It is used to prevent a replay attack.

4.2 Step Sequence

Token Issuance. The token to access data on the local cloud is issued to the application in accordance with the following procedure:

1. An application requests the privacy CA to issue the privacy certificate.
2. The application uses some functions for query control that the global platform provides in order to search the security gateway of the country where the local cloud DBs suited for the objective exist and to request the issuance of a token.
3. The global platform calls the function *Issue Token* that the security gateway provides. The global platform then specifies the data IDs that the application wants to access and sends the privacy certificate of the application.
4. The security gateway confirms the values of the parameters *Expiry* and *Signature* listed on the privacy certificate to verify the validity of the certificate. The signature is validated with the public key of the privacy CA of the country indicated by the parameter *CA Country* listed on the privacy certificate.
5. The security gateway creates a token, encrypts it with the public key of the application, which is listed on the privacy certificate, and returns the encrypted token to the application via the global platform.
6. The application decrypts the token with its own private key and stores the token on the global platform.

Data Request. The application with a token obtains the data on the local cloud as follows:

1. An application uses some query functions of the global platform and creates the *SGW-query*. The global platform generates the MAC of the *SGW-query* with the token of the application.
2. The global platform calls the function *Get Data* that the security gateway provides to transmit the *SGW-query* and the MAC to the security gateway.
3. The security gateway extracts the corresponding token from the token DB with the values of the *Application ID* and *Application IP* headers and checks the expiry date of the token.
4. The security gateway generates the MAC from the *SGW-query* and the token to verify the authenticity of the query. The value of the *Time Stamp* header is also confirmed.
5. The security gateway transmits the *LCD-query* to the local query controller.
6. When the data are returned from the local cloud DBs, the security gateway confirms the privacy type of the data while searching the token DB.
7. If the data stored as non-privacy data are returned, the security gateway returns the data to the application via the global platform without further intervention. Otherwise, the processes of Steps 8–10 are carried out.
8. The security gateway extracts the corresponding owner IDs from the owner DB using the value of the *Owner Attributes* header.
9. The security gateway searches the privacy policy using the extracted owner IDs and the values of the *Application ID* and *LC Data IDs* headers and confirms the consent status of the corresponding data owners.
10. The security gateway extracts the data on the condition that the data owner agrees to the transfer and returns the extracted data to the application via the global platform.

5 Conclusion

The iKaaS (intelligent Knowledge-as-Service) platform integrates data on multiple clouds organically and provides the data as knowledge to the cross-border application. We designed a security gateway that makes it possible to control the access of applications on the iKaaS platform. The security gateway can interpret the differences between countries in terms of their respective regulations and guidelines that govern the treatment of personal data by using the privacy certificate issued by the privacy certificate authority (CA) and the security policy on the local cloud. It also has a function that allows the availability of personal data to be controlled according to the consent status of the data owners.

Acknowledgment. The work is supported by the EUJ-1-2014 Research and Innovation action: iKaaS; EU Grant number 643262, Strategic Information and Communications R&D Promotion Programme (SCOPE), Ministry of Internal Affairs and Communications, Japan.

References

1. Bantouna, A., Poulios, G., Tsagkaris, K., Demestichas, P.: Network load predictions based on big data and the utilization of self-organizing maps. Springer J. Netw. Syst. Manage. **22**, 150–173 (2014)
2. EU: Directive 95/46/EC of the European Parliament and of the Council of 24 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, October 1995
3. EU FP7/ICT project 257115: OPTIMIS: Optimized Infrastructure Services, June 2010–May 2013
4. EU FP7/ICT project 287708: iCore: Internet Connected Objects for Reconfigurable Eco-systems, October 2011–September 2014
5. EU FP7/ICT project 609094: RERUM: REliable, Resilient and secUre IoT for sMart city applications 2013–2016
6. EU HORIZON 2020 project 643262: iKaaS: intelligent Knowledge-as-a-Service 2014–2017
7. Japan: Act on the Protection of Personal Information, Act No. 57 of 30 May 2003
8. Kiyomoto, S., Nakamura, T., Takasaki, H., Watanabe, R., Miyake, Y.: PPM: Privacy policy manager for personalized services. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES Workshops 2013. LNCS, vol. 8128, pp. 377–392. Springer, Heidelberg (2013)
9. de Meer, H., Pöhls, H.C., Posegga, J., Samelin, K.: On the relation between redactable and sanitizable signature schemes. In: Jürjens, J., Piessens, F., Bielova, N. (eds.) ESSoS. LNCS, vol. 8364, pp. 113–130. Springer, Heidelberg (2014)
10. Oosterom, P.V., Zlatanova, S., Fendel, E.M.: Geo-Information for Disaster Management. Springer, Heidelberg (2005)