Identity Management in Agent Systems

David de Groot and Frances Brazier

IIDS Group, Computer Science Department, Faculty of Sciences,
Vrije Universiteit Amsterdam,
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{dra.de.groot, fmt.brazier}@few.vu.nl
http://www.iids.org/

Abstract. If agent-based applications are to be used in large scale, open environments, security is a main issue; digital identity management (DIDM) an essential element. DIDM is needed to be able to determine the rights and obligations of the four main computational entities in such systems: agent platforms, hosts, agents, and services. The framework for evaluation of DIDM in agent systems proposed in this paper is based on four aspects of DIDM: representation, confidentiality, integrity and availability. Two agent platforms (JADE-S and AgentScape) are used to illustrate the potential of this framework.

1 Introduction

Agent technology is a promising and enabling technology in large scale distributed environments. Mobile agent-based applications offer new possibilities for local and secure access to online services [1]. Digital identity management is a prerequisite [2, 3] for secure access: the rights and obligations of all entities in an agent system need to be determined and secured.

This paper distinguishes four main computational entities in an agent system: agent platforms, hosts, agents and services. Section 2 defines these entities and the related principals. Section 3 discusses four aspects of DIDM: representation, confidentiality, integrity and availability. Section 4 proposes a framework for the evaluation of DIDM in agent systems. Two existing agent platforms: JADE and AgentScape are evaluated and compared, illustrating the framework's potential. Section 5 discusses the results and areas for further research.

2 Principals in Agent Systems

Digital identity management (DIDM) is needed to support determine the rights and obligations of all entities in an agent system. This section defines the four main computational entities in agent systems: agent platforms, hosts, agents and services, and their related principals.

An *agent platform* is an environment that hosts agents and services. It is, in fact, middleware, i.e., a software layer between the operating system and the application programs. This middleware supports communication between agents, interaction with (web) services (e.g. look-up services: White and Yellow Pages to find agents and services), and often more. A platform necessarily provides agent life-cycle support (eg. to start, stop, resume and delete agents). Mobile agent platforms also support migration. Agents and services run on *hosts*. Examples of agent platforms designed to support security include: JADE-S [4, 5, 6], AgentScape [7], Ajanta[1], SeMoa[2] and Cougaar[3].

Seven principals can be distinguished and associated to each of the four main computational entities in agent systems: Admin, Auditor, Creator, Developer, Owner, Publisher and User. In this context, the word *principal* is used in the sense of "a person who has controlling authority or is in a leading position"[4] of a certain process (e.g. administration, creating, auditing, development, etc) of an entity. The only legal entities defined in current US and European legislation are individuals and or organizations (a legal entity is "an individual or organization which is legally permitted to enter into a contract, and be sued if it fails to meet its contractual obligations"[5]), see also [8, 9].

Table 1 presents an overview of the computational entities and their related principals that can be distinguished for agent systems. The 3rd column of **Table 1** indicates that, in most cases, principals are either human beings and/or organizations, thus legal entities. The principals responsible for service and agent creation, however, may also be computational entities. (In some agent platforms agents e.g. can spawn off new agents without explicit human intervention). This presents one of the main issues with which legal experts are now confronted, and of which agent developers need to be aware.

To determine responsibility and liability it is important that each computational entity has a legal entity as its responsible principal and computational entities are not legal entities. Borking [10] and Yee and Cunningham in [8], give a more elaborate discussion of Agent Ownership and related legal issues, but have not been able to come up with a complete solution.

3 DIDM Aspects

Digital identity can be defined as "an electronic representation of information about an entity" (adapted from [11] to fit the more specific setting of agent systems), which can be used for two purposes: 1) to denote a representation (a name or a pseudonym), e.g. an email address, or 2) to refer to a partial identity in digital form; any subset of properties associated with an entity, e.g. a profile of a user [11]. In this paper digital identity refers to both.

---

[1] http://www.cs.umn.edu/Ajanta/
[2] http://www.semoa.org/
[3] http://www.cougaar.org/
[4] Marriam-Webster Online, http://www.m-w.com/
[5] http://www.investorwords.com/

Representation, confidentiality, integrity and availability of the information stored and maintained in an agent system, are four of the most important aspects of DIDM. The next subsections elaborate on these four aspects.

Table 1. Principals in Agent Systems

Computational Entity	Related Principals	Principal type
Agent Platform	Agent Platform Admin	Human/Organization
	Agent Platform Auditor	Human/Organization
	Agent Platform Creator	Human/Organization
	Agent Platform Developer	Human/Organization
	Agent Platform Owner	Human/Organization
	Agent Platform Publisher	Human/Organization
	Agent Platform User	Human/Organization
Host	Host Admin	Human/Organization
	Host Auditor	Human/Organization
	Host Creator	Human/Organization
	Host Developer	Human/Organization
	Host Owner	Human/Organization
	Host Publisher	Human/Organization
	Host User	Human/Organization
Agent	Agent Admin	Human/Organization
	Agent Auditor	Human/Organization
	Agent Creator	Human/Organization/Computational
	Agent Developer	Human/Organization
	Agent Owner	Human/Organization
	Agent Publisher	Human/Organization
	Agent User	Human/Organization
Service	Service Admin	Human/Organization
	Service Auditor	Human/Organization
	Service Creator	Human/Organization/Computational
	Service Developer	Human/Organization
	Service Owner	Human/Organization
	Service Publisher	Human/Organization
	Service User	Human/Organization

3.1 Representation

A representation is "a creation that is a visual or tangible rendering of someone or something"[6]. Representation defines who and what is represented in a system, and how (identifier, name, picture, etc.). A *digital identifier* (digital ID) is a name associated with a digital entity: "a string of bits or characters that is used to refer to an entity" [12]. Names can identify a single thing, or a class or category of things, either uniquely, or within a given context (so-called global vs. local uniqueness). Globally unique digital identifiers (GUIDs) are useful for logging purposes and for traceability of mobile entities (e.g. software agents).

Digital identifiers of entities in agent systems are often defined according to an existing convention. Uniform Resource Names (URNs) and Uniform Resource

[6] http://dictionary.reference.com/

Identifiers (URIs) are examples of conventions used in digital environments. Related work focusing on naming of entities (agents, objects, etc.) can be found in [12] and [13].

3.2 Confidentiality

Confidentiality implies the need for information privacy. Any information that could possibly lead to the identification of a specific person is considered personal data (sometimes also called Personally Identifiable Information (PII)). Examples of personal data include user profiles, traces of transactions, and logs of interactions. In the European Union, rules on collection, processing and transfer of personal data are implemented in the member states' laws[7]. (An introduction to the privacy legislation of the EU and its application to software agents can be found in [9].) Examples of privacy related considerations are non-excessiveness, purpose specification, user consent and control-by-source. Designers of systems need to take these considerations into account. Non-excessiveness, for example, implies that a hosting site should not need to know the owner of an agent's date of birth for it to be allowed to access a service. Information that is not necessary in the context of the transaction at hand should not be requested.

Information privacy has a technical component as well. Privacy Enhancing Technologies (PETs), have been designed to protect the privacy of users, for example in on-line transactions. Although PETs are not specifically related to agent systems they can be applied. Examples of PET systems are the MASKS System [14], and PISA [15]. Two new endeavours in this area are the PRIME[8] [16] and FIDIS[9] projects.

In agent systems both the entities and their data need to be protected "only authorized entities can see protected data" [17]. Examples of confidentiality-preserving mechanisms are the use of pseudonyms for agents, anonymization of data, specification of access rights e.g. to protected information carried by an agent, and encryption of communication.

3.3 Integrity

Maintaining integrity can be described as a process with the following three specific goals [18]: 1) Preventing unauthorized users from making modifications; 2) Maintaining internal and external consistency; 3) Preventing authorized users from making improper modifications. A more elaborate discussion of integrity is beyond the scope of this paper (see e.g. [19] and [17]).

---

[7] Directive 95/46/EC of the European Parliament and of the council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

[8] http://www.prime-project.eu.org/

[9] http://www.fidis.net/

3.4 Availability

Availability can be defined as "ensuring that information and information processing resources both remain readily accessible to their authorized users" [18]. In agent systems, agents, for example, need to be able to find other agents, resources and services. White- and Yellow page services can serve this purpose [20, 21]. Whether or not a look-up system is centralized on a single server or distributed over a number of servers (e.g. replicated, peer-to-peer) may influence the availability of resources and services. Note that Denial-Of-Service (DOS) attacks on centralized services, for example, are far easier to realize on centralized systems than on distributed facilities.

4 DIDM evaluation framework

For each of the entities in an agents system (agent platform, host, agents and services) DIDM focuses on the following:

- Name(s)
- Address(es)
- Look-up
- Principals
- Meta-data
- Access regulation

Names are needed for representation, e.g. a unique identifiers, name and/or pseudonym with which an entity can be addressed. Addresses specify an entity's point(s) of access, [12]. *Look-up* services associated with an entity specify where information about other entities can be found. Related *principals* are those responsible for an entity. *Meta-data* describes the characteristics of an entity and its functions, e.g. information stored in a look-up service (FIPA compliant agent platforms, for example, publish agent platform descriptions locally for other entities to acknowledge [22]. *Access regulation* defines the terms and conditions for access.

Table 2 presents a framework with which agent platforms can be evaluated, based on the extent to which (1) this information is made explicit and (2) this information contributes to one or more of the four aspects of DIDM: representation (R), confidentiality (C), integrity (I) and availability (A).

Two agent frameworks, JADE-S and AgentScape, are used to illustrate the potential of this framework. JADE-S is currently the de-facto standard, and AgentScape is a new, open source agent framework. Before analyzing the results of the evaluation and comparison section 4.1 and 4.2 describe relevant aspects of the agent frameworks.

4.1 JADE-S

JADE (Java Agent Development Environment) is a FIPA-compliant agent platform [4, 5]. Each instance of the JADE run-time environment is called a *container* (since it

"contains" agents). A group of connected containers is called a *platform*. Multiple containers can run simultaneously on one single computer system or each container can have its own network-connected machine. JADE supports agent migration from container to container within a platform. In this article the secured version of JADE, JADE-S [6] has been evaluated. It supports multiple users on a platform, user authentication, agent actions authorization and message signing and encryption.

Platform Management (PM)
In JADE-S platforms can have arbitrary names. However, a default naming and addressing convention is advised: a unique name is generated from the host name and a port number (Name and Address, aspect: R) [23]. Information on the name, contact point of the platform and the available services are described in a publishable file "APDescription.txt" (Meta-data, aspects: R, A).

In Jade-S a platform administrator is assumed to take care of the administration of Platform Users (Principal, aspect: R). For each user a login name, password and Platform User Policy is specified. The Platform User Policy consists of a set of rules per user according to the Java Authentication and Authorization Service (JAAS) syntax (Access, aspects: C, I, A). By default, these policy rules are not visible to the users. To enforce security policies, a user authentication mechanism is used.

Host Management (HM)
In JADE-S containers are the actual run-time environments of agents. A JADE-S platform always starts with a Main-Container, which is appropriately named "Main-Container". Other containers connect to the Main-Container and are named "Container-" followed by a number (Name, aspect: R). A host name and a port number form a Container Address (Address, aspect: R).

Before a container is initialized, a login authentication by one of the platform users is required. Once successfully authenticated, this user is registered as the Container Owner (Principal, aspect: R). The owner of a container specifies a container specific policy that states the rights for other platform users who may be using the container (Access, aspects: C, I, A).

Agents can obtain the container names and addresses via the AMS and use these for migration to other containers (Look-up, aspects: R, A).

Agent Management (AM)
In JADE-S agents can have user-chosen, arbitrary names but these names have to be unique within the platform. Moreover, a security policy of the platform can enforce a namespace for a given user. For example, the security policy can specify for a user named Bob that (s)he can only create agents if the name contains the prefix "bob-", resulting in a name like "bob-agent1@platformA.myexample.org". The address of a JADE-Agent is the platform address; an agent can be reached by sending a message (specifying the agent-name) to the platform on which it resides (Name and Address, aspect: R).

A special service agent maintains agent administration at the Main-Container: the Agent Management Service (AMS), this agent is responsible for managing the agent information of all agents on the platform. The AMS provides a White-pages service

(Look-up, aspects: R, A), based on registered AMS Descriptions (Meta-data, aspects: R). Ownership of an agent can be registered at the AMS and, according to the `API`, ownership of an agent can change (Principal, aspect: R). Information registered in the AMS is public to all agents and users in the platform.

Agents can also register in the Directory Facilitator (DF) (Look-up, aspects: R, A), a FIPA-conformant Directory Service, with DF Descriptions (Meta-data, aspect: R). Agent registration at the DF is not obligatory (Look-up, Meta-data, aspect: C).

Service Management (SM)
In JADE-S two system services are available to agents: the AMS and the Directory Facilitator (DF). Other services are assumed to be encapsulated by agents and can be requested by contacting the agent providing the service. The service address is the Agent Name of the agent that encapsulates the service (Address, aspect: R). Service descriptions can be registered in the DF (Look-up, aspects: R, A) and specify a name (Name, aspect: R), an owner (Principal, aspect: R), language, protocols and optionally additional properties (Meta-data, aspects: R). Note that registration of services in the DF is no obligation, implying that information in a service can be kept confidential (Name, Address, Owner, Meta-data, aspects: C).

4.2 AgentScape

AgentScape is a framework for development and deployment of open, large-scale distributed agent systems and includes support for fault-tolerance, security, heterogeneity and interoperability [7]. The concepts used in AgentScape are locations, hosts, agents and services. A location represents multiple hosts, which provide runtime environments for agents. An agent runtime environment is a secure execution environment that supports migration, communication and service access. Agents can access directory services to attain the name and contact point of other agents and services they wish to access.

AgentScape's middleware security features include separate use of globally unique identifiers (GUIDs) and names (of agents and services), leasing of resources, sandboxing of agents, signing agent's code and its state, and secure communication.

Platform Management (PM)
In AgentScape agents migrate from one location to another and are unaware of underlying systems (hosts, operating systems, etc.). A Location Manager (LM) is a special middleware process that represents and manages a location. A LM is responsible for delegation of access rights to agents within a location (Access, aspects: C, I, A). AgentScape provides a lease-negotiation mechanism for agents to acquire time-limited access to other locations, e.g. CPU time, memory, hard-disk space etc. [24].

Upon initiation of the LM, the LM obtains a private/public key pair. Initialization of a location requires specification of a location's identifier (location name) and associated contact point (Name and Address, aspect: R). A location's name and contact point are registered in the Location Look-up Service (LLS). The LLS is only

accessible only to locations and other AgentScape middleware processes. Information in the LLS can be signed by other Locations (Look-up, aspects: R, I, A).

No user management facilities comparable to the facilities in JADE-S have been implemented in the AgentScape.

Host Management (HM)

An AgentScape location consists of hosts, which run instance(s) of the AgentScape software, i.e., running middleware processes. These processes register at a Location Manager, using an identifier (Host Name) and a contact point (Host Address) (Name and Address, aspect: R). A host middleware process may publish its service and resource descriptions (Meta-data, aspect: R) as well as its lease policies to the Location Manager (Access, aspects: C, I, A).

Agent Management (AM)

In AgentScape all agents have a globally unique identifier (GUID). The GUID is generated upon creation of the agent and is kept private to the middleware (Name, aspects: R, C). Agent can use multiple names for public uses, e.g. for registration in a Directory Service. The use of multiple agent names can protect privacy, i.e., it is a confidentiality preserving mechanism, since names can be used as pseudonyms for communication with other agents and interactions with services (Name, aspects: R, C). Agent Names are registered in the middleware in a Name Look-up Service (NLS), and only the middleware can couple an agent's name(s) and to its GUID (Look-up, aspects: R, A). Information in the NLS can be signed by other Locations (Look-up, aspects: I).

For protection of an agent and its state, e.g. to guarantee integrity of the agent's code and data, an agent and its data are stored in an Agent Container [25, 26]. Note that this Agent Container concept differs from the JADE Container concept: in AgentScape, it is storage medium, whereas in JADE it is a runtime environment for the agents. The AgentScape platform implements an integrity verification mechanism based on signing of Agent Containers (Meta-data, aspect: I).

Agent data can be stored in an Agent Container in encrypted segments. For example, a public key of a trusted remote location can be used for encryption of the data, such that the data can be made available again upon arrival at the remote host and only at the appropriate remote location, which can decrypt the data with its own private key (Access, aspect: C, I).

Directory Services are available for agents to find other agents, resources and services. Any agent can publish information in the directory services, such as attributes and a name. Information published in the location and directory services can be signed by other entities (such as a Location Manager). This can ensure integrity of the information and provides a mechanism for trust (Look-up, aspects: R, I, A). Note that agent registration at the DS is not mandatory (Look-up, aspects: C).

Service Management (SM)

In AgentScape services have a globally unique identifier (GUID) (Name, aspects: R, C), just like agents. Services are allowed to have names (Name, aspects: R, C) and just like the Agent Names, Service Names are registered in a Name Look-up Service

available only to the AgentScape middleware. Services can be advertised to agents through Directory Services (Look-up, aspects: R, I, A). Registration is, however, not obligatory. Access to services is regulated at the Location level via leases and by enforcement of Service Access Policies (Access, aspects: C, I, A).

Table 2 DIDM in JADE-S and AgentScape

	Managed Information	JADE-S Feature	AgentScape Feature	R	C	I	A
PM	Name	Platform Name	Location Names	♩α			α
	Address	Main-Container Address	Location Address	♩α			α
	Look-up	-	Location Look-up Service	α		α	α
	Principals	User	-	♩			
	Meta-data	Platform Description	-	♩			♩
	Access	Platform User Policy	Location Policy		♩α	♩α	♩α
HM	Name	Container Names	Host Name	♩α			
	Address	Container Address	Host Address	♩α			
	Look-up	AMS	-	♩			♩
	Principals	Container Owner	-	♩			
	Meta-data	-	Service and Resource Descriptions	α			
	Access	Container User Policy	Lease Policy		♩α	♩α	♩α
AM	Name	Agent Names	Agent GUID	♩α	α		
		-	Agent Names	α	α		
	Address	Platform Address	Location Address	♩α			
	Look-up	AMS	-	♩			♩
		DF	DS	♩α	♩α	α	♩α
		-	Name Look-up Service	α			α
	Principals	Owner	-	♩			
	Meta-data	AMS	-	♩			♩
		DF Descriptions	-	♩	♩		♩
		-	Agent Container Checksum			α	
	Access	-	Agent Container Segment Encryption		α	α	
SM	Name	Service Name	Service GUID	♩α	♩α		
		-	Service Name	α	α		
	Address	Agent Name	Location Address	♩α	♩		
	Look-up	DF	-	♩			♩
	Principals	Owner	-	♩	♩		
	Meta-data	Service Description	DS Registration	♩α	♩α	α	♩α
	Access	-	Service Access Policy		α	α	α

R = Representation, C = Confidentiality, I = Integrity, A= Availability

♩ = aspect in JADE-S
α = aspect in AgentScape

4.3 Comparison

Analysis of JADE-S and AgentScape illustrates the differences in design goals. The frameworks have been designed for different environments: JADE for relatively closed environments, AgentScape for large-scale, open environments. As a result JADE-S specifies a 2-tiered (specified at the platform and at the container level), static, centralized security policy, whereas AgentScape offers a dynamic and distributed approach with leases. Also, JADE-S uses centralized registration facilities (Main-Container, AMS and DF), whereas AgentScape provides distributed facilities. The latter are, for example, more resistant to Denial-Of-Service (DOS) attacks.

User administration, for adding/removing users and specifying their rights, are part of JADE-S. User management is still an open issue in AgentScape. Furthermore, JADE-S has more restrictions and conventions for naming of entities than AgentScape.

Further comparison of the two systems in **Table 2** shows that AgentScape provides more facilities and mechanisms for preserving confidentiality (privacy protection) and integrity of information for Agent Platforms, Agents and Services than does JADE-S.

Remarkable, but not so surprising is that neither of both agent platforms supports explicit management of all of the entity-related principals as presented in section 2. JADE-S supports administration of platform users and support for ownership settings of hosts (containers), agents and services. AgentScape currently does not provide any support.

5 Discussion

This paper presents a framework for analysis and evaluation of digital identity management (DIDM) in agent systems. To illustrate the framework, two exemplary agent platforms, JADE and AgentScape, are evaluated and compared. The results of the evaluation indicate that a more thorough and explicit approach is needed for administration of the principals (Admin, Auditor, Creator, Developer, Owner, Publisher and User) related to each of the entities distinguished: agent platforms, hosts, agents and services.

A number of issues have not been covered in this article. First, the allocation of credentials to agents. A related issue is to which extent rights can be transferred from one entity to another. For example, if users can legally delegate (part of) his/her rights to an agent. In JADE-S the rights a user obtains are automatically assigned to each of its agents.

A second point not covered in this article is procedures for recovery from a breach of integrity: The requirements with respect to liability and accountability need to be identified.

A final remark concerns the dynamic nature of agent systems: Entities (e.g. location, agents and services) come and go. The impact of this nature on the feasibility of some of the criteria mentioned in this paper needs further research.

Acknowledgements
The authors thank Stichting NLnet for their support, Martine Boonk and Anja Oskmap for their contribution to the legal analysis, Martijn Warnier and Benno Overeinder and the external reviewers for their comments on this paper.

References

1. Luck, M., McBurney, P., Shehory, O., Willmott S. and the AgentLink Community (2005), "Agent Technology: Computing as Interaction - A Roadmap for Agent-Based Computing", ISBN 085432 845 9, http://www.agentlink.org
2. "The HP Security handbook – Protecting your Business", 2005
3. Groot, D.R.A. de, Boonk, M.L., Brazier, F.M.T. and Oskamp, A. (2005), Issues in a Mobile Agent-based Multimedia Retrieval Scenario, In: Gleizes, M.P., Kaminka, G., Nowe, A., Ossowski, S., Tuyls, K. and Verbeeck, K. (editors), Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS'05), pp. 103-113
4. Poggi, A., Tomaiuolo, M., Vitaglione, G., Security and Trust in Agent-Oriented Middleware, LNCS Volume 2889, Jan 2003, Page 989
5. Bellifemine, F., Poggi, A., and Rimassa, G. (2000), "Developing Multi-agent Systems with JADE". In: C. Castelfranchi and Y. Lespérance (editors), Intelligent Agents VII. Agent Theories, Architectures, and Languages -7th International Workshop, ATAL-2000, Boston
6. JADE Board, 2005, JADE Security Add-On GUIDE, Administrator's guide of the Security add-on, Version 28-February-2005, JADE 3.3, Copyright (C) 2004, TILAB S.p.A.
7. Overeinder, B.J. and Brazier, F.M.T. (2004), Scalable Middleware Environment for Agent-Based Internet Applications, In: Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04), LNCS
8. Yip, A. and Cunningham, J. (2002), Some Issues on Agent Ownership, In: Sartor, G. and Cevenini, C. (editors), Proceedings of the workshop on the Law of Electronic Agents (LEA02)
9. Subirana, B., Bain, M.,(2005),"Legal Programming: Designing Legally Compliant RFID and Software Agent Architectures for Retail Processes and Beyond", Springer's : Integrated Series in Information Systems, Vol. 4, ISBN: 0-387-23414-4.
10. Borking, J.J., Eck, B.M.A. van en Siepel, (1999) P. "Intelligent software agents and privacy", Registratiekamer, "Achtergrondstudies en Verkenningen 13"
11. M. Wilikens (Joint Research Centre), presentation on "Privacy and Identity Management" at the PAMPAS Dissemination Workshop, Leuven, March 5, 2003.
12. Tanenbaum, A.S. and Van Steen, M. (2002), Distributed Systems: Principles and Paradigms, Prentice Hall, New Jersey.
13. G. Ballintijn and M. van Steen. "Scalable Naming in Global Middleware." In Proc. 13th Int'l Conf. on Parallel and Distributed Computing Systems, pp. 624–631, Las Vegas, Aug. 2000. ISCA
14. L. Ishitani, V. Almeida, W. Meira Jr., (2003), Masks: Bringing Anonymity and Personalization Together, IEEE Security and Privacy, vol. 01, no. 3, pp. 18-23
15. John J. Borking, Privacy Incorporated Software Agent (PISA): Proposal for Building a Privacy Guardian for the Electronic Age, LNCS Vol. 2009, Jan 2001
16. WP 15.1, 2005, "Privacy and Identity Management for Europe - PRIME White Paper", M. Hansen and H. Krasemann (ICPP)(Editors), Reviewers: P. Keller (Swisscom), M. Vanfleteren (KULeuven), http://www.prime-project.eu.org/
17. Pfleeger P. (1997), Security in Computing, international edition Prentice Hall, pp. 158, ISBN 0-13-185794-0

18. Mayfield, T., J. E. Roskos, S. R. Welke and J. M. Boone, (1991) “Integrity in Automated Information Systems”. National Computer Security Center (NCSC), TR 79-91
19. Sandhu, R. S., 1994, On Five Definitions of Data Integrity. In Proceedings of the IFIP Wg11.3 Working Conference on Database Security VII (September 12 - 15, 1993). T. F. Keefe and C. E. Landwehr, Eds. IFIP Transactions, vol. A-47. North-Holland, 257-267
20. Roth, V. (2000), Scalable and Secure Global Name Services for Mobile Agents. 6th ECOOP Workshop on Mobile Object Systems: Operating System Support, Security and Programming Languages (Cannes, France, June 2000)
21. T. Wright, 2004, Naming Services in Multi-Agent Systems: A Design for Agent-Based White Pages, AAMAS 2004: 1478-1479
22. FIPA Agent Management Specification, SC00023K, 2004
23. JADE Administrator’s Guide, update of 25-January-2005. JADE 3.3 by Fabio Bellifemine, Giovanni Caire, Tiziana Trucco (TILAB S.p.A., formerly CSELT), Giovanni Rimassa (FRAMeTech s.r.l.), Roland Mungenast (PROFACTOR GmbH)
24. Mobach, D.G.A., Overeinder, B.J., Marin, O. and Brazier, F.M.T. (2005), Lease-based Decentralized Resource Management in Open Multi-Agent Systems, In: Proceedings of the 18th International FLAIRS Conference
25. van 't Noordende, G., Brazier, F.M.T. and Tanenbaum, A.S. (2004), Security in a Mobile Agent System, In: Proceedings of the First IEEE Symposium on Multi-Agent Security and Survivability
26. Karnik, N. and Tripathi, A., (2001), “Security in the Ajanta Mobile Agent System”. Software - Practice and Experience 31(4), pp. 301-329