Agent Based firewall management Framework in cloud computing environment using SOAP MTP

Noora Shihab Ahmed
Diyala University / Faculty of Education of Pure Science / Computer Science Department
assel_white1982@yahoo.com

Abstract

This paper is presenting framework for Agent based fireworks integration over the emerging cloud computing environment, where Agents will be able to exploit knowledge gained by other Agents who live anywhere.

The new model presented here introduces the idea of building global platform for Agents, and this global platform will represent the gateway for all Agents around the world. SOAP MTP driver has been used to grant Agents the ability to talk to web services and then deliver messages over the internet. Proxy Agents are managed by web services to provide environmental functionalities analogous to JADE (Java Agent Development environment). The presented model grants Agents the ability to globally develop knowledge and share experience.

Keywords: firewall, Intelligent Agent, Web service, UDDI, XML-Schema, JADE

اطار عمل لادارة جدران النار المعتمدة على الايجنت في بيئة الحوسبة السحابية باستخدام SOAP MTP

نورا شهاب احمد
جامعة ديالى/ كلية التربية للعلوم الصرفة / قسم علوم الحاسبات
assel_white1982@yahoo.com

Agent Based firewall management Framework in cloud computing environment using SOAP MTP

Noora Shihab Ahmed

الخلاصة

ان هذا البحث يقدم اطار عمل لتكامل جدرات النار المعتمدة على الايجنت في بيئة الحوسبة السحابية الناشئة حيث ان الايجنت سوف يستخدم ويستعمل المعرفة المستحصلة من قبل ايجنت اخر يعيش في مكان اخر. ان النموذج المقدم هنا يطرح فكرة بناء قاعدة تشغيل وانطلاق للايجنت والتي تمثل بوابات عالمية لكل الايجنت حول العالم. SOAP MTP تم استخدامها لمنح الايجنت القدرة على التخاطب مع خدمات الويب ومن ثم ارسال الرسائل عبر الانترنيت. مجموعة الايجنت الوكيل تم ادارتها من قبل خدمات الويب لتقديم خدمات بيئية مشابهة لما يقدمه JADE وفي هذا النموذج يصبح لدى مجموعة الايجنت القدرة على تطوير المعرفة ومشاركة الخبرة بشكل عالمي غير محدود.

الكلمات المفتاحية: - جدار الناري , العميل الذكي ,خدمة الويب , UDDI, مخطط الXML, JADE.

Introduction

Cloud computing is the new generation of platforms and environments through which applications can be developed using remote resources. These resources include software, platforms and infrastructures, and this reduces the level of professionalism needed to develop such applications. In cloud computing all resources are accessed through HTTP protocol and its embedded protocols such as SOAP, JAX-WS and RMI [1, 2].

Cloud computing firewall is the next generation of network security products due to the differences in platform, architecture and Software deployment strategy, Cloud computing environment is where the outsourcing is the key factor[1].

HTTP protocol has a significant feature that other protocol does not have; this is the ability to come through the firewalls on the permission of the administrators. HTTP is the protocol of the internet (https also can be used to access the internet) through which Web servers can be queried for specific resources, and these resources are identified by URI of URL [1].

SOAP (Simple Object Access Protocol) is an XML-Based protocol embedded in HTTP, and this protocol has been designed to carry out specific task which is Web-service integration[1].

Web services are a collection of Web functionalities designed to support interoperable machine-to-machine interaction over the web. It is ranging from simple task functionality (find factorial of a number) to integrated one (encrypt data). It is more accurate to describe web service as software component rather than software functionality [1, 2].

1. Security enhancement using software Agent

Software Agent has been deployed in many network security schemes as a tool to detect abnormal behavior in the network. In these security schemes two categories of agents were used: first category is the server Agent which is considered the core agent, and the second category is the client Agent where in each workstation there is a running instance of this category [3].

Server agent hosts manager and expert software system, messages are sent from client agent to core agent; these messages describe client agents' perceiving of the events occurred within their system. Actions taken only by core agent and clients agents poll their directions from the core agent [3,4,5].

In Autonomous Agents clients agents and core agent are capable of monitoring the environment and communicate with core agent to identify possible threats; the difference here is client agent is more capable to take decisions in the environment, for example can investigate further the network traffic to get more reliable assessments.

2. Autonomic Networks Management and Configuration

Autonomic Networking is an extension to 'Autonomic Computing' which is an initiative started by IBM in 2001. The initiation was targeting provisioning network management services to present self-managing networks due to the rapid growing complexity of the internet.

Autonomic computing is an emerging paradigm which promotes the management level of large scale computing system to be self-manageability, as it is presented in figure (1) [6].

Autonomic is a type of computing model in which the system is self-healing, self-configured, self-protected and self-managed; these characteristics are accomplished through using artificial intelligence concepts. One significant feature of autonomic model is its ability to adapt to changes occurred in the environment by interacting with neighboring systems [6].

The essential element in autonomic computing environment is the autonomic element

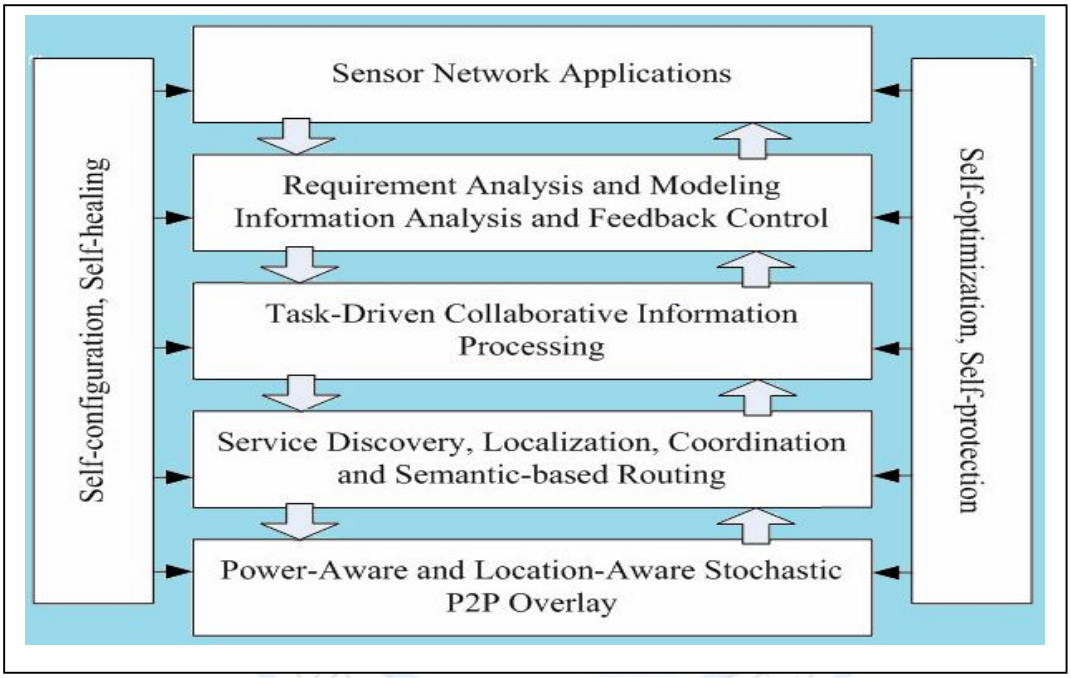

Figure 1 : Protocol Stack for Autonomic Network and Configuration

3. Software Agent and Enterprise Interoperable Applications

The SOAP MTP add-on is a pluggable driver for sending and receiving SOAP messages and translating them to/from internal agent message format. Each agent platform uses the SOAP MTP add-on configured with a virtual endpoint address, which is mapped to the agent platform address in the Gateway component. The virtual endpoint address is also advertised in registries in directories outside the organizational domain, so the external entities will use the virtual address to reach the agent platform, as it is depicted in figure (2) [5].

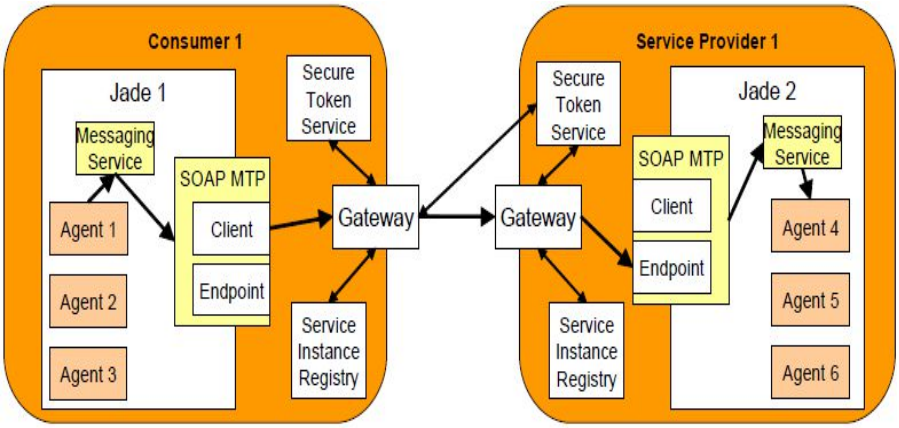

Figure 2: Enterprise Agent Interoperable Schema

In this way we can think of SOAP MTP driver as a transformation entity that maps SOAP messages to ACL messages, as figure (3) presents:

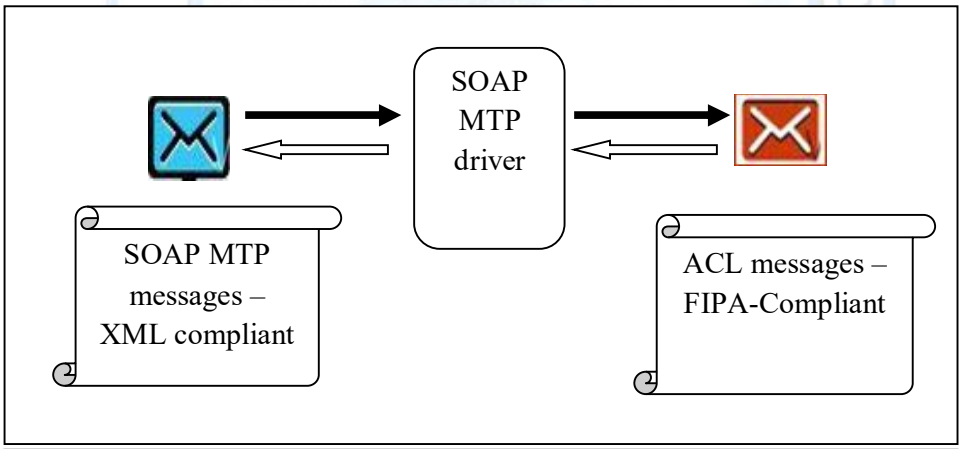

Figure 3: Mapping SOAP messages to ACL Messages and Vice Versa

4. Java Agent Development Framework (JADE)

JADE is the most widespread Agent-oriented middleware and it is a completely distributed middleware system with a flexible infrastructure allowing easy extension with add-on modules. The framework facilitates the development of complete Agent-based applications by means of a run-time environment implementing the life-cycle support features

required by Agents, the core logic of Agents themselves, and of language features.[8] figure (4) presents Agent Management reference model used by JADE environment [5, 6, 7]

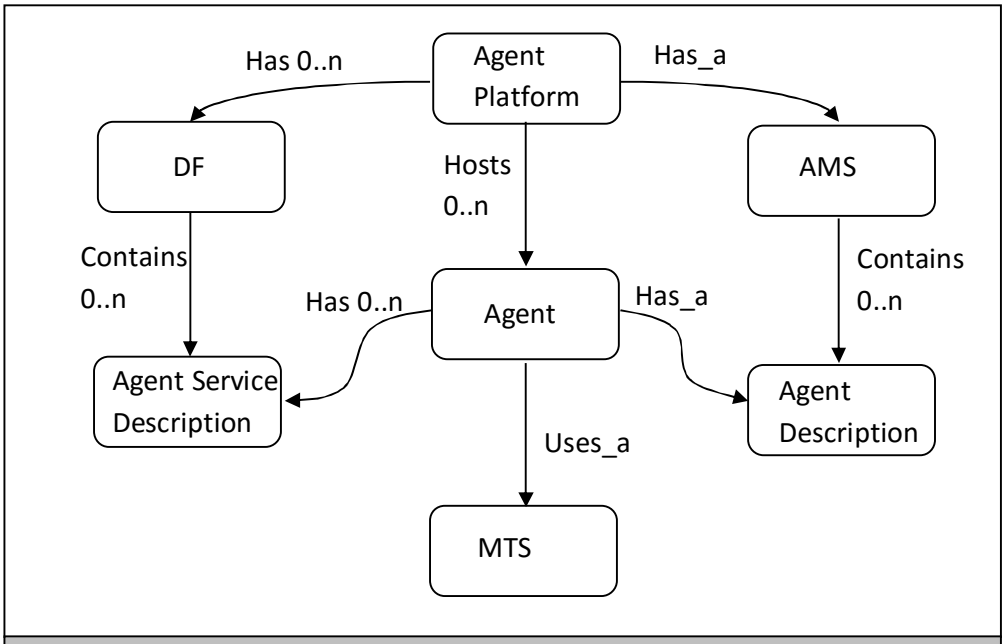

Figure 4 : Java Agent Development Platform Components

From figure(4) main components of the model can be briefly described as the following:

- Agent Platform (AP): Agent physical infrastructure in which Agents are deployed, this component includes machines, operating systems, FIPA agent management components, Agents.
- Agent: computational process that inhabits an AP and typically offers one or more computational services that can be published as a service description.
- Directory Facilitator (DF): the DF is an optional component of an AP providing yellow pages services to other agents. It maintains an accurate, complete and timely list of agents and must provide the most current information about agents in its

directory on a non-discriminatory basis to all authorized agents. An AP may support any number of DFs which may register with one another to for federations. [7,8]

5. Challenge

The challenge that this paper is targeting is: JADE environment provides many frameworks and platforms for Agent communication, where multiple Agents are capable of communicating each other; the communication unit is the ACL message. The challenge is how to address Agents reside far over web platform. Web provides tremendous resource repositories for knowledge and massive collaboration environment, but unfortunately, JADE does not support web platform. In other words, JADE platform should be managed locally.

6. The proposed Cloud Based Firewall

Firewall is rule based software with two actions: allow and deny, these general class of actions are shaping out the network traffic crossing over the firewall. Intelligent software Agents were used to enhance software adaptivity to changes occurred within the environment. Multiple Agents are composing a collaborative society through which the knowledge is developed

The proposal of this paper is presenting an inter-enterprise cross organization framework. In this framework Agent collaboration is beyond organization boundaries where a global Agent System Management (GAMS) is published as a global web service; Agents are capable of logging in local platforms and as the same time in global platform (i.e., AMS is published as web service).

At the initialization phase Agents who are interested in joining Agent Global Management System will send booting request; if the sender Agent is authenticated then the GAMS will launch a proxy Agent for that requester. Anyway, this paper focuses on the deploying of this framework rather than explaining its architecture, the general structure is presented in figure(5)

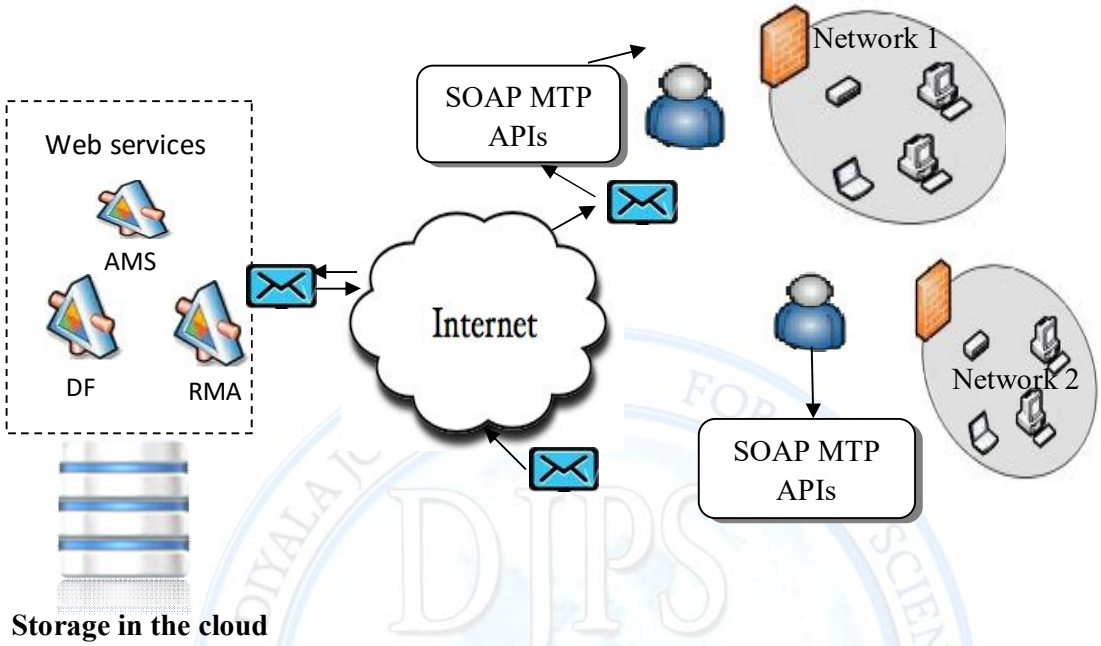

Figure 5: Web Based Firewall Agent Communication Schema

SOAP MTP provides a communication framework through which the Agent's behaviors can be integrated to web services survived in the cloud.

In this proposal web services are designed to play the role of RMA and AMS Agents, and these agents are the integration stub that work as a global platform for all platform siblings. A new class of Agent is to be designed and the initial behavior is to be overridden to extend its functionality; extended initial functionality will be responsible on initiating web service communication session.

We will use WebAgent to refer to the Agent started as web service, WebAgent is normal Agent instance and it is much like Agents created on hosts.

WebAgent initialization method calls traditional Agent initial method but it adds special code to do Agent registration within the global platform, the registration is conducted

through sending SOAP message to RMA Agent which it is himself a web service published as SaaS.

Agent registration over the cloud is initiated by invoking special web service called 'AgentRegService' as figure (6) depicts:

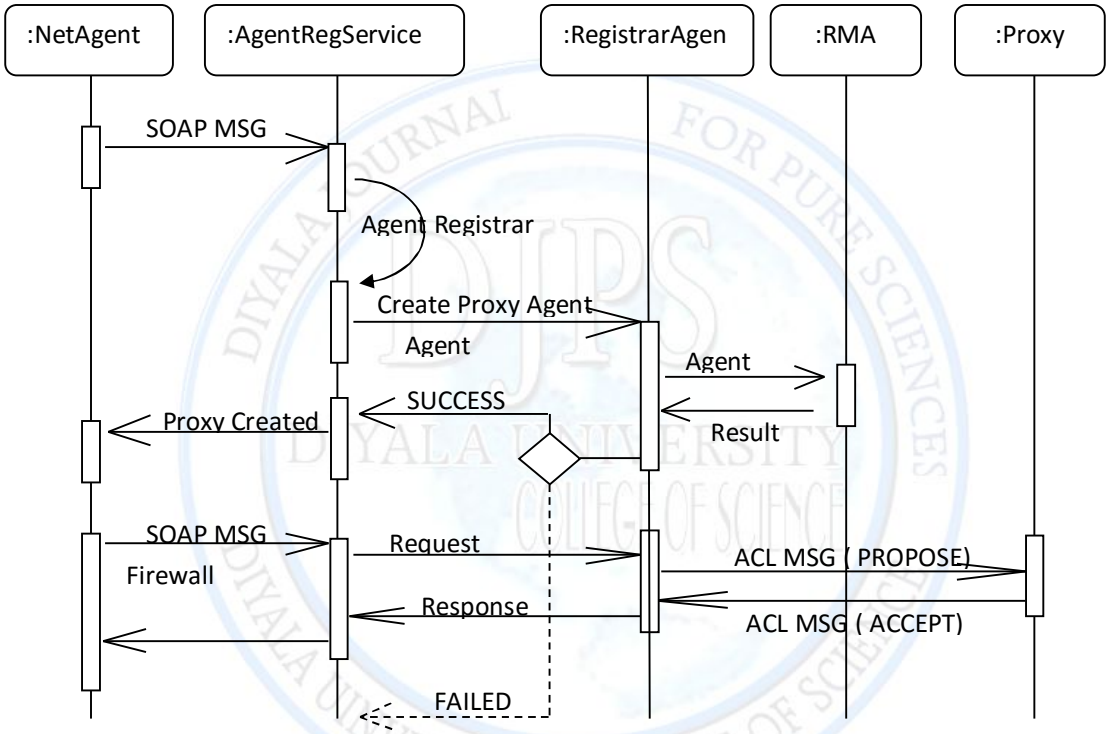

Figure 6: Firewall Agent Communication Sequence Diagram over the Cloud

XML schema for Agent identification is presented below; this is important for Agent-to-Agent communication

```
<agent  xmlns="x-schema:AgentID.xsd">
        <name > "AgnetName"</name>
        <host>  "HostName"</host>
```

```
<platform>  "PlatformName"</platform>
<container>      "URL or IP " </container>
<port> "default: 1099"  </port>
</agent>
```

XML schema for Firewall Policy also needed to better understanding for the content, the following the general XML schema for representing rules.

```
<policy  xmlns="x-schema:FirewallPolicy.xsd">
    <rule id="ruleID">
        <from>   "URL or IP" </from>
        <to>     "URL or IP " </to>
        <action> "action type"  </action>
        <constraints> "if-else form" </constraints>
    </rule>
</policy>
```

As we announced earlier, for each firewall Agent, some Agent proxy is initiated and invoked over the cloud. Firewall Agents are already joined platforms; otherwise it won't be allowed to take off.

Figure (7) presents flowchart of starting new Agents (i.e., proxy Agent) and presents action taken by started proxy Agents when receiving SOAP message.

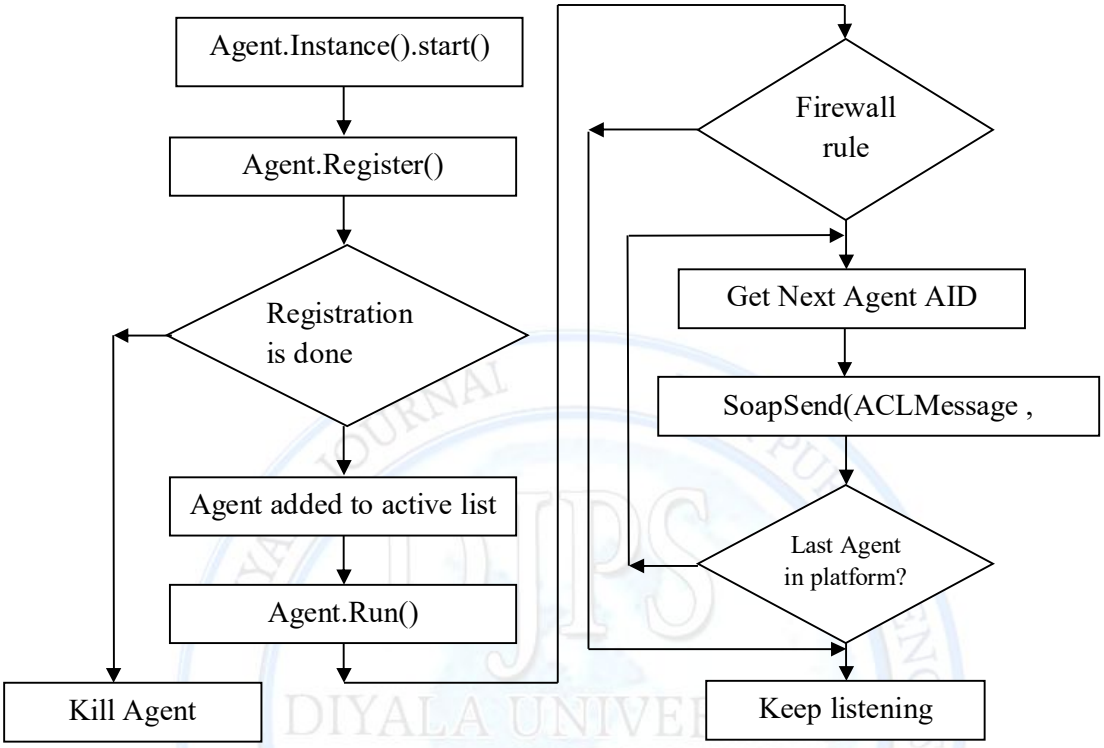

Figure 7: Flowchart Agent Registration Using Web Service published over the cloud

For experimental test we created web service in local host with the following specifications:

Service Name: { http://server/} AgentWebService

Port Name: { http://server/} AgentWebServicePort

Address: http://localhost:8080/WebService/AgentWebService

WSDL: http://localhost:8080/WebService/AgentWebService?wsdl

Implementation class: AgentWebService

Figure (8) presents Java based sequence to call web services remotely and without using the console published by the service provider. Figure (8) is a programmatic approach to search for web services.

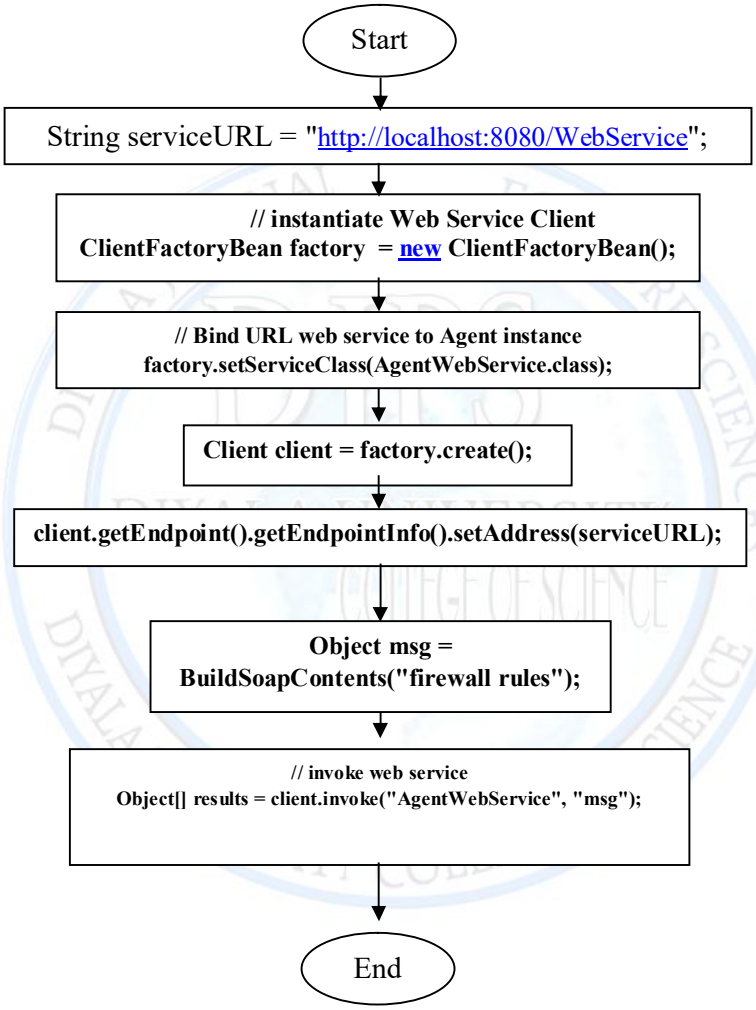

Figure 8: Agent Integration to Web Service using Soap MTP driver

Before individual Agents can started, RMA agent of JADE has to start up as figure (9) presents, where JADE GUI is started with standard and essential Agents.

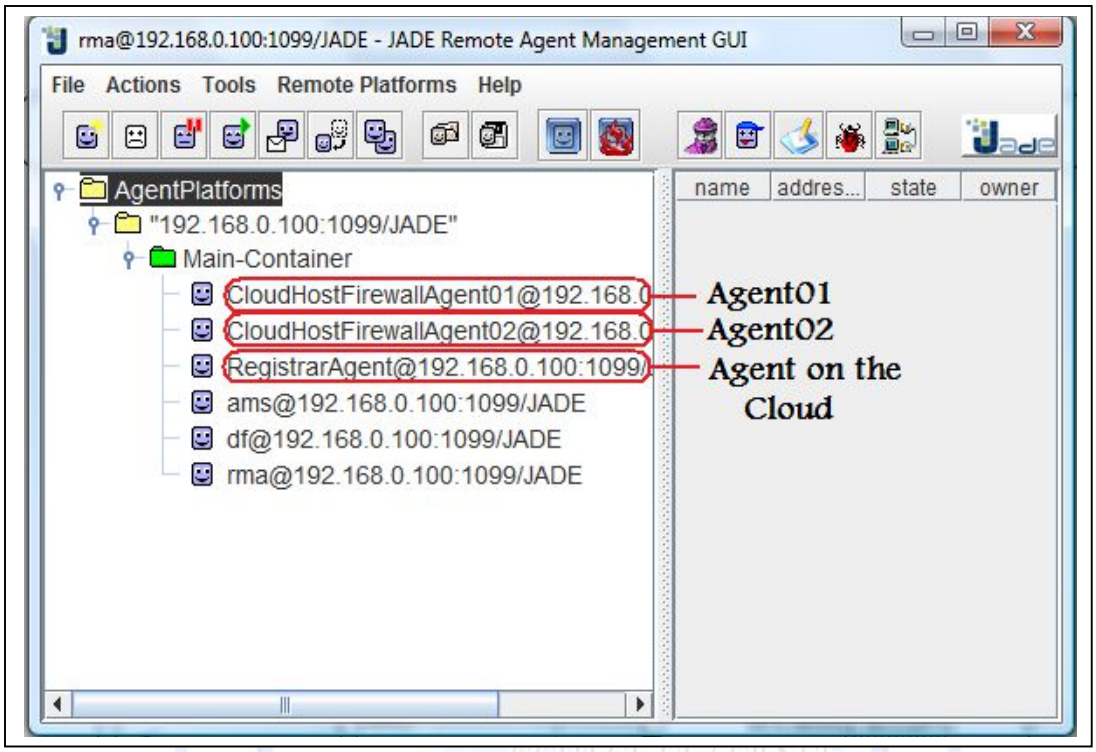

Figure 9: JADE platform Started On Server Side

From figure(9), Agent have started and joined the platform, each Agent has the ability to talk to web service and call its operations, as figure (8) presents. Two Agents have started along the test; this is beside to the monitoring and authentication Agent, RegisrarAgent. Client02 in figure (9) has encountered rule changed on the local machine, hence, SOAP message is built and send over the network to the web service. The whole operation has been presented in figure (6).

The exchanged messaged is SOAP based messaging protocol, but Agent uses SOAP MTP translator to convert SOAP message to ACL Messages. Figure (10) presents captured message crossing to CloudHostFirewall02.

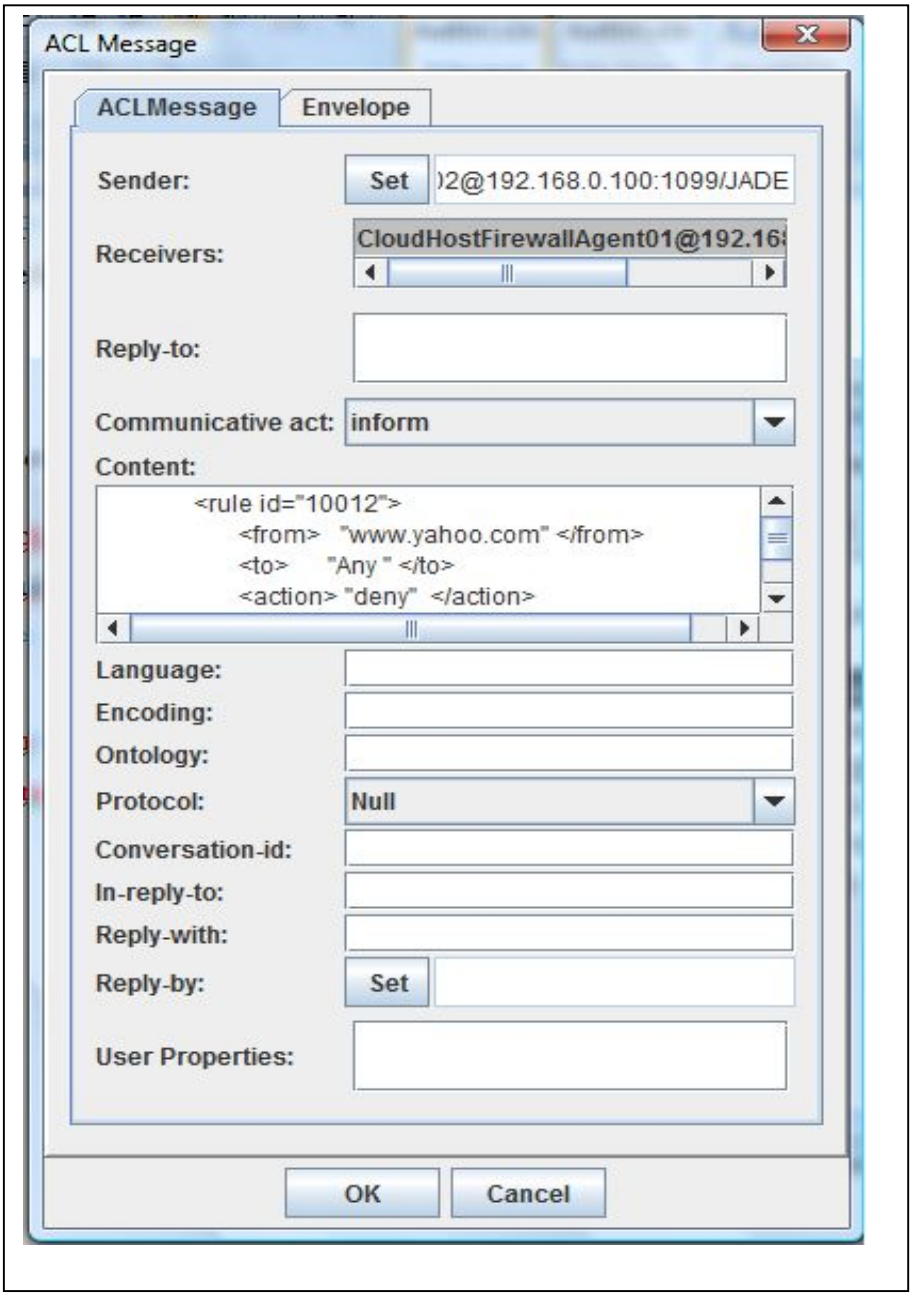

Figure 10: ACL Message Sent Over the JADE platform over the cloud

Conclusions

1- Cloud based applications provide an enormous potential for building globally inter-operable enterprise applications especially for Agent based systems due to the lack of standards to integrate JADE Agents together over the new emerging platform of cloud computing. SOAP MTP driver, which is pluggable to JADE, adds the ability to communicate web service. This can be deployed to build enterprise gateways for wide spectrum of web based applications.

2- Knowledge is developed in enterprise environment in a rate proportional to the interactions occurred among system software components. Multi-Agent based software system is supposed to exploit intuitive properties and characteristics of Agent's systems such as the socialization, where Agent can socialize with other Agents by exchanging knowledge.

References

1. Nick Antonopoulos and Lee Gillam, "Cloud Computing: Principles, Systems and Applications", Springer, USA, 2010.
2. Rajkumar Buyya, James Iroberg and Andrzej M. Goscinski, " Cloud Computing: priniciples and Paradigms", John & wiley, Australia, 2010.
3. Michat Wodczak, “Autonomic Cooperative Networking”, Springer, US, 2012
4. Klaus Fischer, Jorg P. Muller, Renato Levy,” Agent-Based Technologies and Applications for Enterprise Interoperability”, Springer Heidelberg, US, 2012-08-06
5. Lin Padgham and Michael Winikoff, "Developing Intelligent Agent Systems, A practical Guide", John Wiley & sons Ltd, USA, 2004.
6. Andras Micsik, Peter Pallinger and achim Klein, "SOAP based Message transport for the Jade Multiagent Platform", International workshop, ATOP, Toronto, Canada, 2010.
7. Rainer Unland, Matthias Klusch and Lonique Calisti, "software Agent-Based: Applications, Platforms and Development Kits", Birkhauser Verlag, Switzerland, 2005.
8. Rafael H. Bordini, "Multi-Agent Programming: Languages, Platforms and Applications", Springer, USA, 2005.