

# A Method to Solve the Interoperability between Clouds in Semantic Web of Things

Ce Li<sup>1, a</sup>, Ning Liu<sup>2, b</sup>, Guanyu Li<sup>3, c</sup>, Lingyu Deng<sup>4, d</sup>

<sup>1,2,3,4</sup> Faculty of Information Science & Technology Dalian Maritime University

Dalian, China

<sup>a</sup>alice224@163.com, <sup>b</sup>775460536@qq.com,

**Keywords:** Cloud Computing; Semantic Web of Things; Interoperability; Cloud broker; Ontology

**Abstract.** In the Semantic Web of Things (SWoT), different type of sensors will collect a large number of real-time data and heterogeneous information. How to deal with these data and provide support for decision-making is a problem. Cloud Computing mode is particularly suitable for handling the large data sets. It has recently emerged as a new paradigm for providing various services over the Internet. However, in order to provide better services to consumers, cloud providers have to establish partnerships with others, and they often use non-standard formats. So interoperability issues exist in cloud computing environments. In this paper, we analysis existing interoperability challenges. After that we formulate an Ontology based Cloud Service Broker and the semantic technology is the key catalyst to support the broker. Moreover, interoperability is determined at the data level as well as the service level. It is an important issue to discuss how to represent the cloud resources and services. We build upon the current state-of-the-art cloud ontologies to provide formally represented and then achieving interoperability.

## Introduction

Semantic collaboration can be introduced into Internet of Things (IoT) to form Semantic Web of Things (SWoT) to promote semantic interoperability. Cloud Computing is an emerging Internet model for hosting and delivering services over the Internet, and its' main purpose is to provide Everything as a Service (XaaS) to the cloud consumers such as companies, internet users and other clouds [1].

As Cloud Computing develops, many believe that it will reshape the IT industry as a revolution [2]. But till now, there seems to be no consensus on what Cloud is. The National Institute of Standards and Technology (NIST) offers a general accepted definition for Cloud Computing; It summarizes Cloud Computing as: "Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [3]." NIST also outlines five major roles in a Cloud ecosystem namely provider, consumer, auditor, broker and carrier [4, 5]. We build upon a Cloud Resource Broker managing the use, performance and delivery of cloud services and the allocation between different cloud service providers and then achieving interoperability in cloud environment.

The related concepts is discussed in Section 2. An architecture, basic theory, workflow of this system and main algorithm of this method is described in Section 3. In the last section, conclusions are made and future work is presented.

## Related concepts

**SWoT.** The concept of Internet of things can be traced back to Auto-ID center which was founded in 1999 at the MIT. They introduced the characteristics of the IoT, the related technology and the

challenge we were facing. The Semantic Web of Things is a merging concept from the Semantic Web (SW) and the Web of Things (WoT) [5, 11]. The terminology hierarchy of SWoT is shown in Fig. 1.

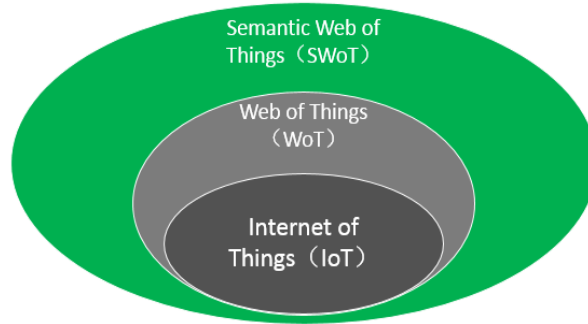


Figure 1. Terminology Hierarchy

**Interoperability.** IEEE defined interoperability as the ability of two or more systems or components to exchange information and to use the information that has been exchanged [3]. Since Cloud Computing is made up of a variety of layered elements, services in every layer are called homogeneous and services in different layers are called heterogeneous [3, 11]. Cloud interoperability refers to the ability of a cloud service works with more than one homogeneous or heterogeneous cloud services to improve its service. The relation of interoperability between clouds is shown in Fig. 2.

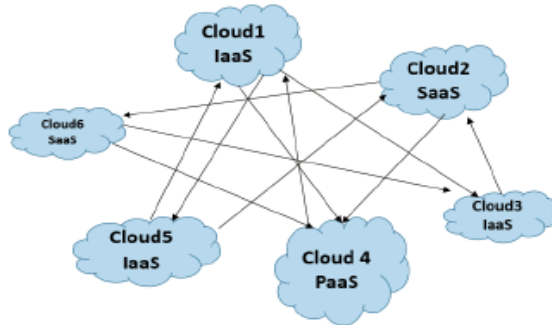


Figure 2. Interoperability between Clouds

**Cloud broker.** Cloud broker is an entity which manages the use, performance and delivery of cloud services and negotiates relationships between cloud providers and cloud consumers [6]. Among the problems that Cloud broker should be coping with, cloud service discovery and selection are the most important two. In next section, we will present OWL-S based dynamic cloud service discovery and selection system.

### 3. An architecture to solve the interoperability issue between clouds in SWoT

**Basic theory.** The formal definition of Service can be defined as Definition 1:

**Definition 1 (Service):** Given  $IN_S$  is a set of inputs and  $OUT_S$  is a set of outputs of service  $S$ , and  $C_S$  is an optional set of constrains. Service  $S$  is represented as:  $S_S = (IN_S, OUT_S, C_S)$ .

By this definition, a requested service is represented as  $S_R = (IN_R, OUT_R, C_R)$  and a published service is represented as:  $S_P = (IN_P, OUT_P, C_P)$ .

**Definition 2** described the definition of Service Matching:

**Definition 2 (Service Matching):** Matching between a service request  $S_R$  and an advised service  $S_P$  can be of the following five types:

- Exact match:  $S_R$  exactly matches with  $S_P$  in all of  $IN$ ,  $OUT$  and  $C$ .

- Subsume match: If  $IN_R \supseteq IN_P$ ,  $OUT_R \supseteq OUT_P$ , and  $C_P$  implies with  $C_R$ , we can project that  $S_R$  subsume match with  $S_P$ .
- Invert-subsume match: If  $IN_P \supseteq IN_R$ ,  $OUT_R \supseteq OUT_P$ , and  $C_R$  implies with  $C_P$ , we can project that  $S_R$  Invert- subsume match with  $S_P$ .
- Partial match:  $S_R$  partially match with  $S_P$  if and only if  $(IN_R \cap IN_P) \neq \emptyset$ ,  $(OUT_R \cap OUT_P) \neq \emptyset$ , and the match is none of above-mentioned three.
- Fail match: If the match is none of the above- mentioned four.

By definition 2, we also define a semantic matching partial order on  $Y$  such that: Exact match  $\gg$  Subsume match  $\gg$  Invert-subsume match  $\gg$  Partial match  $\gg$  Fail match.

By this partial order, we can optionally let the user point out a lower bound on the desired service matching and then we can filter out the services user undesired. For instance, if the lower bound the user specified is subsume match, the matched services should be either subsume match or exact match. Then later part can handle further processing exactly.

Definition 3, Definition 4 and Definition 5 described the definition of User Preference, Service Discovery and Service Selection respectively:

**Definition 3 (User Preference):**  $UP$  (User Preference) is a tuple  $(UP_{IO}, UP_{CON})$ , where the tuple values the lower bound on semantic matching required for input, output and constraints. So  $UP_{IO}, UP_{CON} \in \{\text{Exact match, Subsume match, Invert-subsume match, Partial match, Fail match}\}$ .

**Definition 4 (Service Discovery):** Given a cloud service request  $S_R (IN_R, OUT_R, C_R)$  and  $UP$ . Service discovery is a process to identify all  $S_P (IN_P, OUT_P, C_P)$  such that:

- $IN_P$  and  $OUT_R, IN_R$  and  $OUT_P$ :  $IOMatching(IN_R, OUT_R, IN_P, OUT_P) \gg UP_{IO}$ .
- $\forall C_R, \exists C_P$ :  $ConstraintMatching(C_R, C_P) \gg UP_{CON}$ .

**Definition 5 (Service Selection):** Given a list  $D=\{S, M\}$ , where  $S$  is a discovered service and  $M=(M_{IO}, M_{CON})$  is a set of matched levels for  $S$ . Service Selection denotes a procedure that outputs a ranked list  $D^*$  using a scoring mechanism based on  $M$  [7].

**System architecture.** The proposed cloud resource broker to achieve interoperability between clouds is shown in Fig.3.

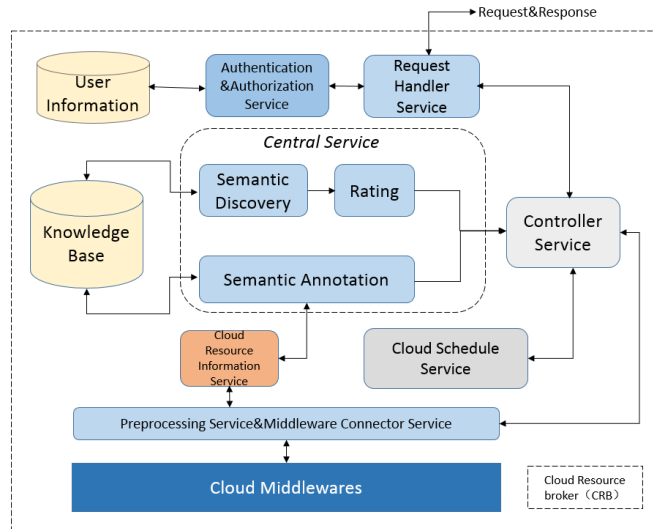


Figure 3. System Architecture

The system architecture primarily concentrated on three hierarchies such as Service Discovery, Service Se-lection and Service Provision [4].

The process of discovering suitable cloud services consists of two major blocks:cloud service description and cloud service discovery [13]. We construct a semantic model to describe the service offers and requests, and then perform matching and selection. The core component of the semantic technology is ontology. By providing a common cloud ontology using OWL-S, we have incorporate

the semantic based cloud resource description in the knowledge base and discover the cloud resource information from the knowledge base.

The OWL-S is designed for the purpose of those applications that need to process the content of information instead of just presenting information to the users. It comprises three main parts: Service Profile, Service Model and Service Grounding [7, 8]. OWL-S Service Profile provides both the functional and non-functional information needed for an agent to discover services, while once service is found, OWL-S Service Model and OWL-S Service Grounding, take together, and provide enough information for an agent to make use [8]. We use Protégé 3.4 (Figure.4) ontology editor to create the cloud ontology which can describe the services offers and requests [9]. And then we can perform discovery action. The cloud ontology we defined is the domain precise [7], and it provides a series of concepts such as hasBandwidth, hasCity, hasCloudSystem, hasUsage and etc.

| OWL Class           | OWL Object Property | OWL Datatype Property |
|---------------------|---------------------|-----------------------|
| owl:Thing           | Object properties   | hasBudgetMax          |
| Cloud_Type          | hasBandwidth        | hasBudgetMin          |
| CloudOffer          | hasCity             | hasCloudType          |
| CloudProvider       | hasCloudSystem      | hasDataBaseSizeMax    |
| CloudRequester      | hasCountry          | hasDataBaseSizeMin    |
| CloudRequest        | hasCPU              | hasDataBaseType       |
| Feature             | hasFeature          | hasDataTransferMax    |
| Architecture        | hasHardware         | hasDataTransferMin    |
| Cost                | hasLocation         | hasMatchType          |
| Functionality       | hasOffer            | hasOS                 |
| Usability           | hasPricePlan        | hasPlanType           |
| Usage               | hasRequest          | hasStorageMax         |
| Vendor_Reputation   | hasSoftware         | hasStorageMin         |
| Hardware            | hasStorage          | hasUsageCost          |
| Price_Plan          | hasUsage            | hasUsageMax           |
| ProgrammingLanguage |                     | hasUsageMin           |
| Region              |                     | hasUsagePrice         |
| Software            |                     |                       |
| swrla:Entity        |                     |                       |

Figure 4. Cloud Service Ontology

**Main algorithm of the method.** To solve the interoperability between clouds in SWoT, a system architecture based on this method is proposed in this paper. Cloud Service Discovery and Selection Algorithm is the key to achieve the goal of the system [10, 13]. Algorithm CSDaSA of service discovery and selection is shown as follows:

---

**Algorithm: CSDaSA**

---

INPUT: Application requests

OUTPUT: Create virtual resource and run the application requests  
Pre-requisite: Start the resource broker to retrieve the registered cloud resource information by interfacing with cloud middleware and annotate/describe the same in the knowledge base.

1.BEGIN

2.Get and parse the application requirements such as operating system, CPU version, Storage, disk memory and, etc. generate the application objects.

3.Do Matchmaking with available cloud resources for application execution and virtual resource creation for each application request.

4.Find out the capacity of matched resources in terms of ram speed ram speed for every request and order the resources.

5.Do Number of virtual resources can be created = Physical Resource Capacity/Application Requirements for every matched resource list.

6.Select the resource based on the semantic score of the matched service.

7.Invoke the appropriate cloud middleware connectors and provision the virtual cloud resources and run the application in virtual resources which have been created.

8. END

---

We design a semantic matching formula to compute the semantic score.

Set  $X = \{IO, Constraint\}$  and  $Y$  is a list of semantic matching types as in Definition 3. We make  $nY$  ( $X$ ) as the number of semantic matching type  $Y$  when matches based on  $X$ . So  $n(IO)$  is the number of

Exact matches for IOmatching. Score E>Score S>Score I >Score P>Score F. We defined semantic score of a service as:

$$Scores = w_1 * \frac{\sum nY(IO) * S_Y}{\sum nY(IO)} + w_2 * \frac{\sum nY(Con) * S_Y}{\sum nY(Con)} \quad (1)$$

Where  $w_1$  and  $w_2$  are real numbers between 0 and 1, representing the weighting of the components.

**The concrete workflow of the system.** The overall architecture of the system work flow is presented in Fig.5, and it works as follows: the cloud services providers offer services to the system, and the advertised services are aggregated to form the semantic repository. The repository consists of two parts, domain ontologies and service ontologies.

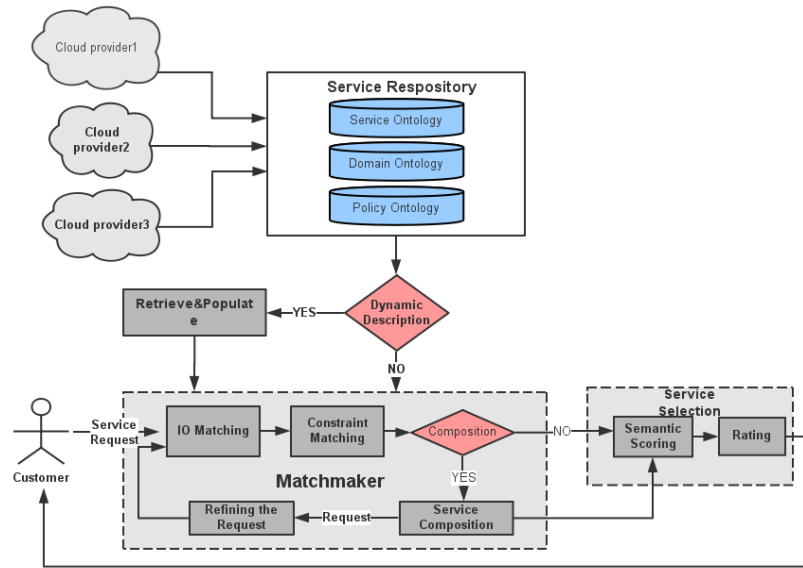


Figure 5. Workflow of the System

As we can see from Fig.5, Matchmaker is the core component which matches the services in the repository against the given services requests and then find the best match(es). Firstly perform IO matching which is followed by constraints matching. This is down over all services in the repository. A simple composition is attempted by refining the request and re- invoking the matchmaker if necessary. Before matchmaking, an additional step is needed to handle dynamic service descriptions. The service descriptions are retrieved on-the- fly and the retrieved descriptions populated into the ontology. The discovered cloud services are scored using a semantic scoring function and the scoring criteria is what we defined in formula (1). After the scoring module, the discovered services returned to the user as the best-matched service(s).

## Conclusion

Based on existing abroad research on interoperability between clouds in SWoT, a method based on cloud broker is proposed, and an architecture based on this method is designed. Service discovery and service selection are the cores of this architecture. The core component Matchmaker of this architecture, which matches the services in the repository against the given services requests and then find the best match(es). It can makes more convenience for users to operate cloud sources, decision-making and it also do the theoretical and technical support for interoperability between clouds in SWoT. Furthermore, the study need to be further improved: Firstly, exploring how to incorporate the Open Virtualization Format (OVF) to solve the interoperability between different cloud middleware; Secondly, how to extend our architecture to support for QoS.

## Acknowledgement

This work is financially supported by the National Natural Science Foundation of China (Grant No.61371090).

## References

- [1] Malik Khalfallah, Mahmoud Barhamgi and Nicol as Figay. A Novel Approach to Ensure Interoperability Based on a Cloud Infrastructure[C]. Proceedings of the 19th ISPE International Conference on Concurrent Engineering, 1143-1154. DOI:10.1007/978-1-4471-4426-7\_96.
- [2] Pahl C, Zhang L, Fowley F. A Look at Cloud Architecture Interoperability through Standards[C]// The Fourth International Conference on Cloud Computing, Grids, and Virtualization. 2013:7-12.
- [3] Rashidi B, Sharifi M, Jafari T. A Survey on Interoperability in the Cloud Computing Environments [J]. International Journal of Modern Education & Computer Science, 2013, 6(6):17-23.
- [4] Somasundaram, T. S., Govindarajan, K., Rajagopalan, M. R., & Rao, S. M. (2012, April). An architectural framework to solve the interoperability issue between private clouds using semantic technology. In Recent Trends In Information Technology (ICRTIT), 2012 International Conference on (pp. 162-167). IEEE.
- [5] NIST, National Institute of Standards and Technology (2011), [http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\\_cloud-definition.pdf](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf).
- [6] Meder, Human Empowerment in a Semantic Web of Things. Engineering and Technology (2014).
- [7] Ngan, Le Duy, and Rajaraman Kanagasabai. "Owl-s based semantic cloud service broker." Web Services (ICWS), 2012 IEEE 19th International Conference on. IEEE, 2012.
- [8] Nambi S N A U, Sarkar C, Prasad R V, et al. A unified semantic knowledge base for IoT[C]// Internet of Things (WF-IoT), 2014 IEEE World Forum on. IEEE, 2014:575-580.
- [9] Kaushik A, Chana I. Ontology based Cloud Framework [J]. Ijca Special Issue on Advanced Computing & Communication Technologies for Hpc Applications, 2012, ACCTHPCA (1):37-41.
- [10] Rimal B P, Choi E, Lumb I. A Taxonomy and Survey of Cloud Computing Systems[C]// 2009 Fifth International Joint Conference on INC, IMS and IDC. IEEE Computer Society, 2009:44-51.
- [11] Berners-Lee T, Hendler J, Lassila O, et al. The Semantic Web", Scientific American [J]. Lecture Notes in Computer Science, 2001, 284(October):34--43.
- [12] García-Galán J, Trinidad P, Rana O F, et al. Automated configuration support for infrastructure migration to the cloud [J]. Journal of Pharmaceutical Sciences, 2015, 55(3):293-300.
- [13] Sun L, Dong H, Hussain F K, et al. Cloud service selection: State-of-the-art and future research directions[J]. Journal of Network & Computer Applications, 2014, 45(10):134-150.
- [14] Shah H, Elzur U, Thaler P A. Method and system for network configuration and/or provisioning.