Issues in a Mobile Agent-based Multimedia Retrieval Scenario

Boonk, M.L.; de Groot, D.R.A.; Oskamp, A.; Brazier, F.M.

published in
Proceedings of the 4th Workshop on the Law and Electronic Agents
2005

document version
Publisher's PDF, also known as Version of record

Issues in a Mobile Agent-based Multimedia Retrieval Scenario

D.R.A. de Groot[1], M.L. Boonk[2], F.M.T. Brazier[1] and A. Oskamp[2]

[1] IIDS Group, Faculty of Sciences, Vrije Universiteit Amsterdam,
de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Phone: +31 - 20 - 5987434; Fax: +31 - 20 - 5987653
{frances, davidra}@cs.vu.nl
http://www.iids.org/
[2] Computer/Law Institute, Faculty of Law, Vrije Universiteit Amsterdam,
de Boelelaan 1105, 1181 HV, Amsterdam, The Netherlands
Phone: +31 - 20 - 598 6215; Fax: +31 - 20 - 598 6230
a.oskamp@rechten.vu.nl
m.boonk@rechten.vu.nl
http://www.rechten.vu.nl/~CLI

Abstract. Mobile agents traverse the Internet, often on behalf of their users. Intelligent search agents access dynamic information in heterogeneous environments. The legal implications of the use of agents in such situations are not fully understood. In this paper a scenario in which a mobile agent searches a multimedia database on behalf of its user, is used to illustrate the legal and technical issues involved. Requirements related to identity management, integrity, traceability and availability are identified and discussed in the context of existing technology.

1 Introduction

Mobile agents traverse the Internet, moving to different sites with different characteristics. Mobile intelligent search agents access information in heterogeneous, often dynamic, environments. The legal implications of the use of agents in such situations are not fully understood. This paper discusses a number of issues related to identity management, integrity, traceability and availability, continuing the research done within the context of the ALIAS project, in which legal implications of the use of agent systems are investigated from both a legal and a technological perspective [1]. A scenario in which mobile search agents searches a multimedia database on behalf of its user, is used to illustrate the legal and technical issues involved, identifying a number of the requirements involved in the context of existing technology.

Klusch [4] uses the term “intelligent information agents” instead of intelligent search agents and defines them to be “autonomous computational software entities that are especially meant to (1) provide a proactive resource discovery, (2) resolve information impedance of information consumers and providers, and (3) offer value-added information services and products.” This definition suffices to define the types of agents to which this paper refers.

Software agents themselves are assumed to have the following properties [2]: (1) autonomy (they have control over their own actions and state); (2) social ability (they can communicate with other agents); (3) reactivity (they react to changes in its environment); (4) pro-activeness (they make plans to reach their goals and can take initiative to pursue these). Additionally, agents are assumed to be mobile, that is, they can move between network-connected computers. Mobility has many potential benefits, which can be grouped into three categories: performance, resource access and security [3], as discussed in section 2 in the context of a specific domain.

It is beyond the scope of this paper to provide an overview of all related research: literature on intelligent search agents[1] and personal assistant agents[2] clearly influences this work as does research on agent platforms and mobile agent security issues (e.g. Borselius [5], Cubillos [6] and Bellavista [7]). Protection of intellectual property[3] briefly addressed in this paper is discussed more extensively in the literature.

2 An intelligent search agent scenario

In our scenario, mobile agents are used to access information across the internet: information stored in a remote multimedia database. Users interact with a personal search service designed to help them find snippets of movies in multi-media databases. To this purpose, a user interacts with the user interface on his/her computer or PDA. The user interface in turn, assigns each agent a specific search task together with an address: the address of the location to which the agent is to migrate. An example search task could be to find a snippet of a movie picturing a scene of a dog riding a bike.

Migration of agents has several advantages. First of all, it reduces the amount of data sent over the Internet [8]. Secondly, a continuous, reliable high-bandwidth connection with the multimedia database is not needed. Thirdly, it provides the multimedia database service providers the option to exercise control over data returned to the users of mobile agents, e.g. to prevent possible misuse of the data. Especially when information is sensitive to reproduction, as is the case with multimedia files in a multimedia database, there is a high risk of infringement to the copyright holder's ability to exercise copyrights on the items in the database.

In this scenario the multi-media database service provider is assumed to be trusted and known to the agents. Agents are also assumed to have appropriate credentials for access, e.g. have a login or signed certificates. An individual agent contacts the remote site to which it wishes to migrate. If the remote location is willing to host the agent and is willing and capable of providing the necessary resources (i.e., in this case, access to the multimedia database), the agent migrates to the remote location to search the database. The search process itself is not addressed in this paper.

Once the search task has been completed, this scenario assumes that the agent contacts a guardian agent provided by the hosting location (as proposed by Noordende [9]). The guardian agent notifies the user and provides low-quality streams of the video snippets for pre-view purposes. The user is given the opportunity to accept one or more of the results, or to provide its agent with feedback on the results. If necessary, the user's agent can continue the search task based on this feedback. Once the user has received the information he/she was looking for, a high quality download or shipment can be arranged along with payment options.

The scenario as described above can be depicted as follows, see also Figure 1. The desired information cannot be found in the user's domain, but is available at a remote location. The personal search service runs at one location, the multimedia database access service at another. Both locations have the necessary middleware, i.e. agent platform software, to host agents and services, migration and communication. In Figure 1 the remote location is depicted as the service provider. At the physical level a network connection, e.g. Internet or a mobile phone network, enables communication between the agent platform locations.

---

1 See for example [4] and Volker Roth, Jan Peters, and Ulrich Pinsdorf. A distributed content-based search engine based on mobile code. In Proc. 20th ACM Symposium on Applied Computing, Special Track on Agents, Interactions, Mobility, and Systems (SAC/AIMS), Santa Fe, NM, USA, March 2005.

2 For example: L. Chen and K. Sycara, 1998, WebMate: A Personal Agent for Browsing and Searching, Proceedings of the 2nd International Conference on Autonomous Agents and MultiAgent Systems, AGENTS '98, ACM, May, 1998, pp. 132 – 139.

3 See for example Sonntag, M., Legal aspects of mobile agents. With special consideration of the proposed Austrian E-Commerce Law. In: Robert Trappl (Ed.): Cybernetics and Systems 2002. Proc. of the 16th European Meeting on Cybernetics and Systems Research. Wien: Austrian Society for Cybernetic Studies 2002, 153-158 and Belmon, S. G., and Yee, B. S. Mobile agents and intellectual property protection. In Rothermel and Hohl,(Eds.) Proceedings of the Second International Workshop on Mobile Agents (MA '98), vol. 1477 of Lecture Notes in Computer Science. Springer Verlag, Berlin Heidelberg, September 1998. pp. 172-182.

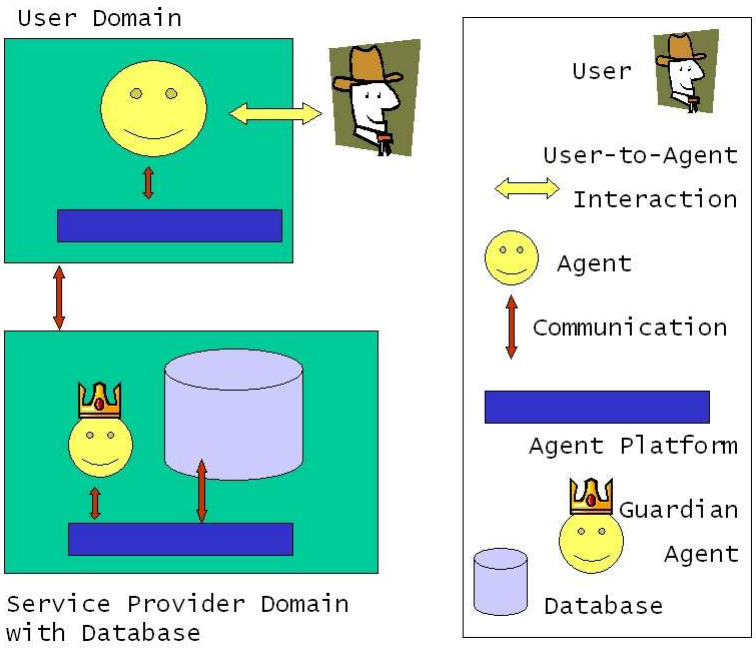

Figure 1: User and Service Provider

3 Legal and security issues

Before an agent can start searching the multimedia database, it will first try to negotiate an agreement with the agent platform that offers access to the multimedia database service. The platform and the visiting agent must use the same protocols to negotiate. An agreement is needed in which both the facilities provided by the platform (both resources and services) and the procedures on the platform (e.g. whether the agent can communicate with its user while the agent is on the platform and whether the agent's process is terminated after it has finished its search) are defined. In addition to the above agreement, an agent may want guarantees on the level of performance offered by an agent platform. If this is the case, a Service Level Agreement (SLA) can be agreed upon (more on SLAs follows later in this section).

As shown in Figure 2, there are four types of entities involved in an agent's search in the multimedia database: the user, the search agent, the owner and the agent platform. An agent represents the user and the platform owner is represented by the platform (in the scenario we assume the service provider to be the platform owner but that is not necessarily the case). To ensure correct functioning of the service, the identity of each agent needs to be distinguished. Identity management is elaborated upon in the next section. For a reliable service, the integrity of both the agent and the agent platform needs to be guaranteed. Integrity is dealt with in 3.2. From the moment the agent contacts the platform to negotiate, logs are kept of the activities engaged in by every agent on the platform. Issues of logging and tracing are addressed in section 3.3. Finally, availability of the platform is an essential part of the performance offered to the agent, which is covered in 3.4.

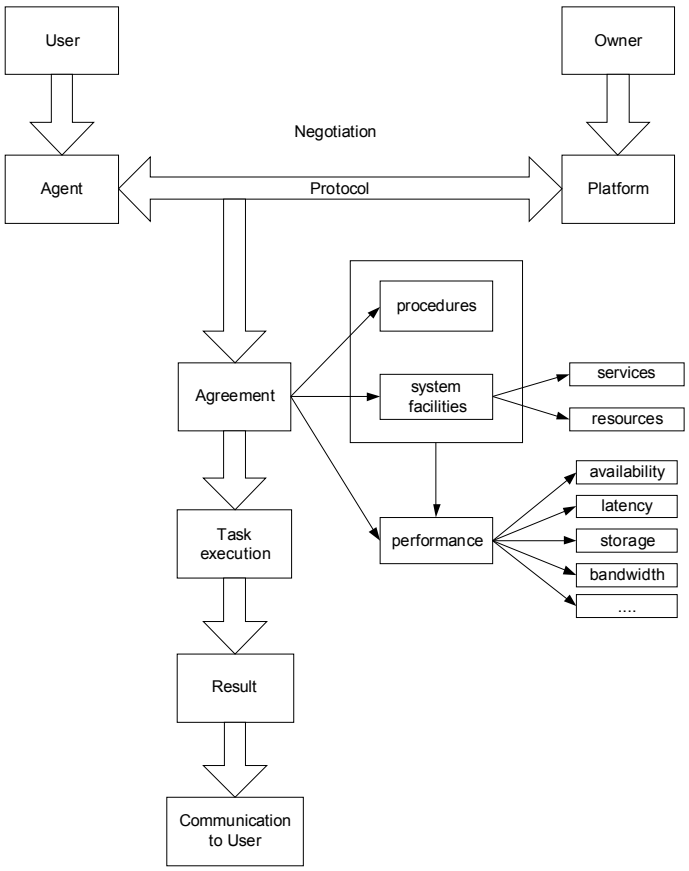

Figure 2: Model of service provision

3.1 Identity management

Identity management in agent systems is important for various reasons. First of all for the purpose of administrating which entities are in the system.

Secondly, identity management is needed for the application of access policies to determine who is allowed to do what and with which credentials. Based on this information, access policies may be enforced, e.g. a migrating agent may need to provide details and proof of its affiliation to gain access to a specific location. Other information about agents may also need to be administrated, such as the agent owner, user, company and programmer.

A service provider may also have special contracts with important clients, e.g. a large television company, determining that these clients' agents are to be run on specific dedicated machines with a guaranteed quality of service and security for example.

Thirdly, identity management is needed for logging and tracing purposes. Suppose two agents in the scenario have the same identifier. If something goes wrong, it cannot be determined which one of the agents caused the problem. Communication and actions that are logged should be relatable to the actor and to the party who is responsible for the actor's behaviour.

Requirement 1: Agents need to have unique identifiers.

A technical solution for registration of agent identities is the use of both locally and globally unique identifiers (LUIDs and GUIDs). GUIDs can be registered at the moment an agent is created; an agent's

identity can be coupled to the identity of a user. An example of an approach to agent identity administration is described by Roth [10]. In addition one or more LUIDs can be used in the course of an agent's life.

Note that there is an apparent discrepancy between a possible wish for anonymity of an agent user on the one hand and the service provider's need to know the exact identity of the user of the visiting agent on the other hand [11, 12]. For example, an agent user buying movies wants his/her agent to search the multimedia database without others knowing what (kind of) movie(s) the agent is requested to search for. On the other hand, when an error has occurred, e.g. due to a malfunctioning agent, the service provider may need to trace which events happened and (to determine liability) whose agent was responsible. Thus, a balance between the user's need for anonymity and the interests of the service provider of the multimedia database is needed. In our scenario of an agent searching for movie snippets requested by his/her user, this implies that different levels of knowledge on the identity of both agent and user may be required in different phases of the agent's activities. Four different levels of knowledge on identities can be distinguished: untraceable anonymity, traceable anonymity, untraceable pseudonymity and traceable pseudonymity [1]. When searching the multimedia database, an agent does not need to reveal its user's identity, traceable pseudonymity would be sufficient. However, when the searching results in the buying of a movie, a user's true identity may be needed to complete the transaction (e.g. for credit card payment, submitting a cardholders' name is required).

Requirement 2: Agent and user identities need to be managed.

Technical solutions entail the introduction of local names (LUIDs) and multi-phased identity management. Local names can function as a pseudonym for a specific agent in a specific environment. For example, an agent's GUID may only be known its user and to a Trusted Third Party (TTP). The TTP can be used for the administration of local names and the mapping on agent GUIDs. When certain conditions are met, the TTP could provide the GUID that is connected to a local name, reveal a user's identity or show links between local names. Agents can also be assigned a local identity within the location at which they reside and with which they are registered and addressed.

Managing levels of knowledge regarding identities in different phases, termed multi-phased identity management is needed. How this should be done and who should be in charge of the administration is still subject of research.

For fault-tolerance and performance reasons, an agent can be cloned or copied. Or agents can spawn child- or helper-agents to search in different databases in parallel. Different approaches have currently been implemented: more research is needed.

3.2 Integrity

To ensure correct functioning of a multimedia database, the integrity of both an agent platform and visiting agents is essential. In other words, it is crucial that the data and transmissions are not "unduly altered, erased or supplemented and that the physical objects involved (…) are not damaged or destroyed."[4]

Protection of entities

Current research in this matter concentrates on prevention of attacks on hosts of agent platforms [7]. However, an agent platform could also modify the code of an agent, e.g. by changing a variable and thus having the agent recommend ten movies instead of five. Furthermore, agents can attack each other. For example, in our scenario our user agent could attack the guardian agent of the service provider. If the service provider disenables direct communication between a user and its agent, a user has no means to verify whether its agent or its messages have been tampered with, until the agent leaves the remote location. Thus, as Yee points out as well, protection and detection measures to safeguard the integrity of the agent and its computation are needed [13].

Requirement 3: Protection and detection measures are needed to safeguard the integrity of the agent and agent platform.

4 The definition of integrity we use is cited from [1], p. 53 below.

Both Borselius [5] and Cubillos [6] give an overview of protection measures for agent platforms and agents. Example measures for platform protection are sandboxing or jailing of agents and the use of signed code [9, 14, 15]. Note that code signing is also useful for agent protection and detection of changes. Other measures for agent protection described include using trusted nodes, execution tracing [16] and encryption of code and functions [17]. Examples of other approaches are watermarking and fingerprinting of agents [18], code obfuscation [19], re-execution of agents [20], semantic encryption [21], and integrity based encryption [22].

Protection of data

A mobile agent carries data, which may be relatable to its user, e.g. the search request of the user, passwords and usernames, etc. Since an agent completely relies on the benevolence of an agent platform, confiscation and modification of these data is obviously undesirable. In fact, it may be hard for agents to keep information secret from its hosting agent platform [23].

Several types of data possessed by an agent can be distinguished, for example: assignment (search request), credentials, internal logs, and gathered information. All of these data need protection and appropriate detection measures to safeguard their integrity.

Requirement 4: Protection and detection measures are needed to safeguard agent data.

Measures for agent protection, as listed above, can often be used to protect the agent data as well. Another technique for protecting the agent state (and detecting changes) is introduced in Ajanta: signed append-only containers for agent data [14]. This approach is followed in Mansion [9] and AgentScape [15]. Examples of other protection techniques are partial result protection [24] and detecting attempts of tampering [25].

Additionally, protection is needed for the data located in the domain of service provider, e.g. logs and identity administration of the agent platform and contents of the multimedia database. For example, in the scenario, a mobile agent interacts with the multimedia database and processes a number of items. Anyone of the items could be confiscated or communicated (e.g. via covert channels) to a third party. This is obviously undesirable and should be prevented.

The multimedia database and its content are protected by intellectual property regulations. In the EU, if a multimedia database matches certain requirements, it may be protected under database law. If that is the case, searching the multimedia database requires the rightholders' consent. Also, the contents of a multimedia database are likely to be protected by copyright. Not only the copying of an entire copyrighted movie, but also the copying of parts of it could entail copyright infringement. Movie snippets can therefore not be extracted without the rightholders' consent. If the service provider is not the copyrights holder, he/she will need permission from the copyright holder to exploit the database and permit others to search the copyrighted items. If the agent copies items from the database without the rightholder's consent, the agent infringes copyright, for which its user is responsible.

The copyright holder may want to keep as much control on the giving out of snippets as possible, in an effort to prevent future copyright infringement.

Requirement 5: Protection and detection measures are needed to safeguard the contents of the multimedia database.

An option to safeguard content data is to inspect an agent before it leaves a platform and to disenable communication. Another option is to agree in advance that an agent will terminate after handing a message transferring its desired items (or their identifiers) to a guardian agent. Noordende suggests the latter approach [9].

Protection of transmissions

Communications between entities in agent platforms need to be secured, e.g. to prevent reading or re-play by third parties. Communication needs to be secured if it takes place over unsafe or unreliable network connections. Depending on the application and architecture of the platform, agent-to-agent communication should be secured. Realisation of secured agent-to-agent communication may be problematic if the agent platform cannot be trusted. If agents cannot have secrets, they may also not be able to keep a private key since it could be captured.

Requirement 6: Protection and detection measures are needed to safeguard the integrity of the transmissions.

3.3 Logs and traceability

Logging communication and actions of entities in the system is useful for reconstructing situations after they have occurred. This can be particularly of use if damage has been incurred. Firstly, for example, if and when a content provider discovers that a particular snippet from his multimedia database is circulating among Internet users, the content provider may want to know who was responsible for distribution of copies of this snippet without authorisation. To find out who was responsible, the content provider will both need to have logged which agent accessed which movies and identify the user of this particular agent. Secondly, logging is needed to trace errors in the system. If the platform logs show that one of the visiting agents is overusing the service, e.g. because it is consuming the far more resources while searching the multimedia database than it is allowed to, the particular agent should be found and perhaps even killed.[5] Thirdly, logs can be used in the process of determining liability, e.g. when a denial of service has occurred which has caused much damage and the logs show that a specific agent was responsible. Fourthly, logs can be used to determine where and when an agent has been tampered with.

Requirement 7: Logging of communication and actions of entities is necessary.

Because of the distributed nature of the system, there is no single solution for all situations. Who should log what, where and how? If logging is done locally, and/or by agents, information could be lost if a node or an agent fails. Updating information is complex and intricate, certainly when malicious hosts are possible. Notions and models of trust play a role in the update policies for distributed logging. Further research in this area is needed.

Requirement 8: Logs must be robust and should not disappear because of a failing node.

Among the data that is logged, there may be personal data that is reducible to the agent's user. If this is the case, privacy regulations play a vital role. For example, gathering information on a personal agent's searching habits could amount to a profile containing personal data of the agent's user. In the EU, any processing of personal data is subject to detailed regulations.[6]

Requirement 9: User privacy needs to be respected in accordance with privacy regulations.

In most cases, a user explicitly has to consent to the processing of any personal data. Therefore, an agent platform must either anonymise any user data to the extent that the data is not reducible to an individual user, or the user has to explicitly consent in use of personal data before the data is processed. In certain circumstances, anonymisation of logged data may conflict with the service provider's need for traceability. On the other hand, when explicit consent of the user is needed, the question arises whether it is possible for an agent to consent to the use of personal data of its user.

Additionally, both logs and the identity administration need to be safeguarded e.g. to prevent agents from maliciously changing logs. Unwanted access to the logs and other administration by users and agents should be prevented, also with regard to privacy issues as described above. It needs to be determined who is allowed to access which logging data and under which circumstances.

Requirement 10: Unauthorized access to the logs and administrative data should be prevented.

---

5 Some issues which need to be considered in deciding whether or not to kill a mobile agent are discussed in: Apistola, M., Brazier, F.M.T., Kubbe, O., Oskamp, A., Prins, J.E.J., Schellekens, M.H.M. and Voulon, M.B. (2002), Migrating agents: Do sysadmins have a license to kill? In: Proceedings of the 3rd International SANE Conference (SANE 2002).

6 Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 regards any data that can be traced to an individual as personal data.

A technical solution is to install access control policies to prevent unwanted access to the logs and administrative data. For logs, another option is to dynamically generate "views" or presentations on the logs, which depend on the credentials of the viewer, e.g. data that may identify a person can be filtered out.

3.4 Availability

To maintain its reputation as a reliable platform, an agent platform tries to limit interruption or disruption of its service (a user may have subscribed to a specific service with a predefined quality of service, at a given cost). As a result, service and system availability is expected.

Requirement 11: Service and system availability need to be guaranteed.

To prevent unavailability, a platform should estimate the available resources, monitor the usage of resources and manage them. Example technical solutions to manage resources entail (automated) agreements on the use of system resources. To regulate the amount of resources agents are using, a platform can use a lease system for resource usage by agents [26].
An essential element of performance is availability. Availability is a hard requirement for an agent to be able to successfully complete its search in the multimedia database. The level of performance provided to the visiting agent, including availability, can be agreed upon in so-called Service Level agreements (SLAs) [27, 28, 29].

Requirement 12: Clear and legally valid agreements are needed.

Whether an SLA that has been agreed upon by an agent and an agent platform can be regarded as a legally valid contract, is uncertain. To establish a legally valid contract, two persons have to perform corresponding acts of offer and acceptance.[7] Since neither agents, nor agent platforms can as yet be considered to be legal entities, the offer and acceptance by an agent and an agent platform must be reduced to both the agent's user and the platform owner. Without two legal entities having the will to establish certain legal consequences, there can be no valid contract.

However, if an agent's user gives an agent a specific assignment (in our scenario: to find and buy a specific movie) and also a specific amount of e-cash the agent is allowed to spend, the question arises whether the user's will was not only directed at closing a contract concerning the purchase of that particular movie, but also at establishing an "underlying" contract between the agent and the platform. If this is the case, there may be a valid contract between the agent user and the platform owner.[8]

Even if an SLA is not regarded as a legally valid contract, it can indeed have legal implications, for example when a lease ends while the agent has not yet finished its tasks on the platform. If an agent is unable to finish its search for movies, the agent user will not receive any movie snippets, and as a consequence may suffer damages.

In the above example of a specific assignment leading to a valid contract between the agent user and the platform owner, the agent's level of autonomy may be at stake. For the agreement to be legally valid, the level of autonomy of the agent may need to be decreased substantially, whereas at the same time a higher level of autonomy of the agent may be desirable for the agent to fulfil its search task properly.

Also, if an agent is to act according to its user's will, a user needs to be able to exercise direct control over the agent at any time. The possibility of communication between the agent and its user has to be guaranteed. This has consequences for the role of the guardian agent in the scenario. To ensure the possibility of direct communication between an agent and its user at any time, technical measures are needed.

Further research in this area is needed, both to exactly determine the circumstances under which an SLA can be regarded as a valid contract, and to maintain a balance between the agents' autonomy and legal validity of the agreements it concludes.

---

7 See for an elaborate analysis Weitzenboeck, E.M., "Electronic agents and the formation of contracts", International Journal of Law and Information Technology, Vol. 9 Issue 3, autumn 2001, Oxford University Press, ISSN 0967-0769, pp. 204-234.

8 See Sartor, G. (2002) Intentional concepts and the legal discipline of software agents. In J. Pitt (Ed.), Open Agent Societies: Normative Specifications in Multi-Agent Systems. See also Sartor, G. Agents in Cyberlaw In: Proceedings of the workshop on the Law of Electronic Agents (LEA02), 2002, Sartor, G. Cevenini, C. See also [1]

4 Technical considerations

An agent platform is middleware (a software layer between the operating system and the application programs) that provides an execution environment for agents. Often, agent platforms offer facilities and services to agents on the platform, for example facilities for communication and life-cycle support (starting, pausing, resuming, deleting agents) and services like a Directory Service (White and Yellow Pages to find agents and services). In addition, agent platforms can offer support for migration and security. Examples of agent platforms are AgentScape [15], SeMoa[9], JADE[10] and Cougaar. A short description of JADE and Cougaar and a comparison with AgentScape is given by Overeinder [15].

The scenario as proposed above can be implemented in the AgentScape agent platform. AgentScape provides the necessary mapping of locations (user and service provider each have their own location) and support for multiple hosts within one location. Migration of agents between locations is supported and user locations can be dynamically connected and removed. The Web Service Gateway (WSG) for Internet and database interactions is another feature that makes AgentScape particularly suitable for this scenario.

Various security features, which relate to requirements mentioned above, have been implemented in AgentScape. The use of global and local identities, leasing of resources, sandboxing of agents, signing agent's code and its state, and secure communication are the most prominent.

Within AgentScape agents have a globally unique identifier, and a locally unique identifier within each location at which they reside. The use of leases for resource access and monitoring is a means to provide availability. To prevent unwanted interactions by agents with other entities, e.g. to protect the host on which the agent is running, agents are 'sandboxed' or 'jailed'.

For agent and agent state protection, e.g. to guarantee integrity of the agent's code and data, an agent and its data are stored in an Agent Container. The AgentScape platform implements an integrity verification mechanism based on signing of the Agent Containers. This fulfils the requirement concerning protection and detection measures to safeguard the integrity of the agent.

Communication between hosts in AgentScape, e.g. for the purpose of agent migration, has been secured. Mutually authenticated, encrypted channels are set up using a key exchange protocol at the host-to-host level. Thus, the requirement of protection of transmission is met.

Efforts to improve AgentScape and its security are still ongoing – requirements with respect to logging for example, still need to be addressed. Mechanisms for logging are available, but more research on applicable policies is needed.

5 Summary and future research

This article describes a scenario in which a mobile agent moves to the location of the service provider and searches a multimedia database on behalf of its user. In this scenario, an intermediary (the guardian agent) is used to communicate the search results to the user and take care of final arrangements. In the context of the scenario, a number of security and related issues have been discussed. In the above discussion, the following points for further research can be identified.

Firstly, there is the issue of logging in distributed mobile agent environments. Protocols need to be developed to propagate updates according to specific policies. These issues appear to be related to update propagation in distributed object systems. The theories and techniques developed for object systems may be applicable to agent systems as well, though complicated by agent mobility.

Secondly, how to realize identity management in mobile agent systems needs further research. Facilities are needed for identity administration. Entities, e.g. users and agents, should to be able to operate anonymously or pseudo-anonymously. However, under certain circumstances, information may be needed with respect to the identity of the user, therefore information on the identities has to be managed and be accessible for entities with appropriate credentials. Another open issue regards identity management of possible agent's clones, children and helpers.

Thirdly, further research is needed in mobile agent and security issues, e.g. agents carrying and communicating confidential information. In case a mobile agent visits a variety of service providers, it may

9 http://www.semoa.org/
10 http://jade.tilab.com/

need to keep secret or prove possession of certain information, e.g. passwords and private keys, to others than its user. Protocols are needed to check the credentials or certificates of a migratory agent.

An example of an agent platform within which a number of these issues are addressed has been briefly discussed describing the technical feasibility of a number of the suggested solutions. It is clear that further research is needed.

Acknowledgements

The authors thank the Vrije Universiteit and Stichting NLnet for their support.

References

1. Brazier, F.M.T., Oskamp, A., Prins, J.E.J., Schellekens, M.H.M., Schreuders, E., Wijngaards, N.J.E., Apistola, M., Voulon, M.B. and Kubbe, O., (2003), "ALIAS: Analysing Legal Implications and Agent Information Systems", Technical Report no. IR-CS-004, Computer Science, Faculty of Sciences, Vrije Universiteit Amsterdam, available from http://www.iids.org/
2. Wooldridge, M. and Jennings, N.R. (1995), "Intelligent agents: Theory and practice", Knowledge Engineering Review, 10(2), pp. 115-152.
3. Brazier, F.M.T., Overeinder, B.J., Steen, M. van, and Wijngaards, N.J.E. (2002) "Agent Factory: Generative Migration of Mobile Agents in Heterogeneous Environments" In: Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002), pp. 101-106.
4. Klusch, M., (2001) "Information agent technology for the Internet: A Survey. In: Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, volume 36:6. D. Fensel (Ed.), Elsevier Science, 2001.
5. Borselius, N., (2002) "Mobile agent security", Electronics and Communication Engineering Journal, IEE Press, Vol. 14, No. 5, pp 211-218.
6. Claudio Cubillos F. and Franco Guidi-Polanco, Security Issues on Agent-Based Technologies, VIP Scientific Forum of the International IPSI-2003 Conference, Sveti Stefan, Montenegro, Former Yugoslavia.
7. P. Bellavista, A. Corradi, C. Federici, R. Montanari, D. Tibaldi, "Security for Mobile Agents: Issues and Challenges" Invited Chapter in the Book "Handbook of Mobile Computing", I. Mahgoub, M. Ilyas (eds.), ISBN 0-84931-971-4, CRC Press, Dec. 2004.
8. Lange, D.B. and Oshima, M. (1999), Seven Good Reasons for Mobile Agents, Communications of the ACM, March, 42(3):88-89.
9. 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.
10. Roth, V., 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).
11. Brazier, F.M.T. Oskamp, A. Prins, J.E.J. Schellekens, M.H.M. Wijngaards, N.J.E. (2004) Law-Abiding & Integrity on the Internet: a Case for Agents In: AI & Law, p. 24.
12. Brazier, F.M.T. Oskamp, A. Prins, J.E.J. Schellekens, M.H.M. Wijngaards, N.J.E. (2004) Anonymity and Software Agents: An Interdiscplinary Challenge In: AI & Law, p. 15.
13. Yee, B.S., A sanctuary for mobile agents. In: Proceedings of the DARPA workshop on foundations for secure mobile code, Monterey CA, USA, March 1997.
14. Karnik, N. and Tripathi, A., (2001), Security in the Ajanta Mobile Agent System. Software - Practice and Experience 31(4), pp. 301-329.
15. 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), Lecture Notes in Computer Science.
16. G. Vigna, Cryptographic Traces for Mobile Agents. Mobile Agents and Security 137-153 LNCS 1419, Springer-Verlag June 1998.
17. Sander, T and Tschudin, C.F., Protecting Mobile Agents Against Malicious Hosts, Lecture Notes in Computer Science, Volume 1419, Jan 1998, Page 44-60.

18. Oscar Esparza, Marcel Fernandez, Miguel Soriano, Jose L. Munoz, Jordi Forné, Mobile Agent Watermarking and Fingerprinting: Tracing Malicious Hosts, Lecture Notes in Computer Science, Volume 2736, Sep 2003, Pages 927 – 936.
19. Ogiso, T., Sakabe, Y., Soshi, M., Miyaji, A., Software Obfuscation on a Theoretical Basis and Its Implementation, 2003, IEICE Transactions on Fundamentals E86-A (2003) 176-186.
20. Kwai-Ki Leung, Kam-Wing Ng, Detection of Malicious Host Attacks by Tracing with Randomly Selected Hosts, Lecture Notes in Computer Science, Volume 3207, Jul 2004, Pages 839 – 848
21. W. Thompson, A. Yasinsac, J. McDonald. "Semantic Encryption Transformation Scheme," in Proc. of 2004 International Workshop on Security in Parallel and Distributed Systems, San Francisco, CA, September 15-17, 2004.
22. Jaewon Lee, Heeyoul Kim, Hyunsoo Yoon, Tamper Resistant Software by Integrity-Based Encryption, Lecture Notes in Computer Science, Volume 3320, Dec 2004, Pages 608 – 612.
23. G. Vigna, "Mobile Agents: Ten Reasons For Failure," Proceedings of MDM 2004, pp. 298-299 Berkeley, CA January 2004.
24. T. McDonald, A. Yasinsac, W. Thompson, Mobile Agent Data Integrity Using Multi-agent Architecture, International Workshop on Security in Parallel and Distributed Systems, San Francisco, 2004.
25. P. Maggi and R. Sisto, "A Configurable Mobile Agent Data Protection Protocol," in Proc. of the 2nd Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03), 2003.
26. 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.
27. T. J. Norman, N. R. Jennings, P. Faratin, and E. H. Mamdani, Designing and implementing a multi-agent architecture for business process management. In Intelligent Agents III (eds. J. P. Mueller, M. J. Wooldridge and N. R. Jennings) LNAI 1193, Springer Verlag, 261-275, 1997.
28. H. Kneer, H. Stormer, H. Häuschen, B. Stiller, 2002, An Agent-based Framework for Monitoring Service Contracts. Proceedings of the Third International Conference on Electronic Commerce and Web Technologies (EC-Web 2002), Aix-en-Provence, France, September 2-6, 2002.
29. Gilles Klein, Francine Krief, Mobile Agents for Dynamic SLA Negotiation, Lecture Notes in Computer Science, Volume 2881, Oct 2003, Pages 23 – 31.