OJADEAC: An Ontology Based Access Control Model for JADE Platform

Ban Sharief Mustafa
Computer Sciences Department, Mosul University
Mosul , Iraq

Najla Aldabagh
Computer Sciences Department, Mosul University
Mosul , Iraq

***Abstract*—Java Agent Development Framework (JADE) is a software framework to make easy the development of Multi-Agent applications in compliance with the Foundation for Intelligent Physical Agents (FIPA) specifications. JADE propose new infrastructure solutions to support the development of useful and convenient distributed applications. Security is one of the most important issues in implementing and deploying such applications. JADE-S security add-ons are one of the most popular security solutions in JADE platform. It provides several security services including authentication, authorization, signature and encryption services. Authorization service will give authorities to perform an action based on a set of permission objects attached to every authenticated user. This service has several drawbacks when implemented in a scalable distributed context aware applications. In this paper, an ontology-based access control model called (OJADEAC) is proposed to be applied in JADE platform by combining Semantic Web technologies with context-aware policy mechanism to overcome the shortcoming of this service. The access control model is represented by a semantic ontology, and a set of two level semantic rules representing platform and application specific policy rules. OJADEAC model is distributed, intelligent, dynamic, context-aware and use reasoning engine to infer access decisions based on ontology knowledge.**

Keywords*—Java Agent Development Framework (JADE); JADE-S; Ontology-Based Access Control Model; Web Ontology Language (OWL)

I. INTRODUCTION

Currently Multi-Agent system provides a platform to build open distributed systems including e-commerce, web-services and pervasive computing environments. Security is an important issue in most of these applications and must be quarantined or they will face significant deployment problems. JADE is a popular Multi-Agent platform used in many commercial, academic and scientific agent-based Projects.

There are a number of extensions to JADE that provide a security platform to the system in particular S-Agent and the JADE-S plug-in. JADE-S is a security Add-ons component providing secure platform with authentication, authorization, encryption and public key infrastructure [1]. The authorization in JADE-S depends on Java Security model and Java access controller. JADE-S extends this controller to act with JADE platform architecture and permissions [2]. It structures the platform as a multi-user environment where every agent or container will be owned by an authenticated user, who is authorized to perform several privileged critical actions. However these permissions are an extension to Java Permission objects to support JADE platform actions. Its access control is depends on Identity Based Access control, where permissions are given based on the identity of the user. Thus it cannot support policy, attributes, and context aware access decisions. Also, a fine grained access rule cannot be adopted in such model.

By introducing Semantic Web and Semantic Web technologies from Berners-Lee [3], a new intelligent and semantic vision is introduced for building security services. The new Semantic Web technologies shows great promising in building semantic based security services especially in building an access control model.

In this paper, OJADEAC model, an ontology based access control based on Semantic Web technologies is proposed. OJADEAC relies on a proposed JMASO ontology that models the JADE Multi-Agent system knowledge with any information needed to support access decisions. OJADEAC is a policy model where an access is taken according to access control policy rules. Policy rules are specified at two levels: platform and application. Complete authorization architecture is provided by building a kernel service that automatically enforces access control policies on request to JADE commands. OJADEAC model shows great advantages over JADE-S authorization model. However it also suffers from some drawbacks including model performance and JADE shortcomings.

The paper is structured as follows: section 2 presents the related works. Section 3 introduces JADE and JADE-S platforms. Section 4 gives a brief introduction to OWL. Section 5 deals with model and model implementation, introducing JADE multi-agent ontology and the model architecture.

II. RELATED WORKS

Several projects build a secure agent platform based on JADE-S secure platform extending and substituting its authorization service. In other side, using Semantic Web technologies in access control mechanisms has taken considerable attention from different researchers who build an ontology based access control for different domains. Related works are reviewed in these two dimensions. The first dimension on extending JADE-S authorization service, Vila and Schuster [4] built an Intelligent Learning Management System called EUME, based on JADE-S. Permission service is extended to define a fine-grained access to specific

functionality provided by special agents by implementing a two authorization layers. Vitabile [5] proposed a new access control model which merges the advantages of the three classical models: Role Based Access Control (RBAC) model, Mandatory Access Control (MAC) model and Credential Based Access Control (CBAC). The new model replaces JADE-S authorization service.

In using Semantic Web technologies to build access control model, Giunchiglia [6] presented Relation Based Access Control (RelBAC) which is a model and logic to deal with the problem of access control in Web 2.0 applications representing access control rules and policies as DL formulas and reasoning about them using reasoner. Masoumzadeh [7] proposes social network system ontology, ontology based access control model and authorization policy rules to address the protection of semantic rich information in the knowledge base ontology for social network.

Shen [8] proposes semantic context-based access control model to be applied in a mobile Web services environment by combining Semantic Web technologies with context-based access control mechanism. His work focuses on the context aware access decision due to the characteristic of the open mobile Web services pervasive applications.

III. JADE AND JADE-S

JADE is a platform that provides basic middleware–layer functionalities which are independent of the specific application and which simplify the realization of distributed applications that exploit the software agent abstraction [9]. A JADE platform is composed of agent containers that can be distributed over the network. A special container called main container represent the bootstrap point of a platform [10]. JADE based on distributed coordinated filters architecture. According to this architecture every agent based operation will be forwarded to the service responsible for implementing it [10].

JADE supports defining a new kernel service to implements new agent operations. The kernel Service composed form several components including: outgoing and incoming sinks responsible for implementing the commands belonging to this service, service slice which represents a service proxy in other nodes. At last a two chain filters work as incoming and outgoing filters. Service can use outgoing filter to intercept any command issued from another service and react in a service specific way. Service use incoming filter to intercept any vertical commands issued by service slices in that node and react in a service specific way [10]. Figure 1 shows the filter architecture for message kernel service.

JADE-S is formed from standard JADE with JADE Security plug-in. JADE-S provide some security features to JADE platform. It extending Java security model provides the advantages of Java Authentication and Authorization Service (JAAS), Java Cryptography Extension (JCE) and Java Secure Socket Extension (JSSE). It allows exchanging critical information through a network using a secure data transmission (SSL) [3] .

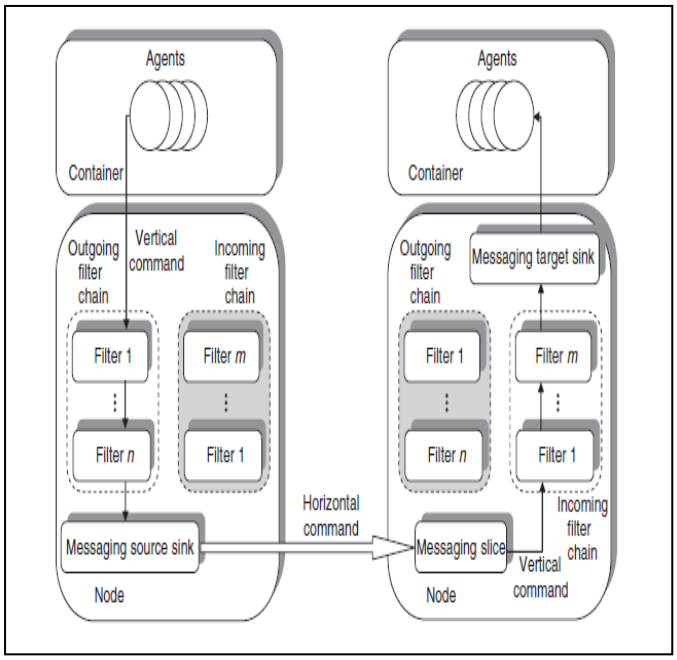

Fig. 1. Filter architecture for message kernel service [8]

JADE-S structures the agent platform as a multi-user environment where every agents and containers will belong to authenticated users. A permissions file contain a set of actions that each user is authorized to perform will represent the security policy of the platform [2]. JADE-S provides four JADE kernel services: ***Security Service*** which represents the base service to run JADE-S, and carried the authentication process. ***Signature and Encryption Services*** which provides Java cryptography message signing and encryption methods to assure message integrity, non-repudiation and confidentiality [1]. ***Permission Service*** based on an access control list saved in a policy file that follows the Java/JAAS syntax to make a decision on all actions that agents can perform in the platform [4]. The main policy file resides in main container, while other containers will have their own local policy file. Right given to an authenticated user will be translated to all its owned agents [4].

Access rights will enforced during the platform execution in a point that based on the Permission kernel service filters. The check command method performs the authorization check on received command. The main drawback for this service is that it is based on Identity Based Access Control (IBAC) model. Where every access decision is taken based on the permissions given to authenticated username. Such model is not suitable to be implemented in an open and distributed applications, thus many research projects that based on JADE-S platform tend to replace this service with another one to overcome its deficiency as mentioned in related work.

IV. WEB ONTOLOGY LANGUAGE

OWL is a Semantic Web language proposed by W3C [11], and is probably the most popular language for creating ontologies today.

OWL defines classes, properties, and their hierarchies. OWL gives a more expressivity in expressing a complex and richer relationships with greatly enhanced reasoning ability [12]. OWL comes in three different versions: Lite, DL and FULL. The most suitable one in building ontologies and reasoning over it is DL, because it is designed to support existing description logics, and has properties that are desirable for reasoning systems [13].

V. Model Implementation

An ontology-based access control model called OJADEAC is proposed based on Semantic Web technologies. OJADEAC is a fine grained semantic aware model that protects resource access in JADE platform and service access in domain specific application. The model relies on an ontology called (JMASO) that model the JADE Multi-Agent platform knowledge by storing keys entities and their relationships typically found in JADE and any information related to access control purpose including inferences based on access control policy rules.

A. JADE Multi-Agent System Ontology

JADE ontology called JMASO is proposed that modelsmain concepts with its relationships found in JADE platform. JAMSO can be extended to cover other aspects in this domain. The current version of JMASO contains 6 concepts, 10 object properties and 8 data properties. Protégé editor has been used to create JMASO ontology. Figure 2 show the complete ontology graph. Node concept represents any container in JADE platform with two descendents: MainContainer and Container.

The Principle concept represent any entity that can be responsible about their actions, it specialize into two concepts: Agent and Owner. Owner represent all users begin a new node. Agent is the class of all agents created in platform. Node is linked to Owner with OwnedBy object properties. Agent is linked to Node with hasLocation object properties. NamedAction is an important class that models all action in JADE that can be requested and checked against access control policy rules. Service concept models all services registered in DF agent. NamedAction concept has three descendents subclasses: AgentAction, ContainerAction and ServiceAction. NamedAction relates to Agent with hasSubject property and relates to Agent or Service with hasObject property.

PermissionAction is specialized to two disjoint classes ProhibiedAction and PermittedAction. Access check is taken according to the inferred type of NamedAction object which must be either PermittedAction or ProhibiedAction.

B. OJADEAC model architecture

OJADEAC authorization architecture is shown in Figure 3. The diagram reflects the following logical actors involved in OJADEAC model:

Knowledge Base: is composed form JMASO ontology and a set of two levels of policy rules (system and application specific). In OJADEAC model, Jena rules are used to express a policy rules. For example, the following platform policy rule says that if agent created in container that is owned by action requester, then it is permitted action.

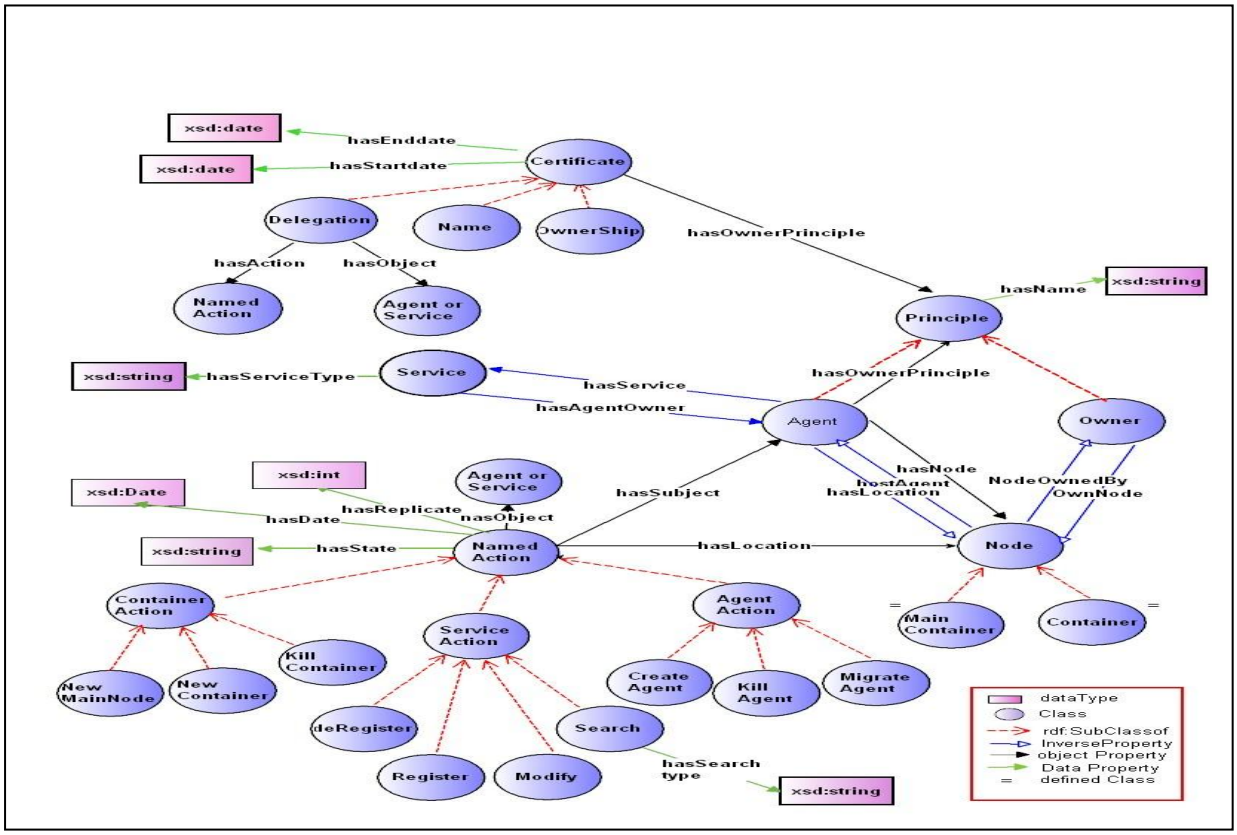

Fig. 2. JAMSO ontology graph

[permittedCreate: //name of rule
(?x rdf:type ac:CreateAgent)

(?x ac:hasSubject ?y)

(?x ac:hasLocation ?z)

(?z ac:NodeOwnedBy ?y)

->

(?x rdf:type ac:permittedAction)]

Ontology manager: It is responsible for gathering and updating A-Box knowledge parts. New individuals and its related properties with other individuals and data values are asserted to knowledge base during runtime. Ontology manager is distributed over nodes gathering its knowledge during runtime. Main container will contain knowledge about the whole platform by sharing and exchanging it with other node's knowledge. RDQL is used in knowledge sharing and exchanging between different ontology knowledge scattered on different nodes in platform. When node knowledge need an information from other node, it send an ACL message with RDQL query as message content to this node or to main container and gets response as sub model to be added to its ontology. For example, the following RDQL query asks for triples that have subject (?x) which own Container-1:

" SELECT ?x WHERE {?x ont:OwnNode ont:Container-1}"

ont: is the prefix name space of JMASO ontology. The query answer is a set of triples with subject zaid binding to ?x variable:

[http://Onto.owl#zaid, http://Onto.owl#hasJob, "Faculty"]

[http://Onto.owl#zaid,http://Onto.owl#hasRole, ContainerAdmin"]

[http://Onto.owl#zaid, http://Onto.owl#hasDegree, "null"]

[http://Onto.owl#zaid,http://Onto.owl#hasEmail, "sss@yahoo.com"]

[http://Onto.owl#zaid, rdf#type, http://Onto.owl#Owner]

[http://Onto.owl#zaid, rdf#type, owl#Thing]

[http://Onto.owl#zaid, rdf#type, http://Onto.owl#Principle]

[http://Onto.owl#zaid, rdf#type, http://Onto.owl#Agent]

Policy Enforcement Point (PEP): OntSecure service is a new kernel service that is added to JADE platform to support OJADEAC model implementation. The PEP will be placed in OntSecure incoming and outgoing filters. PEP is different form one command to other depending on the sequence steps executed to implement this command. Figure 4 shows sequence steps including OJADEAC PEP and ontology assertions when REQUEST-CREATE command is issued.

Policy Decision Point (PDP): Decision is made by inference over JAMSO ontology and policy rules to infer action type. The action will belong either to PermittedAction or ProhibiedAction. A method to add a new action to ontology will activate the inference process, and then action type will be checked. If action is prohibited, then a security exception is raised and command is blocked, else command will continue its implementation.

Policy Administrator Point (PAP): platform administrator and application vendors are responsible for creating policy and policy rules set.

VI. DISCUSSION

OJADEAC model is an ontology based access control for JADE platform. It substitutes its authorization service providing several advantages over it. OJADEAC features over JADE-S authorization service can be summarized in following points:

1) Using ontology provides reasoning ability for access control decision. Also access control information can be accessed, queried and discovered automatically including owner attributes which can be deduced from knowledge by reasoning.

2) The proposed model has a higher degree of interoperability compared with other approaches to access control. This is because of the nature of ontologies in providing semantic interoperability.

3) Fine-grained access control policies are expressed using a set of Jana rules. Policy rules are formulated either as a system level policy rules or application level policy rules.

4) In addition to supporting ABAC decisions and RBAC decisions, OJADEAC model can support context aware decisions because policy rules can take into account any resource, object and environment conditions. These constraints can be evaluated by reasoning over OMASJ ontology model. Date, time and location in addition to others can be included in decision making.

5) OJADEAC model can store past agent/user behaviors. So it can adapt a trust and reputation model and detect any abnormal behavior to act as intrusion detection.

VII. CONCLUSIONS

As OJADEAC model has several features, it also suffers from some draw backs that need to be addressed and solved to apply this model in real applications implementation. The first drawback is its performance due to time spent in reasoning and in managing ontology. Another problem is JADE shortcomings that affect the model behavior. Some commands are implemented in a way that cannot be intercepted in service filters (example, DF service commands). Other problem is the missing parameters attached to some commands (example, KILL-CONTAINER requestor parameter is always NULL) which effect making a decision. At last, the model is a good start to overcome the shortcoming of JADE-S permission service and to reduce the gap with Semantic Web applications.

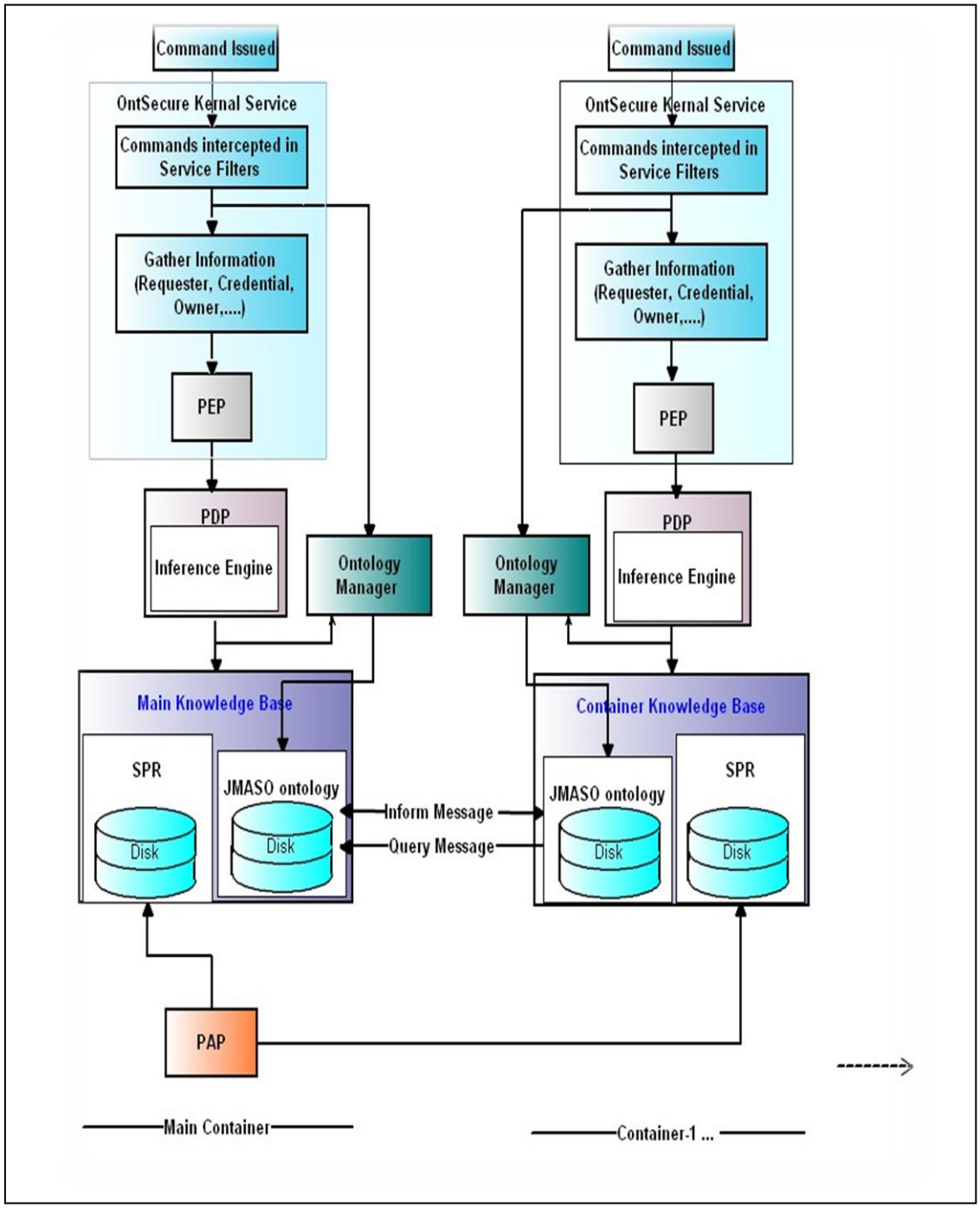

Fig. 3. OJADEAC model architecture

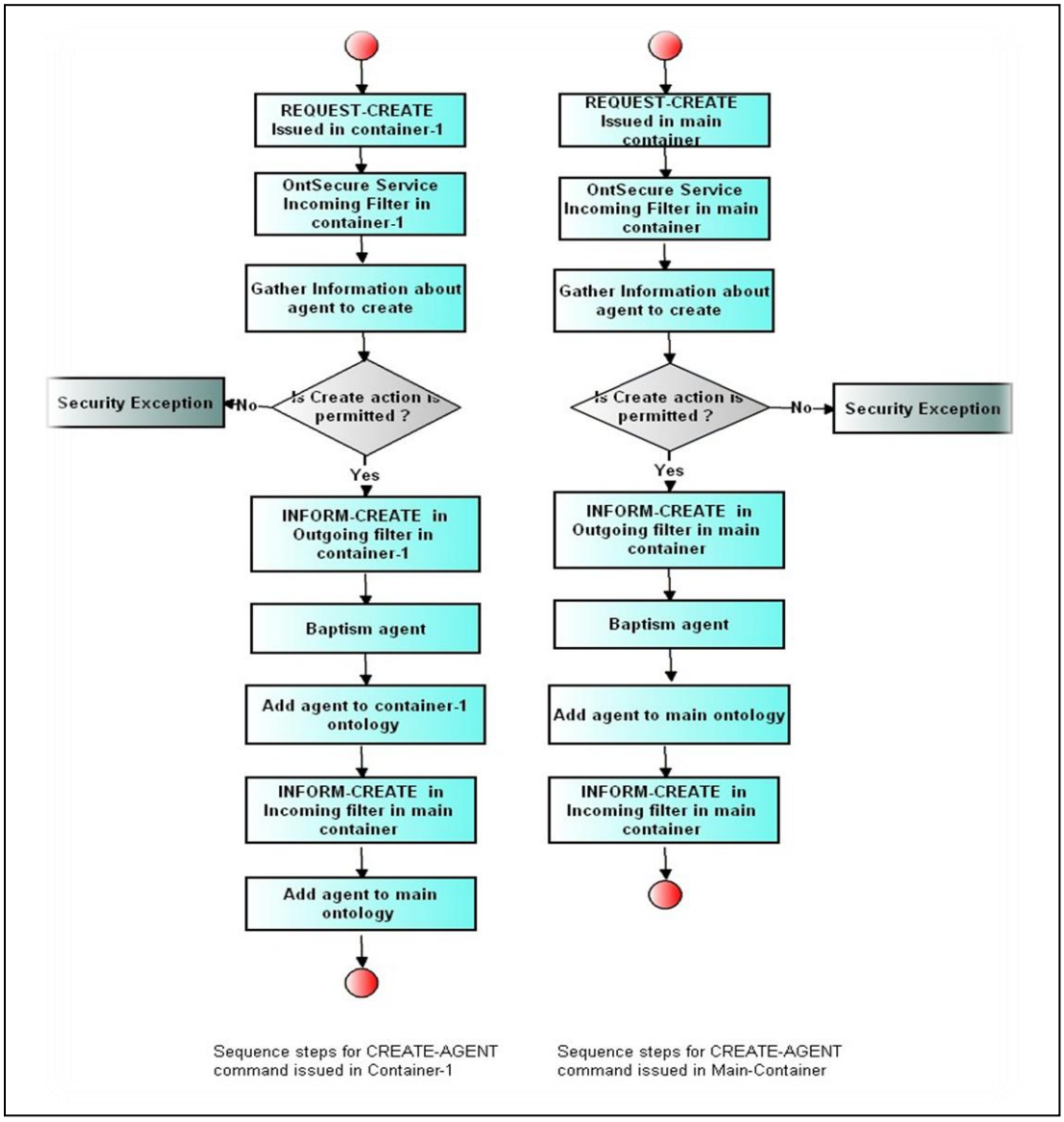

Fig. 4. Sequence steps for CRAETE-AGENT command

REFERENCES

[1] G. Vitaglione, "JADE Tutorial, Security Administrator Guide" 1, 19 Sep. 2002.

[2] A. Moreno, D. Sanchez, and D. Isern, "Security measures in a Medical Multi-Agent System", in Proceeding Artificial intelligence and Applications, 2005.

[3] JADE Board, "JADE Security Guide", Technical Report, TILab, Telecom Italia, 28 February 2008.

[4] X. Vila, A. Schuster, and A. Riera, "Security for a Multi-Agent System based on JADE", Computer and Security, Vol. 26, 2007, pp.391-400.

[5] S. Vitabile, V. Conti, C. Militello and F. Sorbello, (September 2009), "An extended JADE-S based framework for developing secure Multi-Agent Systems," Computer Standards & Interfaces, Vol. 31, No. 5, September 2009, pp. 913-930.

[6] F. Giunchiglia and R. Zhang, "Ontology Driven Community Access Control", Technical Report, University of Trento, Dec. 2008.

[7] A. Masoumzadeh, and J. Joshi, "Ontology based Access Control for Social Network Systems", Information Privacy, Security and Integrity, Vol. 1, No. 1,2011 .

[8] H. Shen and Y. Cheng, "A Semantic Context-Based Model for Mobile Web Services Access Control", Computer Network and Information Security, Vol. 1, 2011, pp. 18-25.

[9] A. LUPAŞC, "A Multi-Agent Platform for Developments of Accounting Intelligent Applications", Economics and Applied Informatics, No. 1, 2008, pp. 79-86..

[10] F. Bellifemine, ,G. Caire and D. Greenwood, "Developing multi-agent systems with JADE", John Wily & Sons, Ltd, 2007.

[11] W3C OWL Working Group, "OWL 2 Web Ontology Language: Document Overview". W3C Recommendation, Internet: http://www.w3.org/TR/owl2-overview/.

[12] N. Aldabagh, and B. Mustafa, "A Comparative Study between Using OWL Technology and Jess Rule Based For Applying Knowledge to Agent Based System", International Journal of Computer Science and Information Security, Vol. 10, No. 7. July 2012.

[13] I. Horrocks, F. Peter and P. Schneider, "KR and Reasoning on the Semantic Web: OWL", In Handbook of Semantic Web Technologies, Ed. Domingue, J., Fensel, D., and Hendler, A., Chapter 9, 2011, pp. 365-398. Springer.