

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/309709143>

A multi-agent system approach for trustworthy cloud service discovery

Article in *Cogent Engineering* · November 2016

DOI: 10.1080/23311916.2016.1256084

CITATIONS

4

READS

52

3 authors, including:



Akinwale Akinwunmi

Bowen University

18 PUBLICATIONS 21 CITATIONS

[SEE PROFILE](#)



Emmanuel Olajubu

Obafemi Awolowo University

31 PUBLICATIONS 38 CITATIONS

[SEE PROFILE](#)

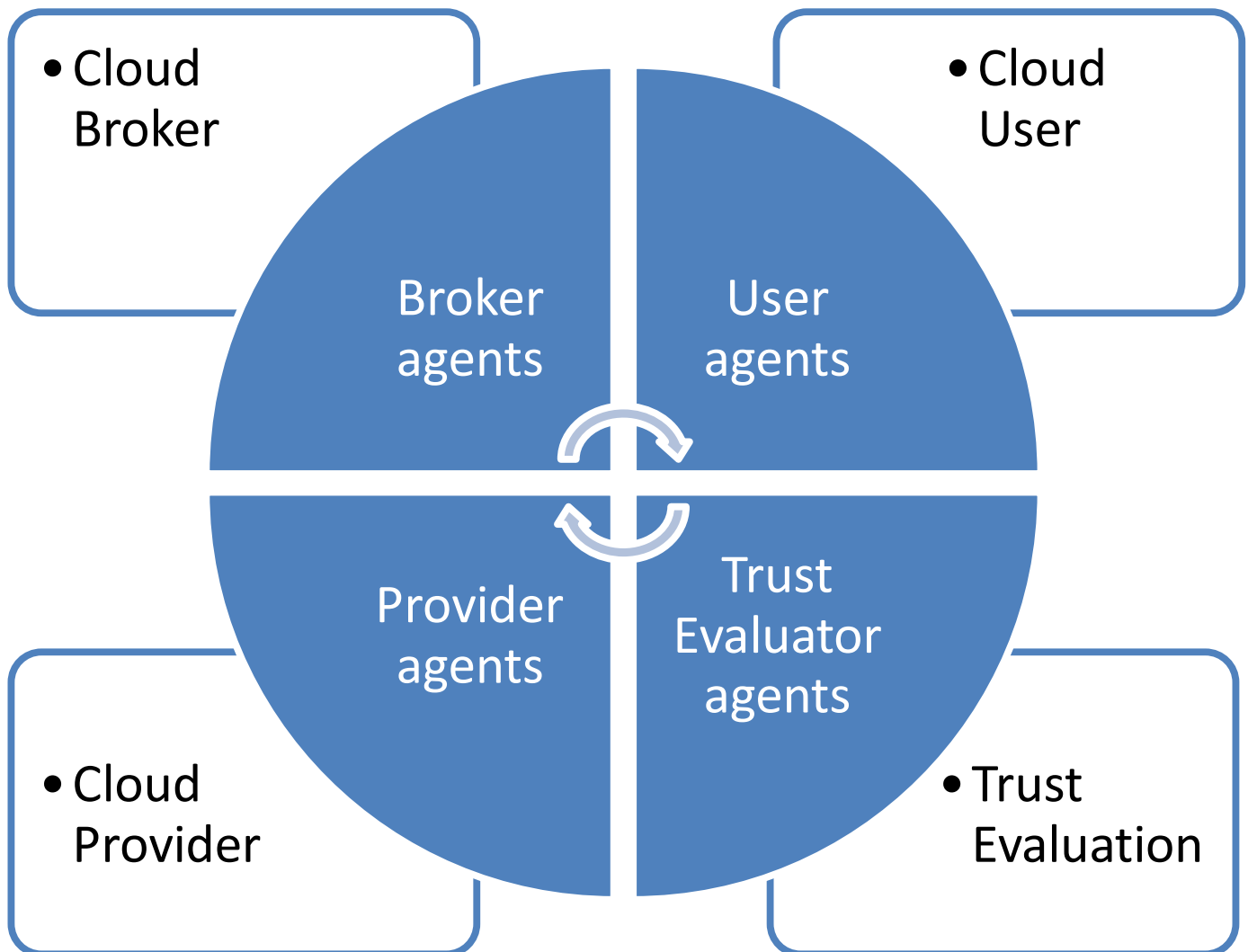
Some of the authors of this publication are also working on these related projects:



Information Privacy [View project](#)



e-learning [View project](#)



COMPUTER SCIENCE | RESEARCH ARTICLE

A multi-agent system approach for trustworthy cloud service discovery

A.O. Akinwunmi, E.A. Olajubu and G.A. Aderounmu

Cogent Engineering (2016), 3: 1256084



Received: 02 September 2016
Accepted: 29 October 2016
First Published: 04 November 2016

*Corresponding author: A.O. Akinwunmi,
Computer Science & Information
Technology Department, Bowen
University, Iwo, Nigeria
E-mails: akinwale.akinwunmi@bowenuniversity.edu.ng, walebunmi@gmail.com, ppp_wale@yahoo.com

Reviewing Editor
Sing Kiong Nguang, University of
Auckland, New Zealand

Additional information is available at
the end of the article

COMPUTER SCIENCE | RESEARCH ARTICLE

A multi-agent system approach for trustworthy cloud service discovery

A.O. Akinwunmi^{1*}, E.A. Olajubu² and G.A. Aderounmu²

Abstract: Harnessing the benefits of cloud computing, demands that a potential user has adequate access to trustworthy cloud services. Locating resources and services in a heterogeneous network such as cloud environment may be a tedious task. The cloud computing is a form of distributed system with a complex aggregation of computing resources from different domains with different administrative policies but having immense benefits that could enhance the mode of computing. Discovering cloud services is complicated by the varied level of quality of service offered by the different providers, hence a monolithic approach to cloud service discovery cannot effectively support the cloud environment. This work proposed a distributive approach to discovering trustworthy cloud services with the use of Multi-Agents System for achieving intelligent cloud service discovery. Simulation experiments were carried out and the results showed that extending the Multi-Agents System approach by integrating trust into the process of cloud service discovery improves the quality of service in terms of response time and scalability. A further

ABOUT THE AUTHORS

A.O. Akinwunmi (PhD), lecturer at the Department of Computer Science and Information Technology, Bowen University, Iwo, Nigeria. Assistant Director, Information and Communication Technology (ICT) Directorate Bowen University with research interest in communication, distributed systems, mobile computing among others.

E.A. Olajubu (PhD), senior lecturer in the Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife with expertise in computer communications (networks), distributed computing, parallel computing among others.

G.A. Aderounmu, (PhD), professor in the Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife. Director, Information Technology and Communications Unit Obafemi Awolowo University, Ile-Ife, Fellow of Nigeria Computer Society, Registered Professional Engineer with Council for Regulation of Engineering in Nigeria with expertise in computer communications and network management among others.

The research is about developing reliable and trustworthy systems, particularly focusing on the application of distributed computing approaches to addressing problems that are usually difficult to solve monolithically in order to ensure efficient resource discovery and sharing.

PUBLIC INTEREST STATEMENT

Locating resources and services in a heterogeneous network such as cloud environment is a tedious task. The cloud computing is a form of distributed system with a complex aggregation of computing resources from different domains with different administrative policies but having immense benefits that could enhance the mode of computing. Therefore, this study proposed a multi-agent system approach to mitigate problem of discovering trustworthy cloud services with a view to harnessing the benefits of the technology. A further comparative analysis of MAS approach for cloud service discovery to monolithic approach showed that it is highly efficient and highly flexible as well as scalable for trustworthy cloud service discovery.

comparative analysis of Multi-Agents System approach for cloud service discovery to monolithic approach showed that it is highly efficient, and highly flexible for trustworthy cloud service discovery.

Subjects: Mathematics & Statistics; Computer Science; Engineering & Technology

Keywords: cloud service; multi-agent; service discovery; distributed computing; functional analysis

1. Introduction

A cloud is a form of parallel and distributed system consisting of a collection of inter-connected and virtualised computers. Their resources are provisioned dynamically and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and users (Buyya, Yeo, & Venugopal, 2008). It offers a new model where computing resources such as power, storage and online applications are shared as “services” over the internet (Vaquero, Rodero-Merino, Caceres, & Lindner, 2009). It is a paradigm that focuses on sharing data and computations over a scalable network of nodes. Examples of such nodes include end user computers, data centres, and cloud services. Such a network of nodes is termed as a cloud (Mei, Chan, & Tse, 2008). The cloud computing is a distributed system with a complex aggregation of computing resources from different domains with different administrative policies but having immense benefits that could enhance the mode of computing (Akinwunmi, Olajubu, & Aderounmu, 2014).

In a multi-domain environment such as cloud that hosts thousands of service instances, discovering cloud services is an enormous task owing to the fact that there is a large number of different services from different providers having different levels of quality of service. A service discovery system that will ensure the correctness and completeness of discovery is germane to having trustworthy cloud services in cloud environment than in a small, restricted and homogeneous environment. Completeness of service discovery requires the system to retrieve relevant results from across the domain boundaries and across potentially-large network distances. Complementarily, a high degree of correctness is necessary so that these relevant results do not get lost among a flood of a possibly huge number of irrelevant results (Kousiouris, Kyriazis, Varvarigou, Oliveros, & Mandic, 2012). A monolithic approach to service discovery cannot support the diversity of the cloud environment that is distributed across the various domain boundaries and across different heterogeneous networks. Discovering a trustworthy cloud service requires a cloud service discovery system that is distributive and not monolithic in its approach. Hence, this work aimed at using the multi-agent system (MAS) another distributed system as a tool for discovering trustworthy cloud services across the domain boundaries in cloud environment. In Section 1.1, we describe related work and appraised them. In Section 2, the conceptual architecture design was explained and its components were discussed. In Section 3 the simulation experiments carried out and their results were also highlighted, followed by conclusions in Section 4.

1.1. Related work

MAS is a distributed system consisting of multiple software agents, which form “a loosely coupled network, called a multi-agent system (MAS), to work together to solve problems that are beyond their individual capabilities or knowledge of each entity” (Durfee & Montgomery, 1989). MAS are a community of autonomous agents working together in order to achieve a goal (Wooldridge, 2009). Of particular interest are MAS in which the individual agents display significant intelligence and autonomy. Over the years, MAS technologies have found applications in many distributed systems such as distributed problem solving, distributed information fusion, and distributed scientific computing (Drashansky, Houstis, Ramakrishnan, & Rice, 1999; Honavar, Miller, & Wong, 1998; Khosla & Dillon, 1997).

Although there are many differences between cloud computing and MAS, both are two distributed computing models, therefore several common problems can be identified and many more benefits

can be obtained by the combined use of cloud computing systems and multi-agents (Talia, 2011). There are several researches that have attempted the use of agent technology in cloud computing.

Kang and Sim (2010) provided a four-stage, agent-based cloud service discovery protocol. Software agents containing negotiation agents, brokering agents and information agents were used for bolstering the resource management system because of their scalability and adaptability with high level abstraction for modelling of complex software systems. The approach employed the use of self-organising agents for automating and coordinating resource management with promising highly dynamic behaviours. In addition, the variation in type of the resources is indispensable in heterogeneous systems like the cloud. Also request matching with service advertised was only the focus in the brokering process and not in the final service deployed for use (Kang & Sim, 2010). Though MAS approach was used in this work, but the issue of trust was not considered among the cloud entities while cloud services were been discovered.

Han and Sim (2010) worked on a cloud service discovery system (CSDS) that supported the cloud users in finding a cloud service over the Internet. The prototype of the CSDS was made up of a search engine and three different agents namely; query processing agent, filtering agent, and cloud service reasoning agent (CSRA). There were two components: (1) CSDS which helps to find the best cloud service on behalf of users and (2) a cloud ontology which comprises taxonomy of concepts of different cloud services to consult with the CSRA. In addition, there was a user interface which allows the user to enter queries containing a service name and requirements considered by their preferences. From the empirical results, the CSDS with the cloud ontology performed better than the CSDS without the cloud ontology. By contacting a cloud ontology to reason about the relations among cloud services, the CSDS was more successful in locating cloud services and more likely to discover cloud services that meet users' requirements (Han & Sim, 2010). The functionalities of query processing, filtering and rating were partially implemented but trustworthiness of the cloud service was not a concern in the process.

Gopinadh and Saravanan (2013) presented an agent-based cloud system for cloud service reservation. The work entails an agent-based cloud service discovery approach that consults ontology to retrieve information about cloud services and implements price and timeslot negotiation (PTN) mechanism for cloud service reservation. It was designed for the cloud service that best suit their price and timeslot. The important contributions of the work includes: developing an agent-based search engine for cloud service discovery and effective adaptation of agent-based PTN mechanisms for cloud service negotiation. CSRA which enables the CSDS was used to reason about the relations of cloud services, rate the search results and design and construction of cloud ontology. Cloud agents in PTN are designed to concurrently make multiple proposals in a negotiation round that generate aggregated utility, differing only in terms of individual price and time-slot utilities (Gopinadh & Saravanan, 2013). Trustworthiness of cloud services was not considered in the effort.

Sim (2009) proposed an agent-based testbed for bolstering the discovery of cloud resources and service level agreement (SLA) negotiation. The testbed for simulating cloud resource management comprises (i) a set of resource consumers, (ii) a set of physical machines, (iii) a set of cloud resources (virtualized machines), and (iv) a set of middleware comprising of provider agents and consumer agents acting as intermediaries between resource providers and consumers, respectively, and a set of broker agents that connects resource requests from consumers to advertisements from providers. Based on the connection algorithm used for connecting buyers and sellers in e-commerce, this work transformed the cloud resource discovery process into a process of matching consumers' resource requests to resources of providers. This was achieved by a four-stage resource discovery process of selection, evaluation, filtering, and recommendation, a set of broker agents matches consumers' requests to advertisements from providers. After the matching of requests to resources, consumer and provider agents then negotiate for mutually acceptable resource time slots (Sim, 2009). It is noted that the ideas, design, and development of an agent-based cloud resource management testbed were still evolving. Meanwhile trust issues were not addressed.

Talia (2011) proposed an approach to develop a multi agent based system that makes use of ontology relationships to find the similarities between the user queries and the available resources from various service providers. The system entails the use of agent approach as agents can be autonomous and mobile for easy communication with each other over a distributed environment. The proposed approach demonstrates a typical prototype of a cloud service discovery system that comprises three different agents—requestor agent, mapper agent, and search agent—and a search engine. The agents are developed using Java agent development framework. The search engine used a concept based search using similarity reasoning from the ontology relationships of the concepts in the cloud ontology. As multiple agents are used, the average waiting time of a search was reduced also; the overall waiting time was also reduced (Talia, 2011). The use of ontology has increased the chances of finding an exact match for the services requested. This work clearly demonstrated that MAS can aid effective cloud discovery in a distributed environment.

Rajendran and Swamynathan (2014) proposed the use of cloud ontology technique to make discovery of cloud services more efficient. It utilized a multi-broker agent based discovery system that consults ontology while attempting to retrieve information about services. A cloud service discovery using broker agents (CSDBA) system was developed. The approach combines the advantages of the agent-based system and the semantic matching mechanism using cloud ontology. The CSDBA, an agent-based service discovery system, interface with the cloud ontology for reasoning about the relations of cloud services. It is designed to support consumers in finding appropriate cloud services. CSDBA was more successful and can be helpful tool to find cloud services that meet requirements of user than traditional approach. In the future, it was indicated that cloud services brokering mechanism would be extended (Rajendran & Swamynathan, 2014). The future agenda included implementing efficient methodology for cloud crawling and rate the services based on QoS (Rajendran & Swamynathan, 2014). Table 1 contained the summary of the studies carried out by the cited authors.

Parhi, Pattanayak, and Patra (2015) proposed a framework for service description and discovery using ontology based on multi-agent cloud environment. Because of the non-standardized specification terminology used by different cloud providers, the traditional search engines such as Google, MSN, bing, etc. are not able to perform the service discovery in an effective manner. There arises the necessity of a reasoning mechanism for service discovery that is capable of resolving the similarities among variety of services. In Parhi et al. (2015), introduced a semantic-based service description and discovery framework using multi-agent approach, where the cloud service descriptions that are automated based on shared ontology, contribute to optimal discovery of appropriate services as requested by consumers was addressed. The reason mechanism failed to incorporate the trustworthiness of cloud services as a factor for reasoning.

Moses, Ojo, and Zuva (2016) presented an efficient and scalable approach for semantic cloud service discovery in a P2P (peer-to-peer) cloud environment. The approach was based on learning automata (LA) and ant colony optimization (ACO). The ability of ACO to adapt to changes in real time makes it a better choice in dynamic environments such as cloud. The approach showed an improved performance against the some existing P2P service discovery approaches. Meanwhile, attention was not given to the trustworthiness of cloud services in the work.

Liu, Xing, Che, and Bao (2015) proposed a centralized robust service discovery approach with local and global service discovery capabilities which can be applied to agent-based cloud computing environment. The system consists of four kinds of agents: resource agents (RA), service provider agents (SPA), broker agents (BA) and consumer agents (CA). The proposed system has an overlay network above the physical network in order to perform global searching. It adopted indexes to be able to handle intermittent and volatile network environment like cloud computing environment. The approach generates minimize traffic during the service publication; service discovery process because of each index handles its own jurisdiction. But the system lacked a structured network topology on the virtual network in order to achieve global service query with higher efficiency and there was no

Table 1. Summary of studies by cited authors

Author(s)	Concern	Contribution	Remark
Kang and Sim (2010)	Lack of tools for supporting cloud service discovery	The paper showed how self-organizing agents were employed for automating and coordinating resource management with promising highly dynamic behaviours	Though MAS approach was used in this work, but the issue of trust was not considered among the cloud entities while cloud services were been discovered
		It was planned that the cloud services brokering mechanism will be extended in the future for more intelligent discovery	
Han and Sim (2010)	There was no discovery mechanism for searching different kinds of clouds	The paper revealed that an agent-based discovery system that consults an ontology when retrieving information about cloud services is a possibility	The functionalities of query processing, filtering and rating were partially implemented but trustworthiness of the cloud service was not a concern in the process
Gopinadh and Saravanan (2013)	Finding cloud service that best suits price and time slot of users	The paper showed that using agent based approach aided effective service negotiation	Cloud agents in PTNs are designed to concurrently make multiple proposals in a negotiation round that generate aggregated utility, differing only in terms of individual price and time-slot utilities. Trustworthiness of cloud services was not considered in the negotiation
Sim (2009)	The concern was on the design and development of software agents for bolstering cloud service discovery, service negotiation, and service composition	The significance of this work was the introduction of an agent-based paradigm for constructing software tools and testbeds for cloud resource management	It is noted that the ideas, design, and development of an agent-based cloud resource management testbed were still in infancy and trust issues were not addressed in the processes
Talia (2011)	Discussion on the potential benefits of integrating multi-agent technology with cloud computing	This paper reported that multiple agents can exchange information and cooperate to perform complex tasks where interaction, intelligence, adaptation and dynamicity are key issues to be handled	This work clearly demonstrated that MAS can aid effective cloud service discovery in a distributed environment and adding trust concern to the system will aid intelligent cloud service discovery
Rajendran and Swamynathan (2014)	Finding appropriate cloud services	The work utilized a multi-broker agent based discovery system that consults ontology while attempting to retrieve information about services	The retrieved information about services did not include trustworthiness of the cloud services
Parhi et al. (2015)	There were no standardized specifications defined for a cloud service provider. Different cloud service providers use different vocabulary to specify similar operations	The paper proposed a semantic-based service description and discovery framework using multi-agent approach	The reason mechanism failed to incorporate the trustworthiness of cloud services as a factor for reasoning

(Continued)

Table 1. (Continued)

Author(s)	Concern	Contribution	Remark
Moses et al. (2016)	Centralized and hierarchical approaches to service discovery have exhibited bottlenecks as network load increases and limitation in scalability	This paper presented an efficient and scalable approach for semantic cloud service discovery in a P2P cloud environment. The approach was based on LA and ACO	The ability of ACO to adapt to changes in real time makes it a better choice in dynamic environments such as cloud. Incorporating trust concern of cloud services will make it highly intelligent
Liu et al. (2015)	Agents utilized the obsolete acquaintance network and service capability tables (SCTs) to record and schedule agents for service discovery	This paper presented a centralized robust service discovery approach with local and global service discovery capabilities	The approach can be applied to agent-based cloud computing environment. But it may be liable to single point of failure in a dynamic cloud environment and its scalability cannot be assured

provision for accommodating trustworthiness of cloud services. This centralized approach is liable to single point of failure in a dynamic cloud environment and its scalability cannot be assured.

The review of the related works showed that attempts have been made in using MAS to address the cloud service discovery process but the issue trustworthiness of cloud service while discovering cloud services have been neglected and addressing the issue will aid the cloud user to discovery trustworthy cloud services. This work is leveraging on the capabilities of the MAS to extend the discovery system to ensure that trustworthy cloud services are discovered. Extending the multi-agent paradigm for trustworthy cloud service discovery will enable efficient cloud service discovery and improve the process.

2. Methodology

The conceptual design for a trustworthy cloud service discovery system was based on Functional Analysis which is a crucial tool for the design of process to explore new concepts and to define their architectures in system engineering (Viola, Corpino, Fioriti, & Stesina, 2012). It was used to define the functional requirements for a trustworthy cloud service discovery system. It ensured that all necessary components are included and that no unwanted components were included. The relationships between the system components were also specified. The main step of Functional Analysis was made up of the functional tree, the functions/devices matrix and the product tree. Through the functional tree and particularly through the identification of the basic functions, the functional requirements of the

A Multi-Agent System Approach for Trustworthy Cloud Service Discovery

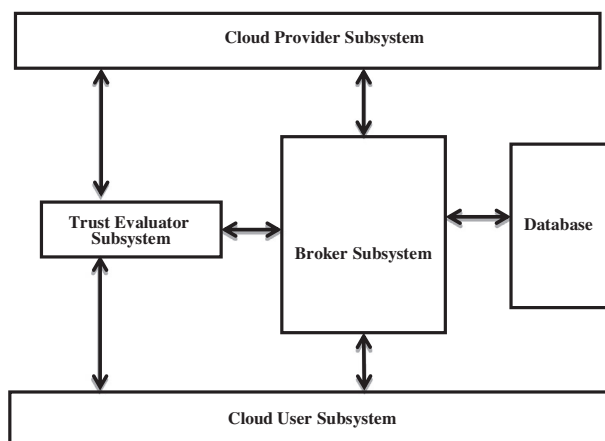


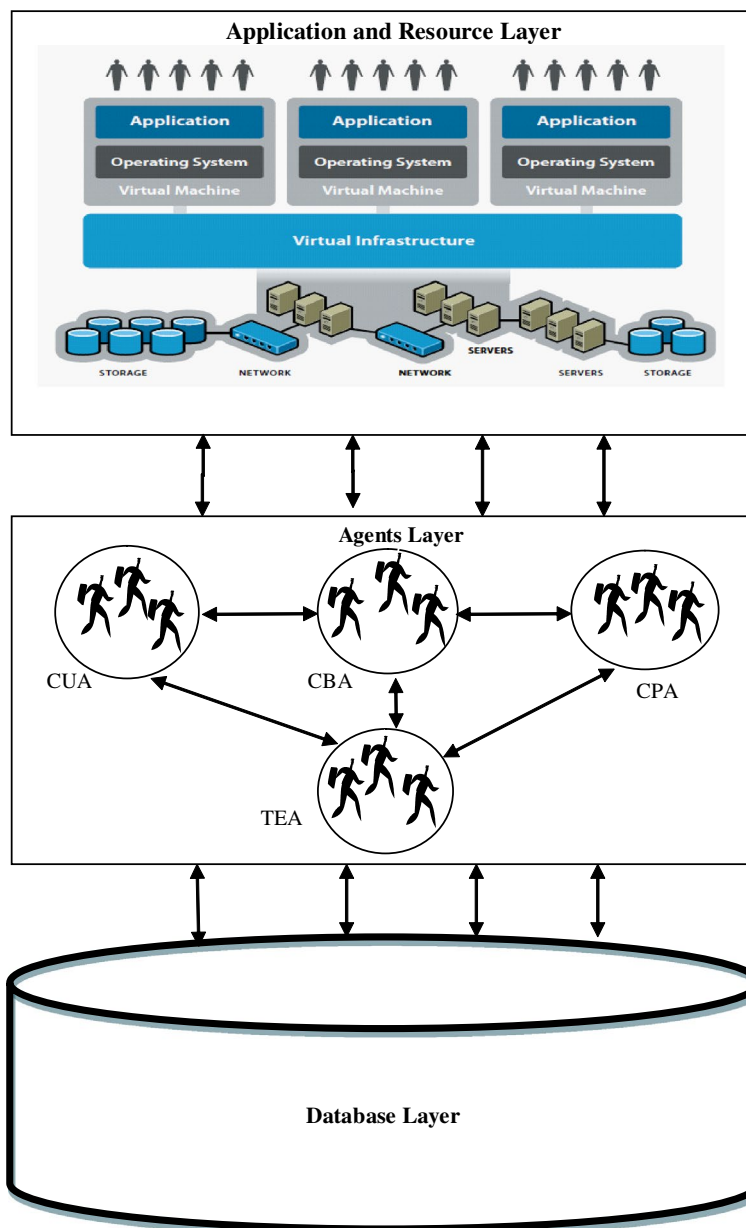
Figure 1. Cloud service discovery functional block diagram.

system were defined, and through the product tree the building blocks of the system were determined, thus providing the functional architecture of the system. Starting from the top level system requirements of trustworthy cloud service discovery using multi-agent approach the functional analysis led to the proper identification of the building blocks, which comprises the proposed system, and their interconnection to build up the functional architecture of the proposed system. The functional analysis method produced the following fundamental subsystems as requirements for achieving efficient cloud service discovery: cloud user subsystem, cloud provider subsystem, trust evaluator subsystem, broker subsystem and database as depicted in Figure 1. Achieving a trustworthy cloud service discovery system required that the various subsystems must collaborate, cooperate and interact together.

2.1. System architecture

The system architecture was constructed from the basic components of the functional block for cloud service discovery and the architecture is a distributed system. The architecture has the following three layers as shown in Figure 2.

Figure 2. System architecture.



- (i) Application and resource layer.
- (ii) Agent layer: classes of agents.
- (iii) Database layer: holding information status of applications/resources/services.

2.1.1. Application and resource layer entails the cloud service models

The layer is the collection all cloud services available providing the interface for connecting to the cloud resources and services. It entailed the cloud service models. A cloud service is any resource that is provided over the internet. The most common cloud service resources are software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. PaaS refers to the delivery of operating systems and associated services over the internet without downloads or installation. IaaS involves outsourcing the equipment used to support operations, including storage, hardware, servers and networking components, all of which are made accessible over a network.

2.1.2. Agent layer

The system construction was based on the multi-agent approach for achieving effective and reliable discovery. The agents are used as basic components for achieving intelligence in the system, making it more adaptive, flexible, and autonomic in its operation. This opportunity is employed for implementing efficient MAS and, from a more general point of view, for advancing the way to design and implement a service discovery system that is distributive and trust enabled in cloud environment.

In this layer the various agents that are involved in trustworthy cloud service discovery are situated. The agents are; cloud user agents (CUA), trust evaluator agents (TEA), cloud provider agents (CPA) and cloud broker agents (CBA).

These autonomous agents with their various functions establish appropriate link among themselves for efficient cloud service discovery. In doing this, a peer-to-peer approach is adopted among the various agents for the discovery. The participating agents, referred to as a peer, functions as a client with a layer of server functionality in their operations. This allows the peer to act both as a client as well as a server within the context of a given cloud service discovery interaction. Interaction may spread over all the peers, and peers can communicate directly with each other, and hence, are constantly aware of each other, or they may communicate indirectly through other peers for effective discovery. In the latter case, the arrangement or the organization of peers may either have a regular structure, or be unstructured. An indirect unstructured approach removes the need for any form of management to enforce a specific network configuration; however, the emphasis and the workload is shifted to the cloud service discovery effort as changes to the cloud environment can happen without the awareness of all the peers.

In this layer, cloud service discovery occurs by means of and through communication among the agent entities. Given this approximation, discovery becomes an inherently social activity, instead of solitary or monolithic, leading to efficient forms of discovering trustworthy cloud services. The cloud discovery is no longer monolithic but rather distributed by agent entities. These agents form coalitions among themselves to attain their current goals of cloud service discovery.

2.1.2.1. Functions of each agent. Having identified the various agents needed for achieving a trustworthy cloud service discovery, the roles of each of the agents are as follows:

2.1.2.1.1. User agent. The user agent provides interface for the users to make service request. It sends request to broker agents using message passing and shows the results to user through a user interface. It interacts with provider agent through the broker agents for service usage. The user agent interacts with trust evaluator agent to monitor the performance value of the provider in terms of service type, service integrity and service accessibility.

Algorithm

```

Start
Let N = total number of users request within a specified time
While (User's request  $\neq \emptyset$ ) Do
  For i = 1, i++, N
    User_Agent  $\leftarrow$  User's request:  $UR_i$ 
    Broker_Agent for  $UR_i$  discovery  $\leftarrow$  User_Agent
    Wait for Broker_Agent
    User_Agent  $\leftarrow$  Broker_Agent returns Result of  $UR_i$  discovery
    User Interface  $\leftarrow$  User_Agent Result for use
    If ( $UR_i$  is used) Then
      User_Agent interacts with Trust_Evaluator_Agent on  $UR_i$ 
    End if
  Next i
End do
End

```

2.1.2.1.2. *Broker agent.* The broker agent analyses the user's request to determine its type. It then locates a possible match for the identified service request and returns the list of such services to the user agent. It sends user-provider's interaction information to the database. It also interacts with database for other broker agent recommendation. The broker agent receives trust value updates from trust evaluator agent.

Algorithm for

```

Start
If (Broker_Agent received User_Agent Message) Then
  Broker_Agent Analyse  $UR_i$ 
  Search the database for a match
  If ( $UR_i$  match is found) then
    Broker_Agent  $\leftarrow UR_{imatch}$ 
  Else
    Other_Broker_Agents  $\leftarrow UR_i$ 
    If ( $UR_i$  match is found by Other_Broker_Agents) Then
      Broker_Agent  $\leftarrow UR_{imatch}$ 
    Else
      Broker_Agent  $\leftarrow UR_{imiss}$ 
    End if
  End if
  User_Agent  $\leftarrow$  Broker_Agent
End if
End

```

Algorithm for

```

Start
While (a typical service is in use) Do
    Broker_Agent Interact with Trust_Evaluator_Agent
    Broker_Agent ← Trust_Evaluator_Agent (trust value updates)
    Database ← Broker_Agent (Stores user agent—provider agent interaction information)
End Do
End
    
```

2.1.2.1.3. *Trust evaluator agent*. It keeps track of the provider service provider's behaviour contributing to a proper operation of the cloud in terms of the following factors: service type, service integrity, and service accessibility by interacting with user and provider agents. It determines the trustworthiness of cloud service provider. It maintains a table for keeping the performance of the provider and updates it when new performance value is available.

```

Start
While (User_Agent Interacts with Provider_Agent) Do
    Trust_Evaluator_Agent: Keeps track of provider's performance
    Trust_Evaluator_Agent: Determines the trustworthiness of providers
    Broker_Agent ← Trust_Evaluator_Agent (trust value updates)
    Trust_Evaluator_Agent: Cache the providers' performance
End Do
End
    
```

2.1.2.1.4. *Provider agent*. Provider agent is distributed in the cloud, one at each cloud service provider site. It sends advertisement, removal and updates information of cloud services to the broker agent. It provides services for usage by the user agent.

Algorithm

```

Start
While (a service is being offered by service provider) Do
    Provider_Agent: Create service instance
    Broker_Agent ← Provider_Agent (Broadcast service availability)
    Broker_Agent ← Provider_Agent (Broadcast service expire information)
    Broker_Agent ← Provider_Agent (Broadcast service update)
    User_Agent ← Provider_Agent (Provides service)
End Do
End
    
```

The corresponding class diagram and sequence diagram of the services rendered by the agents are shown in Figures 3 and 4 respectively.

2.1.3. Database layer

The database layer stores the service usage, agents interaction information. It Stores service information and trust value rating for providers. It also keeps the brokers interaction information. In a typical cloud service discovery system, users submit their request to the broker. The broker consults the registry or directory (database) to find a possible match for the user's request. The provider publishes its available services into the service registry or directory (database) and also updates the status of the published service.

Figure 3. Class diagram of the agent operations.

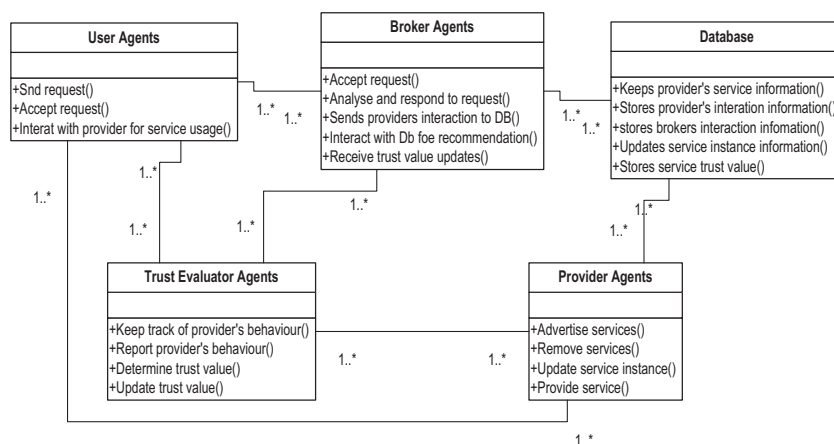
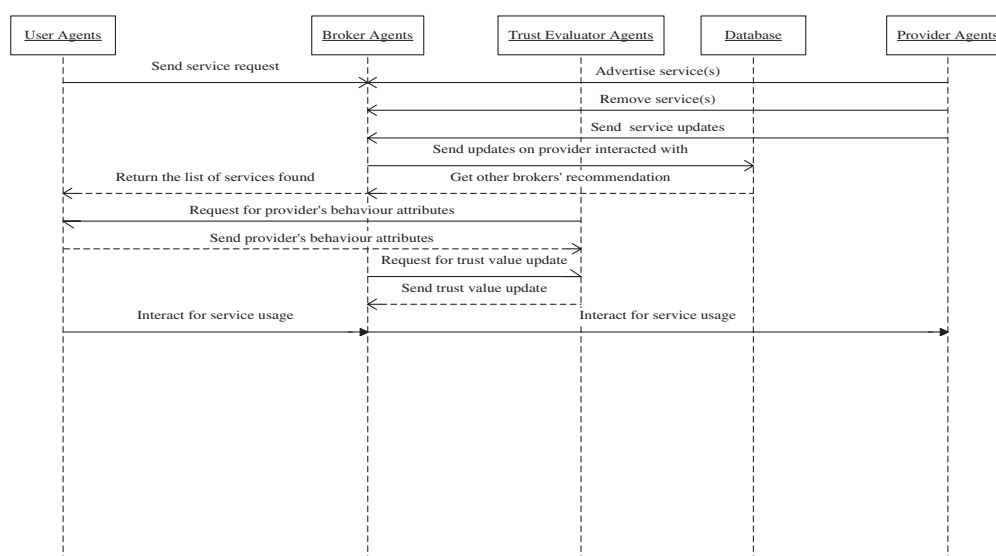


Figure 4. Sequence diagram of the agents operations.



3. Simulation, results and discussion

Simulation experiments were conducted to investigate the effect of extending MAS approach to achieve a trustworthy cloud service discovery using response time and scalability as the evaluation parameter. The response time in this regard refers to time lag between the instant of service request by the users to the instant of having the result of the service request. This depends on round trip time (RTT) and the users load (UL). Scalability is a metric that observes how the system can be easily altered to accommodate changes in the number of users, resources and computing entities attached to it. Scalability depends on processing capability of the provider expressed as processing time (PT) and users load (UL).

The experiment examined the cloud service discovery transactions carried out without involving trust factor against the same kind of cloud service discovery transaction involving trust factor in order to investigate their difference. CloudAnalyst (Wickremasinghe, Calheiros, & Buyya, 2010) useful to model and analyze large scale cloud computing development was used for the simulation. Considering the description and behaviour of throttled load balancing policy of the CloudAnalyst, the integration of trust capability in cloud environment was achieved using this policy while the round-robin load balancing policy, considering its behaviour was used for the cloud transaction without trust integration.

Table 2. Simulation parameter settings for response time with varied number of user agents' growth factor and varied number of provider agents

User agents	Provider agent	User growth factor	Request growth factor	Execution instruction per length
5	1	100	10	100
10	2	1,000	10	100
15	3	10,000	10	100
20	4	100,000	10	100
25	5	1,000,000	10	100
30	6	10,000,000	10	100

The first experiment simulated cloud service discovery transaction using CloudAnalyst. In this experiment, there were 5, 10, 15, 20, 25 and 30 groups of users agents with a varied number cloud provider agents and the user agent growth factor varied from 100 to 10,000,000. These groups of user agents were generated randomly using Poisson distribution by varying the number of users in a realistic manner. The detailed parameter settings for the experiment were shown in Table 2. In this experiment, the number of user agents' growth factor were varied while the number of the provider agent was varied in order to observe the behaviour of the existing system without trust integration and proposed system with trust integration in terms of the response time.

Table 3 shows the results obtained from the simulation. The in graph Figure 5 shows the plot of the round trip time against the number of user agents to verify the behaviour of the proposed and existing systems in terms of response time. From the results, the proposed system ensures lower and gracefully stable response time than the existing system with higher response time. This is evident from the percentage difference of the systems' response time as shown in Table 3. It was observed that integration of trust into the process of cloud service discovery was responsible for the lower response time of the proposed system.

This second experiment simulated cloud service discovery transaction using CloudAnalyst. In this experiment there were 5, 10, 15, 20, 25 and 30 groups of user agents with varied number cloud provider agents having a capacity of 5 virtual machines each. The user agents' growth factor was varied from 100 to 10,000,000. The detailed parameter settings for the experiment were as shown in Table 4.

In this experiment, the number of user agent growth factor were varied while the number of the provider agent was varied in order to observe the behaviour of the existing system without trust integration and proposed system with trust integration in terms of the scalability.

Table 3. Response time with varied number of user agents' growth factor and varied number of provider agents

User agents growth factor	Round trip time (ms) (Existing system)	Round trip time (ms) (Proposed system)	Percentage difference (%)
100	302.12	300.81	-0.434
1,000	302.42	300.19	-0.737
10,000	303.10	301.15	-0.643
100,000	303.10	301.08	-0.666
1,000,000	303.71	301.13	-0.849
10,000,000	303.49	301.40	-0.689

Figure 5. Response time with varied number of user agents' growth factor and varied number of provider agents.

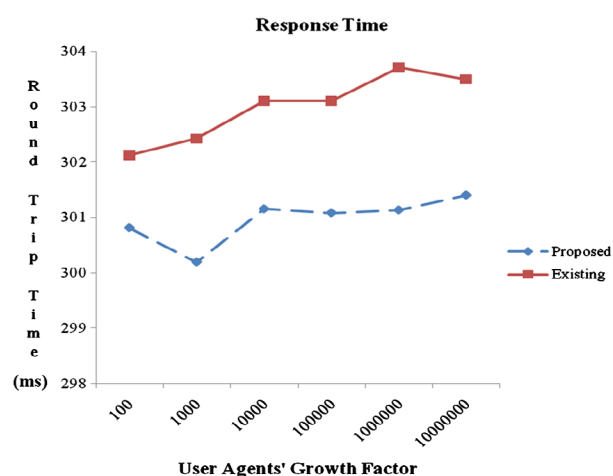


Table 4. Simulation parameter settings scalability for varied number of user agents' growth factor and varied number of provider agents

User agents	Provider agents	Number of virtual machines	User growth factor	Request growth factor	Execution instruction per length
5	1	5	100	10	100
10	2	5	1,000	10	100
15	3	5	10,000	10	100
20	4	5	100,000	10	100
25	5	5	1,000,000	10	100
30	6	5	10,000,000	10	100

Table 5 shows the results obtained from the simulation. The graph in Figure 6 shows the plot of the provider agent processing time against the number of user agents' growth factor to verify the behaviour of the proposed and existing systems in terms of scalability. From the results, the proposed system scales the load of user agents fairly better than the existing system as the number of user agents increases and the processing capability of each of the provider agents remain fixed. This is evident from the percentage difference of the systems' scalability metric as shown in Table 5. It was observed that integration of trust into the process of cloud service discovery brought about this achievement.

3.1. Simulation results validation

From the simulation results, the proposed system with trust performed better than the existing one without trust. In order to validate the simulation results, paired difference *t*-test was performed on the percentage difference of the response time with six group size in the simulation settings for the proposed and the existing systems. The null hypothesis states that there is no significant difference between the proposed and existing system while the alternative hypothesis states that there is significant difference between them.

The calculated *t* values for the response time in the experimental setting was -70.43 as shown in Table 6. The degrees of freedom (*df*) for the total group size of six is five. Entering a *t* table with 5 *df*, at 95% confidence interval, the table *t* value is 2.02 (one-tailed, a significance level (α) of 0.05). The absolute calculated *t* value is higher than the tabled *t* value of 2.02 showing that the proposed system with lower response time is significantly different ($p = 0.05$) from the existing system as a result of trust integration in it.

Table 5. Scalability with varied number of user agents' growth factor and varied number of provider agents

User agents growth factor	Provider agent processing time (ms) (Existing system)	Provider agent processing time (ms) (Proposed system)	Percentage difference (%)
100	1.91	0.59	-69.110
1,000	2.60	0.38	-85.385
10,000	2.87	0.92	-67.944
100,000	2.98	0.95	-68.121
1,000,000	2.95	0.37	-87.458
10,000,000	3.06	0.97	-68.301

Figure 6. Scalability with varied number of user agents' growth factor and varied number of provider agents.

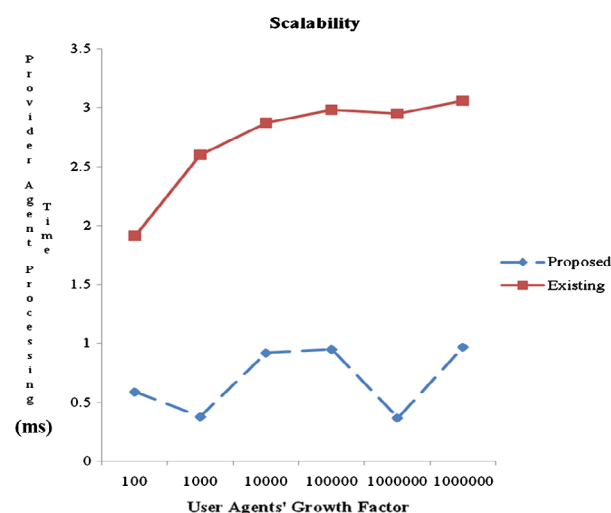


Table 6. Calculated t values for the existing system and proposed system

Performance metrics	Calculated t value
Response time	-70.43
Scalability	-3.09

Performing a similar t -test on the percentage difference of the scalability metric with six group size in the simulation settings for the proposed and the existing systems, resulted to the calculated t values of -3.09 respectively as shown in Table 5. Entering a t table with 5 df , at 95% confidence level, the table t value is 2.02 (one-tailed, a significance level (α) of 0.05). The absolute calculated t value 3.09 is higher than the table t value showing that the proposed system is more scalable and significantly different ($p = 0.05$) from the existing system as a result trust integration in it.

4. Conclusion

In this work, we have proposed a distributive approach using MAS for trustworthy cloud service discovery. Simulation experiments were carried out and the results showed that extending the MAS approach by integrating trust into the process of cloud service discovery improves the quality of service in terms of response time and scalability. This distinguished the proposed system with trust integration in the process of cloud service discovery from the existing system without trust integration. The research indicated that the use of MAS within cloud environment can aid intelligent cloud service discovery and support the complexity of the cloud environment. A further comparative

analysis of MAS approach for cloud service discovery to monolithic approach showed that it is highly efficient and highly flexible for trustworthy cloud service discovery. MAS provide efficient solution in environment where resources are spatially and potentially distributed whereas a monolithic approach is highly inefficient and not reliable for a distributed environment such as cloud. The proposed distributive approach using MAS for trustworthy cloud service discovery is not susceptible to single point of failure. Future work will consider optimizing the MAS distributive approach to cloud service discovery.

Funding

The authors received no direct funding for this research.

Author details

A.O. Akinwunmi¹

E-mails: akinwale.akinwunmi@bowenuniversity.edu.ng,
walebunmi@gmail.com, ppp_wale@yahoo.com

E.A. Olajubu²

E-mail: emmaolajubu@oauife.edu.ng

G.A. Aderounmu²

E-mail: gaderoun@oauife.edu.ng

¹ Computer Science & Information Technology Department,
Bowen University, Iwo, Nigeria.

² Computer Science & Engineering Department, Obafemi
Awolowo University, Ile-Ife, Nigeria.

Citation information

Cite this article as: A multi-agent system approach for trustworthy cloud service discovery, A.O. Akinwunmi, E.A. Olajubu & G.A. Aderounmu, *Cogent Engineering* (2016), 3: 1256084.

Cover image

Source: Authors.

References

- Akinwunmi, A. O., Olajubu, E. A., & Aderounmu, G. A. (2014). A trustworthy model for reliable cloud service discovery. *International Journal of Computer Applications*, 87, 23–30. doi:10.5120/15293-3962
- Buyya, R., Yeo, C. S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC)*, Dalian.
- Drashansky, T., Houstis, E. N., Ramakrishnan, N., & Rice, J. R. (1999). Networked agents for scientific computing. *Communications of the ACM*, 42, 48–53. <http://dx.doi.org/10.1145/295685.295699>
- Durfee, E. H., & Montgomery, T. A. (1989). MICE: A flexible testbed for intelligent coordination experiments. In *Proceedings of the 1989 Distributed Artificial Intelligence Workshop, Rosario* (pp. 25–40). Eastsound, Washington, DC.
- Gopinadh, R., & Saravanan, K. (2013). Cloud service reservation using PTN mechanism in ontology enhanced agent-based system. *International Journal of Engineering Trends and Technology (IJETT)*, 4, 791–798.
- Han, T., & Sim, K. M. (2010). An ontology-enhanced cloud service discovery system. In *Proceedings of International Multiconference of Engineers and Computer Scientists (IMEC)* (Vol 1, pp. 644–649), Hong Kong.
- Honavar, V., Miller, L., & Wong, J. (1998). Distributed Knowledge Networks. In *Proceedings of the IEEE Information Technology Conference* (pp. 87–90). Syracuse, NY: IEEE Press.
- Kang, J., & Sim, K. M. (2010). Towards agents and ontology for cloud service discovery. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 483–490), Beijing.
- Khosla, R., & Dillon, T. (1997, June 9–12). Intelligent hybrid multi-agent architecture for engineering complex systems. In *1997 Proceedings of the IEEE International Conference on Neural Networks* (pp. 2449–2454), Houston, TX.
- Kousiouris, G., Kyriazis, D., Varvarigou, T., Oliveros, E., & Mandic, P. (2012). Taxonomy and state of the art of service discovery mechanisms and their relation to the cloud computing stack. In D. Kyriazis, T. Varvarigou, & K. Konstanteli (Eds.), *Achieving real-time in distributed computing: From grids to clouds* (pp. 75–93). Hershey, PA: Information Science Reference. <http://dx.doi.org/10.4018/978-1-60960-827-9>
- Liu, D., Xing, W., Che, X., & Bao, P. (2015). A centralized service discovery approach for agent-based cloud computing system. *The Open Cybernetics and Systemics Journal*, 9, 526–535.
- Mei, L., Chan, W. K., & Tse, T. H. (2008). A tale of clouds: Paradigm comparisons and some thoughts on research issues. In *Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference (APSCC)*. Los Alamitos, CA: IEEE Computer Society Press.
- Moses, O., Ojo, S., & Zuva, T. (2016). An adaptive multi agent service discovery for peer to peer cloud services. In *Emerging Trends and Advanced Technologies for Computational Intelligence* (pp. 147–163). Basel: Springer International Publishing.
- Parhi, M., Pattanayak, B. K., & Patra, M. R. (2015). A multi-agent-based framework for cloud service description and discovery using ontology. In *Intelligent Computing, Communication and Devices* (pp. 337–348). New Delhi: Springer India.
- Rajendran, V., & Swamynathan, S. (2014). A novel approach for semantic service discovery in cloud using broker agents. In *International Conference on Advances in Computing, Communication and Information Science (ACCIS-'14)* (pp. 242–250), Kollam. ISBN: 9789351072478. doi:10.13140/2.1.4022.6085
- Sim, K. M. (2009). Agent-based cloud commerce. In *Proceedings of the 2009 IEEE/IEEM* (pp. 717–721). Hong Kong.
- Talia, D. (2011). Cloud computing and software agents: Towards cloud intelligent services. In G. Fortino, A. Garro, L. Palopoli, W. Russo, & G. Spezzano (Eds.), *WOA* (Vol. 741, pp. 2–6). Rende: CEUR-WS.org.
- Vaquero, L. M., Roderio-Merino, L., Caceres, J., & Lindner, M. (2009). A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39, 50–55.
- Viola, N., Corpino, S., Fioriti, M., & Stesina, F. (2012). Functional analysis in systems engineering: Methodology and applications. In B. Cogan (Ed.), *Systems engineering - practice and theory* (pp. 71–96). Rijeka: InTech.
- Wickremasinghe, B., Calheiros, R. N., & Buyya, R. (2010). *CloudAnalyst: A cloudsims-based visual modeller for analysing cloud computing environments and applications*. Retrieved from <http://www.cloudbus.org/cloudsim/>
- Wooldridge, M. (2009). *An introduction to multiagent systems* (2nd ed.). Chichester: Wiley.



© 2016 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Cogent Engineering (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com

