Such Aparicio, JM.; García Fornes, AM.; Espinosa Minguet, AR.; Bellver Faus, J. (2013). Magentix2: a Privacy-enhancing Agent Platform. Engineering Applications of Artificial Intelligence. 26(1):96-109. doi:10.1016/j.engappai.2012.06.009.

# Magentix2: a Privacy-enhancing Agent Platform

Jose M. Such, Ana García-Fornes, Agustín Espinosa, Joan Bellver

*Departament de Sistemes Informàtics i Computació*
*Universitat Politècnica de València, Camí de Vera s/n, València, Spain*
*{jsuch,agarcia,aespinos,jbellver}@dsic.upv.es*

**Abstract**

Agent Platforms are the software that supports the development and execution of Multi-agent Systems. There are many Agent Platforms developed by the agent community, but they hardly consider privacy. This leads to agent-based applications that invade users' privacy. Privacy can be threatened by two main information activities: information collection and information processing. Information collection can be prevented using traditional security mechanisms. Information processing can be prevented by minimizing data identifiability, i.e., the degree by which personal information can be directly attributed to a particular individual. However, minimizing data identifiability may directly affect other crucial issues in Multi-agent Systems, such as accountability, trust, and reputation. In this paper, we present the support that the Magentix2 Agent Platform provides for preserving privacy. Specifically, It provides mechanisms to avoid information collection and information processing when they are not desired. Moreover, Magentix2 provides these mechanisms without compromising accountability, trust, and reputation. We also provide in this paper an application built on top of Magentix2 that exploits its support for preserving privacy. Finally, we provide an extensive evaluation of the support that Magentix2 provides for preserving privacy based on that application. We specifically test whether or not privacy loss can be minimized by using the support that Magentix2 provides, whether or not this support introduces a bearable performance overhead, and whether or not existing trust and reputation models can be implemented on top of Magentix2.

*Keywords:* Privacy, Agent Platforms, Multi-agent Systems, Security, Trust, Reputation

## 1. Introduction

A Multiagent System (MAS) consists of a number of agents that interact with one-another [1]. MAS represents a key issue, especially from the development point of view in Distributed Artificial Intelligence (DAI). This is because the MAS community has produced both methodologies and actual frameworks to make the implementation of agent-based applications possible. In particular, Agent Platforms (APs) are the software that supports the development and execution of MAS. APs provide all the basic infrastructure (for message handling, tracing and monitoring, run-time management, and so on) required to create MAS [1].

There are many APs developed by the MAS community – for an overview of current APs and the features they provide refer to [2]. However, privacy is seldom considered [3, 4]. This leads to agent-based applications that invade individuals' privacy. This is due to the fact that an agent usually encapsulates personal information describing its principal[1] [5], such as preferences, names, and other informa-

☆NOTICE: this is the author's version of a work that was accepted for publication in Engineering Applications of Artificial Intelligence. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published: Jose M. Such, Ana Garcia-Fornes, Agustin Espinosa and Joan Bellver. Magentix2: a Privacy-enhancing Agent Platform. Engineering Applications of Artificial Intelligence, Vol. 26 N. 1 pp. 96-109 (2013). `http://www.sciencedirect.com/science/article/pii/S0952197612001522`

[1] In this paper, we use the terms principal and user indistinctly to refer to the user that the agent is acting on behalf

tion. Moreover, agents carry out interactions on behalf of their principals so that they exchange personal information. For instance, agents act on behalf of their principals in agent-mediated e-commerce [6], as personal assistants [7], in virtual worlds like Second Life[2] [8], as recommenders [9], and so on.

The modern conception of privacy started more than a hundred years ago, with the seminal work of Warren & Brandeis [10] *The right of privacy*. These two lawyers defined privacy as "the right to be let alone". They were pioneers in considering the implications of technology in privacy. Specifically, they were very concerned about the implications of instantaneous photographs and portraits in injuring the feelings of the people in those photographs and portraits. Privacy was later recognized as a fundamental human right by the United Nations Declaration of Human Rights, the International Covenant on Civil and Political Rights, the Charter of Fundamental Rights of the European Union, and many other international treaties [11].

In the second part of the twentieth century, Alan Westin defined privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated" [12]. This is what is currently known as the informational self-determination right [13]. The concept of informational self-determination changed the right to privacy from the right to be let alone to its current incarnation as a means to limit the abuse of personal data [14]. Informational self-determination represents today's European understanding and regulation of privacy in the context of information and communication technology (EU Directives 95/46/EC, 45/2001/EC, and 2002/58/EC).

Despite all these regulations, as the Internet has no governing or regulating body, privacy breaches are still possible. Nowadays, in the era of global connectivity (everything is inter-connected anytime and everywhere) with more than 2 billion world-wide users with connection to the Internet as of 2011[3], privacy is of great concern. In the real world, everyone decides (at least implicitly) what to tell other people about themselves. In the digital world, users have more or less lost effective control over their personal data. Users are therefore exposed to constant personal data collection and processing without even being aware of it [15]. Garfinkel [16] suggests that nowadays users have only one option to preserve their privacy: becoming hermits and not using online social networks, e-commerce sites, etc. Considering the increasing power and sophistication of computer applications that offer many advantages to individuals, becoming a hermit may not really be an option. However, all of these advantages come at a significant loss of privacy [17]. Recent studies show that 90% of users are concerned or very concerned about privacy [18]. Moreover, almost 95% of web users admitted they have declined to provide personal information to web sites at one time or another when asked [19].

In this paper, we describe the support that the Magentix2[4] AP provides for preserving privacy. The remainder of this paper is organized as follows. Section 2 introduces the main concepts treated in this article. Section 3 gives a brief overview of the Magentix2 AP. Section 4 presents the support that Magentix2 provides for avoiding information processing. Section 5 presents the support that Magentix2 provides for avoiding information collection. Section 6 presents an application that takes advantage of the support for preserving privacy that Magentix2 provides. Section 7 presents the evaluation we carried out. Section 8 presents related relevant works. Finally, Section 9 presents some concluding remarks and future work.

## 2. Background

In this paper we consider two information-related activities that can represent a major threat for privacy: information collection and information processing [13, 4]. These activities can lead to many privacy breaches [20]. We now introduce both activities and outline how these activities can be prevented when they are not desired. We also detail the implications that preventing these activities may have in accountability, trust, and reputation.

### *2.1. Information Collection*

Information collection refers to the process of gathering and storing data about an individual. Personal data is transferred on-line even across the Internet.

of. Principals are also called agent owners, or simply users in the related literature.

[2] `http://secondlife.com/`

[3] `http://www.internetworldstats.com/stats.htm` to consult updated statistics on world Internet users and population.

[4] `http://magentix2.gti-ia.upv.es`

Without appropriate protection mechanisms a potential attacker could easily obtain information about principals without their consent. For instance, an attacker can be listening to transferred information over the network (files, messages, e-mails, etc) and simply gather the information flowing in the network [21]. Moreover, the attacker could even use the information it gathers about an individual to impersonate her/his, which is known as *identity theft* [22]. For instance, in [23] the authors present how to clone an existing account in an online social network and to establish a friendship connection with the victim in order to obtain information about her/him.

In order to avoid undesired information collection, sensitive personal information must be protected from access by any other third party that is different from the agent to which the information is directed to. Therefore, avoiding information collection requires security to control the access to personal information [24]. In particular, confidentiality is a security property of a system that ensures the prevention of unauthorized reading of information [25]. In distributed environments, confidentiality usually means that sensitive information is encrypted into a piece of data so that only parties that can decrypt that piece of data can access the sensitive information.

Confidentiality can be achieved by using existing secure data transfer technologies such as Kerberos [26], SSL [27], and TLS [28]. These technologies allow the encryption of messages before transferring them and the decryption of messages once they are received. As a result, if an agent A sends a message to an agent B using these technologies, A is sure that B will be the only one able to read this message.

Confidentiality is a necessary condition to preserve privacy, but it is not sufficient. It prevents undesired information collection from unauthorized third parties. If an agent A sends personal information to an agent B in a confidential fashion, external third parties will not be able to access it. However, agent B will obviously receive this personal information. The point is that agent B can then process the received personal information, unless specific measures for preventing information processing are adopted before sending this information.

### *2.2. Information Processing*

Information processing refers to the use or transformation of data that has already been collected [29], even though this information has been collected by mutual consent between two parties. An example of information processing is profiling [30]: "*the process of 'discovering' patterns in data that can be used to identify or represent a human or nonhuman subject (individual or group) and/or the application of profiles (sets of correlated data) to individuate and represent an individual subject or to identify a subject as a member of a group (which can be an existing community or a discovered category) and/or the application of profiles to individuate and represent individuals or groups*".

One of the most common types of profiling is called buyer profiling in e-commerce environments, in which vendors obtain detailed profiles of their customers and tailor their offers regarding customers' tastes. These profiles can represent a serious threat to privacy. For instance, these profiles can be used to perform *price discrimination* [31]. Vendors could charge customers different prices for the same good according to the customers' profiles, i.e., if a vendor knows that some good is of great interest to one customer, the vendor could charge this customer more money for this good than other customers for the same good. For instance, in 2000, Amazon started to charge customers different prices for the same DVD titles [32]. When the story became public, Amazon claimed that this was part of a simple price test and discontinued this practice. Another example of privacy threat due to the use of these profiles is what is known as *poor judgment* [33]. This is when individuals are judged and subsequently treated according to decisions made automatically based on incorrect or partial personal data. For instance, companies usually divide their potential customers into similar groups based on customers' characteristics (known as customer segmentation). This practice can lead to exclusion of people from services based on potentially distorted judgments [29].

Most of the work on protecting against the processing of information already collected is based on minimizing data identifiability. Identifiability can be defined as "the degree to which (personal) data can be directly linked to an individual" [29]. The degree of privacy of a system is inversely related to the degree of user data identifiability. The more identifiable data that exists about a person, the less she/he is able to control access to information about herself/himself, and the greater the privacy risks. Identifiability ranges from complete identifiability to anonymity.

Pseudonymity [34] is the use of pseudonyms as

identifiers. A pseudonym is an identifier of a subject other than one of the subject's real names. Pseudonyms have been broadly used by human beings in the real world. For instance, in the 19th century when writing was a male-dominated profession, some female writers used male names for their writings. Nowadays, in the digital world, there is a great number of pseudonyms such as usernames, nicknames, e-mail addresses, sequence numbers, public keys, etc. [35].

The most important trait of pseudonymity is that it comprises all degrees of identifiability of a subject (from identified to anonymous) depending on the nature of the pseudonyms being used. Complete identification is when the linking between a pseudonym and its holder is publicly known. Anonymity can be achieved by using a different pseudonym for each different interaction. This is known as transaction pseudonyms [34]. For instance, let us suppose that two agents A and B act as a buyer and a seller, respectively, in a e-marketplace. Agent A can use a different pseudonym (e.g. a random generated numeric identifier) for each specific interaction with agent B. Hence, Agent B collects information about the interactions performed but is unable to relate different interactions to each other or relate any of these interactions to agent A[5].

As we detail later on in this paper (Section 4), Magentix2 allows agents to use as many pseudonyms as they need to prevent information processing. We refer to these pseudonyms as *regular* pseudonyms.

### *2.3. Implications in Accountability*

Minimizing data identifiability may have a direct impact on accountability. Accountability refers to the ability to hold entities responsible for their actions [38]. Accountability usually requires an unambiguous identification of the principal involved [39]. Then, this principal can be held liable for their acts. For instance, a buyer agent pays a seller agent for a good. The seller commits to shipping the good to the customer agent's principal. In the event that the customer agent's principal does not receive the good, the seller agent's principal[6] may be held liable for this. Although determining exactly who should be held liable for this depends on the applicable laws in the specific country, it usually requires the identification of the seller agent's principal. Then, the seller agent's principal can be sued for fraud.

Many systems (such as commercial systems) emphasize accountability because, in these environments, principals can be subject to serious losses such as money loss. Moreover, the sense of impunity generated by the lack of accountability could even encourage abuse. Thus, accountability is of crucial importance for agent-based applications because it helps to promote trust in these applications, which is needed for principals to be willing to engage with and delegate tasks to agents [42].

Pseudonyms can be utilized to implement accountability [43]. However, this implies that there is a trusted entity that issues pseudonyms and knows the association between pseudonyms and the real identity of the principal that is behind an agent. Therefore, when an agent misbehaves using a given pseudonym, this pseudonym can be traced back to the principal behind this agent for law enforcement. One approach to do this would be that the AP is the one that issues pseudonyms. Therefore, the AP itself can disclose the principal behind the pseudonym, removing pseudonymity and producing identity and accountability as a result. The main drawback of this approach is that the AP itself knows the relation of pseudonyms to each other and to the principal involved. As APs are usually to be run by the same company that hosts the specific system (e.g. eBay), information processing could still be easy. We think that decoupling the place in which the identities are issued and the place in which the identities are effectively used can make this harder. Therefore, we

[5] Even when buyer agents are able to change its pseudonym, purchases may include information that can be used to relate different purchases to each other and to the buyer agent, e.g., the credit card number to perform the payments and the shipment address may be the same for different transactions. We assume that payments are carried out using some kind of anonymous payment mechanism and deliveries are carried out using some anonymous delivery system. Hence, credit card numbers and delivery addresses do not need to be disclosed when an agent acquires a product. For instance, the untraceable electronic cash presented by Chaum et al. [36] can be used for anonymous payments. For anonymous deliveries, the privacy-preserving physical delivery system presented by Aïmeur et al. [37] can be used.

[6] Software entities (intelligent agents, virtual organizations, etc.) cannot have real identities because, until now, they could not be held liable for their acts in front of the law. However, this may change in the future if they finally achieve some kind of legal personhood, as suggested by [40] and [41]. In this case, software entities may be provided with legal personhood to be (partially) held liable for their acts. The point is that according to the law, someone must be liable for frauds like this.

consider that pseudonyms are issued and validated by external trusted third parties (as explained later on in Section 4) that do not participate in any way in the specific system.

### *2.4. Implications in Trust and Reputation*

There is also the need to equip agents with models to reason about and assess trust towards other agents and their reputation in a MAS [5]. These models allow agents to select the best and most reliable partners in a specific situation and to avoid partners of previous unsuccessful interactions. Trust and reputation are even more important in open systems, in which previously unknown parties may interact. For instance, if a buyer agent enters into an e-marketplace for the first time, it will need to choose among all of the available seller agents. As the buyer agent has no previous interactions with the seller agent, the reputation of the seller agent in the e-marketplace can play a crucial role for the buyer agent to choose an specific seller agent.

The agent community has developed a vast number of trust and reputation models [44, 45]. However, current trust and reputation models are based on the assumption that identities are long-lived, so that ratings about a particular agent from the past are related to the same agent in the future. However, when such models are actually used in real domains this assumption is no longer valid and the so-called identity-related vulnerabilities emerge. These vulnerabilities can be exploited by means of *white-washing* attacks [46] (also called *change of identities* [47]). For instance, an agent that has a low reputation due to its cheating behavior may be really interested in changing its identity and restarting its reputation from scratch. These vulnerabilities can also be exploited by means of *sybil* attacks [48]. An example of this attack could be an agent that holds multiple identities in a marketplace and attempts to sell the same product through each of them, increasing the probability of being chosen by a potential buyer.

Due to the identity-related vulnerabilities of trust and reputation models, malicious agents may be able to modify the expected behavior of trust and reputation models. As a result, these reputation models may become completely useless. For instance, in our previous example, a seller agent may be able to cheat the reputation model used by the buyer agent. Thus, the buyer agent may end up interacting with a malicious agent instead of what it believes a reputable agent. These has the potential to cause many damages such as money lose. Therefore, these vulnerabilities have the potential to place the whole system in jeopardy.

A possible solution for these vulnerabilities is the use of *once-in-a-lifetime* pseudonyms [49]. Agents can only hold one *once-in-a-lifetime* pseudonym in each marketplace. Therefore, they cannot get rid of the trust and reputation ratings they got from other agents in the marketplace. However, minimizing data identifiability requires that agents have the ability to hold and use many pseudonyms. Thus, it could seem that a system that allows the prevention of information processing may have important difficulties to also prevent identity-related vulnerabilities.

To address this, Magentix2 bases on the agent identity model proposed in [50], as we explain later on in Section 4. Agents in Magentix2 can hold two kinds of pseudonyms: permanent pseudonyms (that correspond to *once-in-a-lifetime* pseudonyms), which avoid identity-related vulnerabilities; and regular pseudonyms which agents can use without any limitation in number to obtain their desired degree of privacy.

## 3. The Magentix2 Agent Platform

Magentix2[7] is an AP that focuses on providing support for open MAS. Magentix2 provides support at three levels, as suggested in [51]: (i) organization level: technologies and techniques related to agent societies; (ii) interaction level: technologies and techniques related to communications between agents; and (iii) agent level: technologies and techniques related to individual agents (such as reasoning and learning).

In this paper we focus on the technologies that Magentix2 provides at interaction level to enhance the privacy of the applications that are developed and executed on top of it. Firstly, we briefly introduce the Magentix2 agent communication mechanism in order for the reader to have a bird's-eye view of the system. Then, we thoroughly describe the support that Magentix2 provides to avoid information processing and information collection in Sections 4 and 5 respectively.

[7] `http://users.dsic.upv.es/grupos/ia/sma/tools/magentix2/index.php`

### 3.1. Agent Communication

Magentix2 uses AMQP[8] [52] as a foundation for agent communication. This standard facilitates the interoperability between heterogeneous entities. Magentix2 allows heterogeneous agents to interact with each other via messages that are represented following the FIPA-ACL [53] standard, which are exchanged using the AMQP standard.

Magentix2 uses the Apache Qpid[9] open-source implementation of AMQP for Agent Communication. Apache Qpid provides two AMQP servers, implemented in C++ (the one we use) and Java. Qpid also provides AMQP Client APIs that support the following languages: C++, Java, C# .NET, Ruby, and Python. Qpid allows distributed applications made up of different parts written in any of these languages to communicate with each other. What is more, any client that is developed using one of the Qpid Client APIs is able to communicate with any client that is developed using any other AMQP-compliant API via any AMQP server implementation, as long as both server and clients implement the same version of the AMQP standard.

Figure 1 shows an overview of the Magentix2 agent communication architecture. Magentix2 is composed by one or more (in this case federated) AMQP Servers (QPid brokers). Magentix2 agents act as AMQP Clients (using Qpid Client APIs) that connect to the Qpid broker and are then able to communicate with each other. Magentix2 agents can be located in any Internet location, they only need to know the host on which the Qpid broker (or one of the federated Qpid brokers) is running.

Magentix2 provides a Java library, which is called the Magentix2 Agent Library (MAL), to facilitate the development of agents. This API allows agent programmers to focus on creating FIPA-ACL messages and sending and receiving them, without dealing directly with the Qpid Client Java API. Currently, this API is only written in Java, but the existence of multiple QPid Client APIs for several programming languages enables the development of agents written in different programming languages. Moreover, any proprietary implementation that follows both AMQP and FIPA-ACL standards would be interoperable with Magentix2 agents.

[8] http://www.amqp.org/
[9] http://qpid.apache.org/

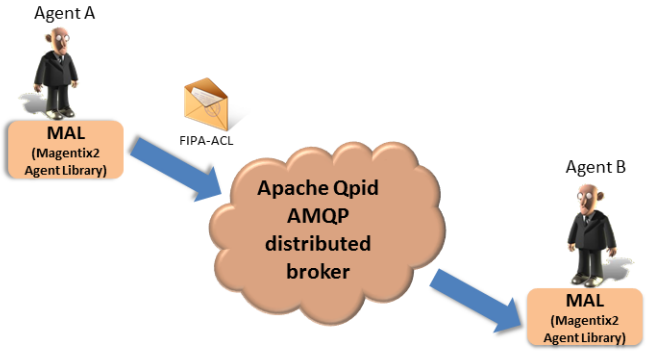


Figure 1: Magentix2 Agent Comunication Architecture

## 4. Protection against Information Processing

As we detailed in section 2, information processing can be minimized by minimizing data identifiability, i.e., using many pseudonyms (even complete anonymity can be achieve by using a different pseudonym for each interaction). However, this introduces problems regarding accountability, trust and reputation. We describe in this section the support that Magentix2 provides for avoiding information processing but preserving accountability and avoiding identity-related vulnerabilities of trust and reputation models.

The building block of the Magentix2 support for avoiding information processing is how it manages the identities of the agents. In particular,the agent identity management in Magentix2 is based on the partial identities model presented in [50]. In a nutshell and informally speaking, a partial identity can be seen as a set of attributes that identifies an entity in a given context. They are composed of a pseudonym that is unique within a context and other attributes that describe the entity within that context (roles, location, preferences, etc.).

Magentix2 considers two kinds of partial identities:

- **Regular Partial Identities (RPIs)** A RPI must contain a regular pseudonym. Agents have no limitation in the number of regular pseudonyms that they are able to hold in a given system. Thus, agents can hold as many as RPIs as they need to achieve they desired data identifiability level. For instance, an agent could use a different RPI in each interaction to achieve anonymity.
- **Permanent Partial Identities (PPIs)** must contain a permanent pseudonym (*once-in-a-lifetime*

pseudonym). An agent can only hold one permanent pseudonym for a given system. Thus, if trust and reputation is established trough PPIs, identity-related vulnerabilities of trust and reputation are avoided. This is because an agent can only hold one PPI in a given system.

Although both kinds of partial identities enable trust and reputation relationships, only PPIs guarantee that identity-related vulnerabilities are avoided. Therefore, agents will choose to establish trust and reputation through PPIs if they want to avoid identity-related vulnerabilities. However, if they want to avoid information processing, they can use as many RPIs as needed. For instance, in an e-marketplace a seller agent could be very interested in interacting with buyer agents using a PPI. This is because buyer agents would know that the seller agent is not able to perform attacks that exploit identity-vulnerabilities, e.g., the seller agent is not able to change its PPI. However, a buyer agent can be very interested in using a different RPI each time it interacts with a seller agent, so that the seller agent is not able to make a profile of the buyer agent's tastes and then use this profile to perform practices such as price discrimination, poor judgement, etc.

Magentix2 also considers the concept of real identities. Real identities identify entities that can be liable for their acts in front of the law, such as human beings, companies, etc. Real identities are used for accountability concerns such as law enforcement. For this reason, real identities are restricted to only legal persons. A real identity, for example, would be: *Bob Andrew Miller, born in Los Angeles, CA, USA on July 7, 1975.* Agents might also have a real identity for accountability concerns if they finally achieve some kind of legal personality, as suggested by [40] and [41]. In the event of this, the support that Magentix2 provides would not need any modification.

### *4.1. Identity Management Architecture*

Magentix2 complies with the client part of the Identity Metasystem Interoperability standard[10]. This standard specifies the interfaces for the secure web services provided by User-Centric Privacy-Enhancing Identity Management Systems [54]. These systems support the process of management of partial identities. In particular, they provide the following facilities:

- **Identity Providers (IdPs)**, which issue partial identities and validate these identities to other Relying Parties.
- **Relying Parties**, which are a set of APIs for verifying partial identities against an Identity Provider.
- **Identity Selectors**, which provide a simple way to manage partial identities and choose which partial identity to use in a given context.
- **Attribute Services**, which allow the specification of access control rights of relying parties over the attributes in a partial identity.

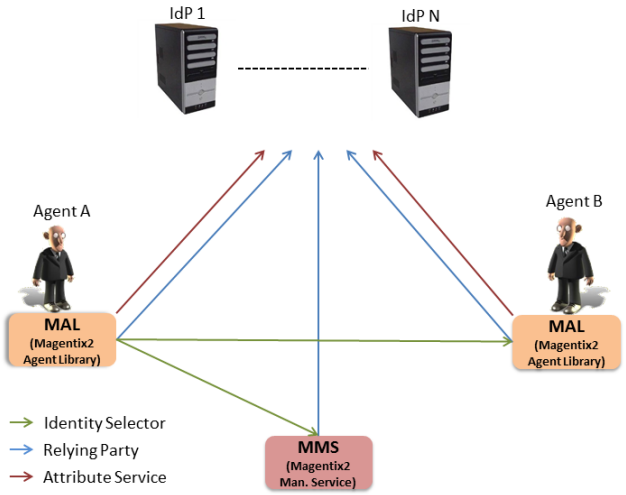


Figure 2: The Magentix2 agent identity management architecture.

Figure 2 shows an overview of the Magentix2 agent identity management support. As one can observe, Magentix2 components act as clients of IdPs. The Magentix2 Agent Library (MAL) implements clients for Identity Selectors, Relying Parties, and Attribute Services. Therefore, agents in Magentix2 can select the partial identity to use in a given interaction, verify the partial identities of other agents, and specify access control for attributes in their partial identities.

We can also see in Figure 2 that there is another component of Magentix2 that is called the Magentix2 Management Service (MMS). The MMS is a secure web service that acts as a Relying Party, i.e., it is able to request IdPs to verify partial identities. The MMS

[10] `http://docs.oasis-open.org/imi/identity/v1.0/identity.html`

is in charge of dynamically signing digital certificates for agents to communicate to each other in a confidential fashion, and thus, avoiding information collection from unauthorized parties (as detailed in section 5). Agents request the signing of digital certificates to the MMS using one of their partial identities. The MMS must verify the partial identity that the agent used before signing the digital certificate.

IdPs are classified according to the type of partial identities they issue. The Permanent Identity Provider (PIdP) is an IdP (or a federation of IdPs[11]) that issues PPIs to the agents taking part in the specific marketplace. Agents must register using a real identity that the PIdP will not reveal to other agents or to Magentix2. The PIdP is also in charge of forcing agents to only hold a single PPI in this specific marketplace. Moreover, an agent is able to know when it validates a partial identity of another agent that this partial identity is effectively a PPI so that the second agent cannot change it and is unable to perform identity-related attacks.

Regular Identity Providers (RIdPs) issue RPIs to agents. Agents request RPIs by providing either a real identity, or a PPI that RIdPs will not reveal to others. There is no limitation in the number of RIdPs per marketplace or in the number of RPIs per agent and per marketplace.

### *4.2. Obtaining Partial Identities*

We now detail the dynamics of agent identity management in Magentix2. Users provide their real identity when they launch an agent. This real identity is only used by the local instance of the MAL. Therefore, it is only in the local computer of the user that launches the agent. Moreover, the API offered by the MAL allows the user to choose an identity for the newly created agent. Then, the MAL calls to a PIdP/RIdP to obtain a PPI/RPI. From then on, and as we will explain in section 5, the agent can use this PPI/RPI to interact with other agents.

The important point is that neither Magentix2 nor the agents running in Magentix2 have access to the real identity of the user. They are only able to know the PPI/RPI that the agent is using. In case of law enforcement, a court could ask IdPs to disclose the real identity behind a particular PPI/RPI. Thus, accountability is preserved.

Whenever an agent wants to change its RPI (recall that PPIs cannot be changed), it only needs to call to the MAL API. Then, this API will be in charge of contacting the appropriate RIdP and obtaining a new RPI.

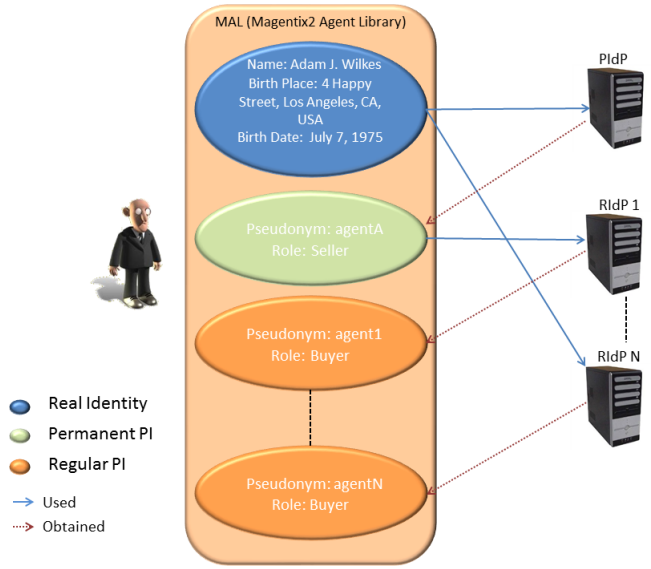


Figure 3: An example of the Partial Identities of an agent.

Figure 3 shows an example of an agent and its partial identities. The agent's principal has a real identity with an attribute name *Adam John Wilkes*. Using this real identity, the agent has obtained a PPI from the PIdP that includes two attributes: pseudonym `A` and role `seller`. This entity has also obtained N RPIs from N different RIdPs. Some of the RPIs are obtained by providing a PPI (such as RPI 1, with pseudonym `agent1` and role `buyer`) and other RPIs are obtained using a real identity (such as RPI N, with pseudonym `agentN` and role `buyer`).

## 5. Protection against Information Collection

In this section, we detail the support that Magentix2 provides for avoiding information collection. As we explained in Section 2, information collection by

[11] User-Centric Identity Management Systems support the federation of IdPs that belong to the same and also different remote security domains across the Internet. Therefore, a PIdP can be implemented as a federation of IdPs instead of only one IdP, minimizing the typical drawbacks of a centralized trusted third party, such as being a single point of failure (SPOF) and a possible efficiency bottleneck. Examples of identity federation standards are the Liberty Alliance Identity Federation Framework `http://projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications/` and WS-Federation `http://www.ibm.com/developerworks/library/specification/ws-fed/`.

unauthorized parties can be avoided by means of confidentiality. Therefore, we explain in this section how Magentix2 makes agent communications confidential. Moreover, we detail how this confidential agent communication integrates with the agent identity management model presented in the previous section.

Agent communication in Magentix2 is based on AMQP. The AMQP standard specifies secure communication by tunneling AMQP connections through SSL [27] (so-called AMQPS). Apache Qpid implements SSL support for AMQP. SSL authenticates communicating parties based on digital certificates. Thus, it needs a configured Public Key Infrastructure (PKI). The Magentix2 PKI is set during installation time. Firstly, the Magentix2 certificate authority (MCA) is created. Secondly, certificates for the Magentix2 Management Service (MMS) and the Qpid Broker are created using this certificate authority. Digital certificates for agents are created automatically by the MAL and dynamically signed by the MCA through the MMS at execution time (as described below).

The MMS is a front-end of the MCA. It is implemented as a secure web service. The MMS is in charge of dynamically signing digital certificates for agents, which can use these certificates to communicate securely. The MMS service needs two inputs: the agent pseudonym and a non-signed digital certificate. The first input is the pseudonym in the permanent or regular partial identity (issued by a permanent or regular IdP) that the agent uses to invoke the MMS. The second input is a non-signed certificate that contains the agent's public key (this is the certificate that is to be signed). The agent key pair (private and public key) and this certificate are created by the MAL *locally* for each agent and for each new partial identity.

The MMS produces one output: the digital certificate *signed* by the MCA. The MMS produces this output after: (i) verifying that the pseudonym is the same as the one in the partial identity used to invoke the secure web service; (ii) verifying the partial identity against the IdP that issued it; (iii) and finally signing the certificate using the MCA. Agents can then use this signed certificate to communicate to other Magentix2 agents.

The AMQP connection of every agent to the Qpid broker is tunneled through SSL. Hence, the communication between two Magentix2 agents is provided with confidentiality and integrity out of the box. To ensure the authenticity of the sender pseudonym in a FIPA-ACL message (recall that in Magentix2 FIPA-ACL messages are encapsulated into AMQP messages), an agent must verify that the pseudonym of the sender in the AMQP sender message field is the same as the pseudonym of the sender in the FIPA-ACL sender message field upon receiving a new message. This is performed automatically by the Magentix2 agent library.

Each time an agent obtains a new partial identity (RPI or PPI) as described in Section 4, the MAL repeats the process for obtaining a MCA-signed certificate. This certificate is for the agent to communicate in a confidential fashion when it uses this partial identity. Moreover, agents can also choose the partial identity to be used in each communication. Therefore, the MAL is in charge of using each time the certificate that corresponds to the partial identity that the agent intends to use.

Figure 4 shows an example of an agent that holds a RPI with two attributes: the pseudonym A, and the role buyer. The MAL automatically creates a certificate that contains the pseudonym A and calls to the MMS using the agent's RPI. The MMS validates the RPI to the IdP that issued it. Then, it signs the certificate using the MCA and sends back the signed certificate to the A agent. From this moment on, when the A agent sends a message, the MAL automatically send this message trough an AMQPS connection established with the signed certificate received from the MMS. Thus, agent A can communicate with agent B in a confidential fashion.

## 6. Application to an Agent-based E-marketplace for Wines

In this section, we detail an application built on top of Magentix2. It is an agent-based e-commerce application. Agent-mediated e-commerce [42, 6] refers to electronic commerce in which agent technologies are applied to provide personalized, continuously running, semi-autonomous behavior. In agent-mediated e-commerce, agents can be distributed among different remote locations over the Internet. For instance, buyer agents mostly act as personal assistants that usually run on a local mobile/personal device/computer and interact with other seller agents that usually run in remote servers.

Agents partially or fully automate some of the processes involved in e-commerce, providing one or more of the following (and also other) facilities [42]:

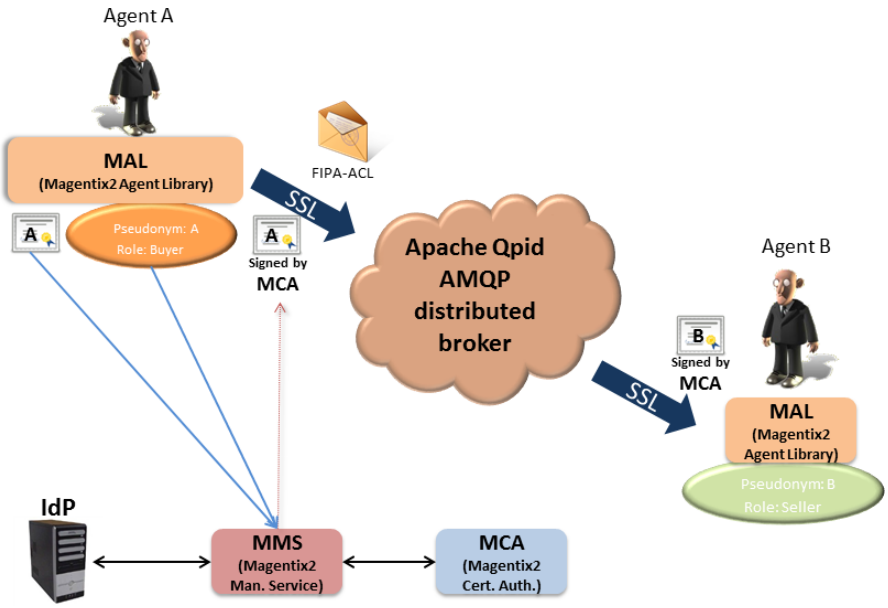


Figure 4: Secure Agent Communication (information collection avoidance).

- find, recommend and compare products, vendors, or services;
- participate in electronic markets and negotiate the price/terms of transactions or contracts with other participants;
- perform transactions on behalf of their users;
- track the user's interests and offer personalized services;
- monitor conditions and provide notifications;
- retrieve, filter, and mine information and knowledge;
- produce and deliver e-services, such as information gathering, processing and management.

In agent-based e-commerce, agents encapsulate personal information describing their principals [42]. They usually have a detailed profile of their principals' names, preferences, roles in organizations and institutions, location, transactions performed, and other personal information. Moreover, agents carry out interactions on behalf of their principals, so they exchange this personal information. Therefore, privacy is of crucial importance in this kind of applications.

In this paper, we consider an electronic market where seller agents and buyer agents trade wines on behalf of their users. Seller agents act on behalf of wine merchants. Buyer agents act on behalf of the users that are willing to buy wines. As shown in Figure 5, each buyer agent can interact with any of the seller agents that is available to buy wines. Moreover, each agent is supposed to be running in a different Internet location.

Agents in the e-marketplace follow the negotiation protocol depicted in Figure 6 for each purchase of a bottle of wine. A buyer agent makes a request to buy a bottle of wine with a *request* message. This message can be replied by the seller agent with either a *model* message (which means that the requested wine is available and includes its price) or an *alternative* message (which means that the requested wine is not available but there is another one that is very similar). Then, the buyer agent can reply to both messages with: an *accept* message (which means that the buyer agent accepts the wine offered), a *quit* message (which means that the negotiation was broken by the buyer agent), or a *request* message (which means that the agent request a different bottle of wine).

We based on the wine attributes considered in the preference modeling approach described in [55]. Thus, we consider the following attributes to describe wines: color, body, flavor, sugar, and country. The possible values for each of these attributes are shown in Table 1.

### *6.1. Privacy*

Privacy can be a great concern in this application. This is because agents are subject to informa-

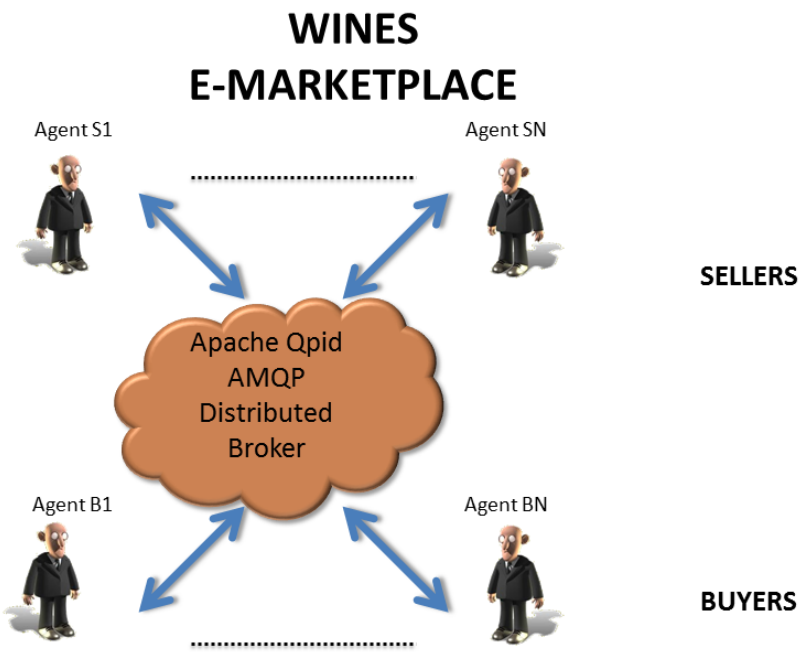


Figure 5: Wines e-Marketplace

| Attribute | Values |
|---|---|
| Color | red, rose, white |
| Body | light, medium, full |
| Flavor | delicate, moderate, strong |
| Sugar | dry, offDry, sweet |
| Country | France, Portugal, Spain, Italy, USA, Germany, Australia, NewZeeland |

Table 1: Considered Wine Attributes.


Figure 6: Negotiation Protocol for the Wine e-marketplace scenario.

tion collection and processing. Specifically, agents can exchange personal information when purchasing bottles of wine. This information must be protected from unauthorized parties, i.e., the information that a seller agent and a buyer agent exchange in a purchase should not be accessible for other agents in the e-marketplace.

Regarding information processing, seller agents could perform buyer profiling, i.e., seller agents obtain detailed profiles of buyer agents and tailor their offers regarding buyer agents's tastes. Seller agents could even charge buyer agents different prices for the same wine according to the customers' profiles (price discrimination), i.e., if a seller agent knows that some wine is of great interest to one buyer agent, the seller agent could charge this customer more money for a bottle of wine than other buyer agents for the very same bottle of wine.

We assume that seller agents follow an approach to build buyer agents' preference profiles similar to [56]. Over the course of a negotiation, the seller agent marks wines that are not accepted by a buyer agent as a negative instance (class "-"), while a seller agent marks the wine that the buyer agent accepts to buy (i.e., the wine of the last step in the nego-

tiation protocol) as a positive instance (class "+"). The seller agent use all the collected instances about a buyer agent to train a statistical classifier. The resulting trained classifier models the buyer agent's preferences with a given accuracy. Thus, the seller agent has a detailed profile on what wines the buyer agent likes/dislikes. Moreover, the seller agent can use this profile in future purchases from the buyer agent, e.g., the seller agent could then perform price discrimination to a buyer agent with respect to an item that the seller agent knows that is of great interest for the buyer agent.

### *6.2. Trust and Reputation*

Trust and reputation also play a crucial role in this application. Buyer agents must be able to choose among seller agents that sell the same wines. One of the important dimensions that a buyer will take into account in its decision is the trust it has in each seller agent. This trust can be based on successful previous purchases with the same seller agent. A buyer agent can trust in a seller agent regarding past interactions by measuring: whether or not the seller agent provisioned the wine on time, the overall quality of the wine bought, if there were hidden costs, etc. A buyer agent can also trust in a seller agent regarding some attributes of the seller agent's partial identity in the electronic market: registration date, corporate title, skills, etc.

Another important dimension that a buyer agent will take into account in its decision of buying a bottle of wine is the reputation of the seller agent. This is particularly useful when the buyer agent had no previous transactions with a given seller agent. In this case, it is not what an agent thinks of a given seller agent but what it is generally said about the seller agent in the e-marketplace.

In this scenario, malicious seller agents could take advantage of the identity-related vulnerabilities of trust and reputation models. Thus, seller agents should not be able to get rid of their trust and reputation assessments. This could cause important money loss. For instance, a seller agent could ship bottles of wine not on time. This obviously decreases the trust and reputation that buyer agents have in this seller agent. Hence, this seller agent decides to quit the electronic market and re-entry into it with a new fresh identity, restarting its trust and reputation assessments from scratch. Another example would be a seller agent which sell the same wines under different partial identities. In this sense, the probability that a buyer agent chooses one of their partial identities as the provider of the wine increases.

### *6.3. Accountability*

As in many other applications, accountability must be preserved. Moreover, accountability is even more important in applications that involve commercial transactions, such as the one that we present in this paper. Indeed, the lack of accountability could cause principals not to be willing to participate in the e-marketplace. This is because if there is not a guarantee that accountability will be preserved, principals may be subject to many types of fraud. For instance, a buyer agent could purchase a bottle of wine from a seller agent. However, the seller agent may not ship that bottle of wine. If we consider a great number of bottles of wine (or a very expensive bottle of wine), the seller agent's principal should be sued for fraud. Although the final punishment may depend on the applicable laws for such a case, the important point is that the principal behind the seller agent should be completely identifiable.

## 7. Evaluation

In this section, we provide an extensive evaluation of the support that Magentix2 provides for preserving privacy based on the application we just presented in the previous section. In particular, we test whether or not information processing can be minimized by using the support that Magentix2 provides, whether or not this support introduces a bearable performance overhead, and whether or not existing trust and reputation models can be implemented on top of Magentix2. Finally, we also provide a discussion at the end of this section in which we argument how Magentix2 complies with all of the stated requirements of the application presented in Section 6 regarding privacy, trust and reputation, and accountability.

### *7.1. Avoiding Information Processing*

Our aim in this section is to experimentally demonstrate that information processing, and thus, its possible undesired effects, can be minimized by changing RPIs. To this aim, we designed an experiment in which 10 different buyer agents purchase bottles of wine from a seller agent. The primary objective is for buyer agents to avoid that the seller agent is capable of obtaining a preference model from them.

As we explained in the previous section (Section 6), the seller agent trains an statistical classifier that models buyer agents' preferences (following the approach described in [56]). The statistical classifier is trained considering instances that correspond to wines that have been offered by the seller agent and that buyer agents have either accepted or declined during the negotiation protocol for a given purchase. The resulting trained classifier models the buyer agents' preferences with a given accuracy.

The aim of the experiment is to demonstrate that changing RPIs can significantly reduce the accuracy of the preference models obtained by the seller agent. Therefore, the seller agent cannot take advantage of these models to abuse buyer agents, e.g., performing price discrimination on buyer agents. In order to prove that changing RPIs can reduce information processing, we performed our experiment in which 10 buyer agent repeat 100 purchases of a bottle of wine with a varying number of RPI changes. That is, buyer agents start with 100 purchases and without any RPI change and end up using a different RPI for each of the 100 purchases. For each number of RPI changes we calculate the accuracy of the resulting classifier.

For our experiment, we considered the parameters that we sum up in Table 2. Each buyer agent has different preferences with respect to the wines that it likes. The specific preferences for each agent are shown in Table 3. According to that preferences, each buyer agent performs 100 different purchases of a bottle of wine. Each purchase involves a negotiation with a seller agent to get the desired wine. We assume that negotiations are always successful. However, we consider that negotiations can randomly involve from 1 up to 10 rounds of the protocol. That is, we simulate negotiations on which a buyer agent and a seller agent perform a maximum number of 10 rounds of the protocol. Based on this, a seller agent marks wines that are not accepted by a buyer agent as a negative instance (class "-"), while a seller agent marks the wine of the last step in the protocol (i.e., the wine that the buyer agent accepts to buy) as a positive instance (class "+"). After the 100 purchases, the seller agent use all the collected instances about a buyer agent to train a classifier. We measure the accuracy of the resulting trained classifier as the percent of correctly classified instances from an extra set of test instances (positive and negative) that we generate according to the buyer preferences.

We implemented the seller agent so that it uses a different classifier to obtain a model of the buyer agents' preferences based on the previous negotiations with them. In this way, we repeat the overall experiment to obtain the results regarding three different classifiers. Specifically, we consider the same classifiers as in [56]: the J48 decision tree algorithm (an implementation of the C.45 algorithm), the NNge classification rules algorithm (Nearest neighbor like algorithm using non-nested generalized exemplars) and the BayesNet classifier that is a classifier based on Bayesian networks.

| Parameter | Description | Value |
|---|---|---|
| Ni | # of interactions per negotiation | 10 |
| Nn | # of negotiations | 100 |
| Nre | # repetitions of the experiment | 100 |
| Nbu | # of buyer agents | 10 |
| Nse | # of seller agents | 1 |
| Alg | Learning algorithms used | J48, NNge, BayesNet |

Table 2: Parameters used in the privacy preservation experiments.

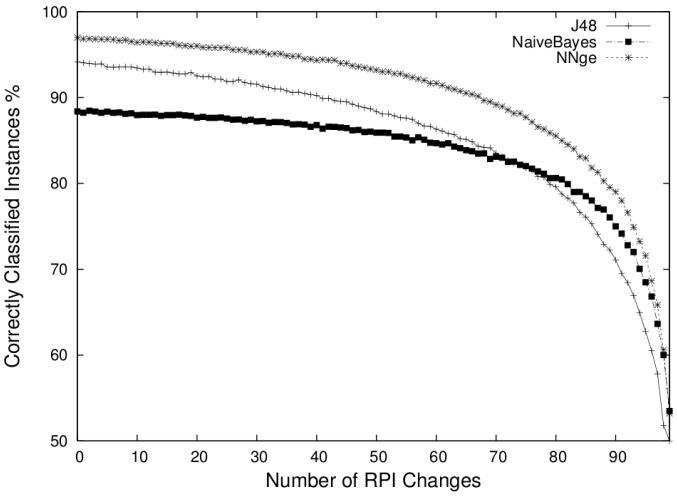


Figure 7: Privacy Preservation when changing RPIs

Figure 7 shows the results obtained for our experiment. These results are the average of the results obtained for each individual buyer agent. We can see that the percent of correctly classified instances behaves very similar regardless of the learning algorithm used. As expected, the more a buyer agent changes its RPI, the less the accuracy of the learning algorithms. In other words, the more a buyer agent changes its RPI, the less the seller agent is able to obtain a preference model of the buyer agent. Therefore, buyer agents can avoid that the seller agent performs information processing. Thus, it cannot make

| Agent | Preferences |
|---|---|
| 1 | $(body = light \wedge flavor = delicate) \vee$<br>$(sugar = dry \wedge country = Portugal)$ |
| 2 | $(color = red \wedge body = full \wedge flavor = strong) \vee$<br>$(color = white \wedge body = light \wedge flavor = moderate) \vee$<br>$(sugar = offDry \wedge country = Germany)$ |
| 3 | $(sugar = sweet \wedge country = France) \vee$<br>$(sugar = dry \wedge country = Spain) \vee$<br>$(color = rose \wedge flavor = moderate)$ |
| 4 | $(color = red \wedge body = medium \wedge flavor = moderate) \vee$<br>$(color = rose \wedge body = light \wedge sugar = dry) \vee$<br>$(color = white \wedge body = full \wedge sugar = dry)$ |
| 5 | $(color = red \wedge body = medium \wedge flavor = moderate) \vee$<br>$(color = rose \wedge body = full \wedge country = Italy)$ |
| 6 | $(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$<br>$(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA)$ |
| 7 | $(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$<br>$(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA) \vee$<br>$(sugar = sweet \wedge color = rose \wedge body = medium \wedge country = Portugal)$ |
| 8 | $(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$<br>$(sugar = offDry \wedge color = white \wedge body = medium \wedge flavor = delicate \wedge country = Australia)$ |
| 9 | $(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$<br>$(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA) \vee$<br>$(sugar = sweet \wedge color = rose \wedge body = medium \wedge country = Australia)$ |
| 10 | $(sugar = dry \wedge color = red \wedge body = medium \wedge flavor = moderate \wedge country = France) \vee$<br>$(sugar = sweet \wedge color = white \wedge body = light \wedge flavor = delicate \wedge country = USA) \vee$<br>$(sugar = sweet \wedge color = rose \wedge body = medium \wedge country = Portugal) \vee$<br>$(sugar = offDry \wedge color = red \wedge body = full \wedge flavor = strong \wedge country = NewZeeland)$ |

Table 3: Buyer agent's preferences

any secondary use of the processed data.

Another remarkable phenomenon is that it seems that there is a threshold in the number of RPI changes ($\approx 90$ RPI changes) from which the accuracy of learning algorithms decreases at a faster rate. This reinforces the thesis of some privacy-enhancing technologies researchers that encourage users to change their identities as often as possible. Moreover, it is clear that the maximum privacy preservation is achieved when buyer agents change their RPI for each new interaction, which is known as transaction pseudonyms in the privacy-enhancing technologies literature.

### *7.2. Performance Evaluation*

In the previous section, we described an experiment that demonstrates that changing RPIs can minimize information processing. However, changing RPIs could also have costs associated to the change, e.g., a temporal cost. If this temporal cost is too high, it could discourage agents from using the RPI change capability offered by Magentix2. In this section, we describe the experiment that we carried out in order to evaluate the temporal cost of changing RPIs in the Magentix2 agent platform.

We performed a similar experiment as the one presented in the previous section, in which agents change their RPI a number of times in order to reduce information processing. In this case, we only focus on two agents, one buyer agent and one seller agent. Moreover, we do not calculate the accuracy of the preference model that the seller obtains but we calculate the temporal cost for the buyer to change its RPI a specific number of times. This is in order to ascertain whether or not it is temporally feasible for a buyer agent to change their RPI as many times as needed to prevent the seller agent from constructing a detailed model on the buyer agent's preferences.

We performed a simulation in which the buyer agent carries out 100 different purchases of a bottle of wine. Each purchase involves a negotiation with the seller agent to get the desired wine. We assume that negotiations are always successful. Moreover, we consider that negotiations can randomly involve from 1 up to 10 rounds of the protocol. The buyer agent repeats the 100 purchases with a varying number of RPI changes. That is, the buyer agent starts with 100 purchases and without any RPI change, and it ends up using a different RPI for each of the 100 purchases (transaction pseudonyms). For each number of RPI changes we calculate the RTT time of the messages exchanged between the buyer agent and the seller agent.

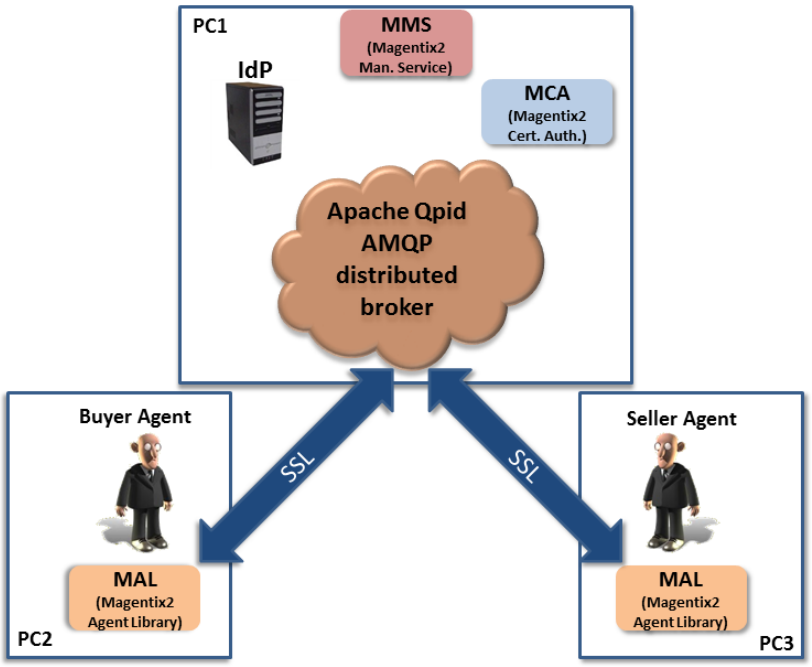


Figure 8: Location of the different components.

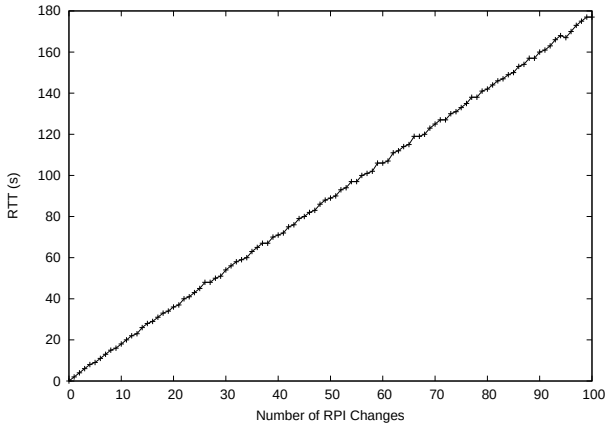


Figure 9: Performance per number of RPI changes

In order for the experiment to be in an absolutely controlled environment, we do not use any external IdP but we use the IdP prototype described in [50] as both the PIdP (the IdP that issues PPIs) and the RIdP (the IdP that issues RPIs). Moreover, we used 3 PCs Intel(R) Core(TM) 2 Duo CPU @ 2.60GHz, 1GB RAM, Ubuntu 11.04 (x86_64) and Linux Kernel 2.6.38. The computers are connected to each other via a 100Mb Ethernet switch. The security parameters are the following: certificate keys are 1024 bits RSA keys, SHA-1 hash function with 96-bit keys to perform HMAC computations, and the saml2 tokens to be issued by the IdP contain keys of 256 bits. The location of the different components is shown in Figure 8: PC number 1 runs the Qpid Broker, the MMS, the MCA, and the IdP; PC number 2 runs the buyer agent; and finally, PC number 3 runs the seller agent.

Figure 9 shows the results obtained. These results mean that changing a RPI has a temporal cost that is linear with the number of changes to be made. These results also mean that the temporal cost of a single change is constant, and thus, it is not related to the number of previously performed changes. Therefore, we can claim that agents developed in Magentix2 can minimize information processing about their princi-

pals' data (as shown in the previous section) without incurring in a not affordable temporal cost. Moreover, as the cost of one single change is constant, a buyer agent can predict in advance the temporal cost of the changes it requires to reduce seller agents from processing its data, and then, decide if it performs the required changes or not.

### *7.3. Building Trust and Reputation*

Finally, we also wanted to test whether trust and reputation models can be implemented using partial identities. The aim is to validate that agents in Magentix2 can use the trust and reputation model that they prefer. Thus, if two agents establish trust or reputation trough permanent partial identities, identity-related vulnerabilities are avoided because IdPs do not allow agents to change their permanent partial identity.

We implemented one seller agent and three buyer agents. Each buyer agent uses its own trust and reputation machinery to model the trustworthiness of the seller agent based on the previous interactions and the personal attributes of the seller agent. The PPIs issued by the PIdP take values for two attributes: name and role. Both seller agents and buyer agents register into the system using the PPI that the PIdP issued for them – so that the system does not know the real identity of the legal person that agents are acting on behalf of. In this way, buyer agents are able to identify providers from previous interactions and build their own trust and reputation models, being sure that the seller agent will not be able to hold any other PPI.

The seller agent follows a normal distribution with a mean of 0 and standard deviation of 1 to model whether it carries out the service requested in the way that buyer agents expect it. In this sense, when a buyer agent requests a service to the seller, if the value returned is in the interval [-1,1], the buyer agent considers that the seller performed as expected. If the value returned is out of this interval the buyer agent considers that the seller agent did not perform as expected. When the seller agent performs as expected, the buyer agent rates it with 1. When the seller agent does not perform as expected, the buyer agent rates it with 0. These ratings are inputs of the trust and reputation model each buyer agent has.

Each buyer agent runs a different trust and reputation model that is fed using past interactions with the seller agent and attributes from the seller agent's partial identity. We implemented three models (each one for each buyer agent), one simply using a mean of all the previous performances to compute a trust value, one using the SinAlpha trust model [57] that considers previous interactions, and finally, one using the Fire trust and reputation model [58] which uses, among other information, previous interactions and the role of the agents to be trusted. Figure 10 presents trust values for each buyer after 100 interactions with the seller agent. These trust values are the result of each buyer agent's trust and reputation model given the results of the interactions with the seller agent. We can observe that the trust models perform as expected and the tendency is very similar in all of the models implemented . This tendency corresponds to whether or not the seller agent performs as buyers expect. Thus, we can conclude that the trust models considered can be implemented using partial identities.

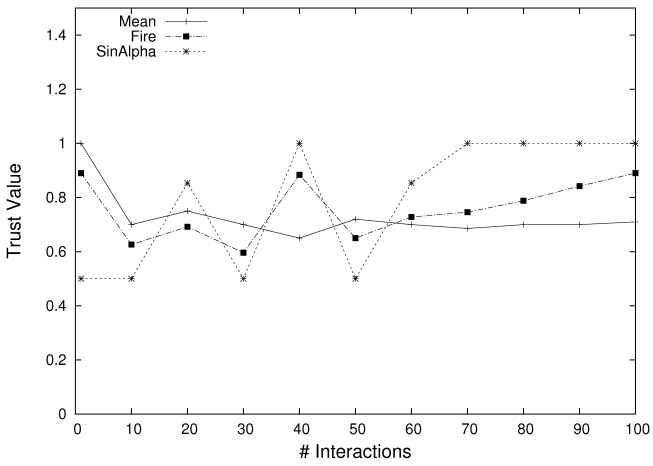


Figure 10: Trust values for the seller agent obtained by each trust model (implemented by each buyer agent).

### *7.4. Discussion*

The agent-based e-marketplace for wine application benefits from the features that Magentix2 provides for preserving privacy and accountability as well as avoiding identity-related vulnerabilities of trust and reputation. Magentix2 provides support for preserving privacy by means of mechanisms that can be used by agents to avoid information collection and processing.

Regarding information collection, the information that a buyer agent and a seller agent exchange is confidential. To this aim, Magentix2 is in charge of automatically generating a X.509 certificate for each agent

partial identity so that they can communicate trough SSL. Therefore, communications are encrypted from one agent to another so that the information contained in such communications cannot be accessed by any other unauthorized agent.

Regarding information processing, buyer agents can hold multiple RPIs. Therefore, they can avoid information processing as shown in Section 7.1. Indeed, the most privacy preserving option is for buyer agents to change their RPIs for each different purchase of a bottle of wine. Moreover, as shown in Section 7.2, RPI changes come at a bearable efficiency cost. Thus, agents can change their RPIs without an important efficiency penalty.

In Section 7.3, we demonstrate that trust and reputation models can be implemented using partial identities. Buyers and sellers are able to authenticate their partial identities (both PPIs and RPIs). Therefore, they are allowed to recognize to each other from interaction to interaction and establish trust and reputation relationships. Moreover, if trust and reputation is established using PPIs, identity-related vulnerabilities of trust and reputation models are avoided. This is due to the fact that there is no chance for a buyer or a seller to have two different PPIs.

Finally, accountability is preserved. IdPs conceal real identities. Therefore, both Magentix2 and the rest of the agents are a priori not able to link a given partial identity (PPI or RPI) to the corresponding original real identity. However, if an agent misbehaves when using one of its partial identities (PPI or RPI), a court could require the IdP that issued the partial identity to disclose the real identity that is behind the partial identity. For instance, if an agent performs fraud, the real identity of its principal could be known so that this principal can be sued for fraud.

## 8. Related Work

### *8.1. Privacy-enhancing Agent Platforms*

There are many Agent Platforms (APs) developed by the agent community [2]. However, only a few of them currently take security concerns into account. For instance, Jade [59], Magentix[12] [60], AgentScape [61], SECMAP [62], Tryllian ADK [63], Cougaar [64], SeMoA [65], and Voyager [66] are security-concerned APs.

Current security-concerned APs provide confidentiality for the messages exchanged by the agents running on top of them. To this aim, APs use existing secure data transfer technologies such as Kerberos [26], SSL [27], and TLS [28]. These technologies allow the encryption of messages before transferring them and the decryption of messages once they are received. As a result, if an agent A sends a message to an agent B using these technologies, A is sure that B will be the only one able to read this message.

Confidentiality is a necessary condition to preserve privacy, it can prevent information collection but it is not sufficient to prevent information processing. Only a few of the security-concerned APs explained above implement some kind of support to prevent information processing. Specifically, only Magentix, Secmap, AgentScape, and Cougaar allow agents to authenticate each other using their unique agent identity. With this identity, agents can act pseudonymously, i.e., agents can act on behalf of their principal without using the identity of their principal. However, agents cannot hold more than one pseudonym, i.e., principals should use a different agent each time they want to use a different pseudonym.

Warnier and Brazier [67] also present a mechanism for the AgentScape AP that offers pseudonymity by means of what they call *handles*. Handles are pseudonyms that agents can use to send/receive messages to/from other agents. At will, agents can request new handles to the AP. Moreover, the AP is the only one that knows the association between handles and GUIDs (global unique identities of the agents). An agent can also obtain anonymity by simply using a different handle for each transaction (transaction pseudonyms). AgentScape also offers an automatic anonymity service. Agents can send messages anonymously without having to manage pseudonyms. This service is provided by agents called *anonymizers*. When an agent wants to send a message anonymously, this message is redirected to an anonymizer. Then, this anonymizer is in charge of removing the original handle of the sender from the message, replacing it with another (possibly new) handle, and sending the message to the intended recipient. If the intended recipient replies, this reply is forwarded to the sender of the original message. The original sender of the message must notify when a transaction ends. For each new transaction the anonymizer

[12]Note that the support we present in this paper is for Magentix2, which is a completely redesigned version of Magentix [60].

generates a new handle.

APs that provide support for pseudonymity (e.g. by providing APIs to create and manage pseudonyms) do not consider that pseudonyms can be issued by external third parties. That is, APs themselves are in charge of issuing the pseudonyms. Thus, the AP itself (and the anonymizer agents for the case of AgentScape) knows the relationship of pseudonyms to each other and to the principal involved. This usually implies that the organization or company that hosts the specific system (e.g. eBay in the case of an e-marketplace) knows the association of pseudonyms to each other and to principals. Therefore, information processing could still be easy. In Magentix2 partial identities are issued by IdPs, and then, these partial identities are used in Magentix2. Therefore, identity management is decoupled from the system where the identities are to be used. Note that this may not completely prevent information processing. There is still the possibility that an agent running on Magentix2 or Magentix2 itself could collude with some of the IdPs in order to be able to link a partial identity to its corresponding real identity. However, agents could (partially) address this by obtaining different partial identities from different IdPs so as to decrease the probability of being traced back in case of collusion.

Finally, there are also approaches that have been implemented on top of APs that by themselves do not provide support to avoid information collection and processing. In this way, Lee et al. [68] present an approach based on P3P[13] for the Jade AP. The privacy-enhancing agent (PEA) is in charge of automatically retrieving P3P policies of service providers and evaluates whether or not these policies are compliant with its principal's policy. When a principal attempts to access a website, PEA automatically retrieves the website P3P policy and compares it to its principal's preferences. If PEA detects potential privacy violations (i.e., the principal's preferences and the website's P3P policy do not match) or is unable to read the policy of the website, it notifies its principal so that the principal can decide to desist in accessing the website. This approach does not consider that a website may not comply with its announced policy, and, thus, principals' privacy breaches are still possible. Moreover, this approach is not a generic support from the AP and may not be suitable in other domains.

[13]The Platform for Privacy Preferences 1.0 `http://www.w3.org/TR/P3P/`

### *8.2. Accountability Preservation*

Secure APs that do not provide support for pseudonymity (such as Jade [59], Tryllian ADK [63], SeMoA [65], and Voyager [66]) usually preserve accountability. This is because they base authentication on the the identity of the agents' users. Therefore, the identity of the agents' users is always available. If an agent misbehaves, its users identity can be known.

Accountability is more difficult in secure APs that provide support for pseudonymity (such as Magentix [60], Secmap [62], AgentScape [61], and Cougaar [64]). This is because these APs authenticate based on the identity (pseudonym) of the agents. Thus, the identity of the corresponding user may not be known. In order to avoid a lack of accountability that could cause a sense of impunity and encourage abuse, Magentix and AgentScape keep track of the association between principals and pseudonyms.

Magentix2 does not keep track of the association between principals and pseudonyms. It relies on trusted external identity providers to keep this information. However, accountability is preserved because these trusted external identity providers can reveal the corresponding real identities for law enforcement.

### *8.3. Identity-related vulnerabilities of Trust and Reputation Models*

There have been some different approaches in the related literature to tackle identity-related vulnerabilities of trust and reputation [69]. There is one approach that bases on adding a monetary cost for entering a given system [49]. Thus, an potential malicious agent would have a sufficient incentive (if the fee is high enough compared to the benefit expected) not to re-entry the system with a new identity. The main problem of this approach is that if the cost for entering the particular system is too high, even potentially benevolent agents may choose not to enter the system because of the high cost associated to it.

Yu et al. [70] present an approach based on social networks represented as a graph in which nodes represent pseudonyms and edges represent human-established trust relationships among them in the real world. They claim that malicious users can create many pseudonyms but few trust relationships. They exploit this property to bound the number of pseudonyms to be considered for trust and reputation. However, this approach is not appropriate for

open MAS in which agents act on behalf of principals that may not be known in the real world.

There is another approach that consist of reputation models specifically designed to meet some mathematical properties that are proved to avoid identity-related vulnerabilities. For instance, Cheng et al. [71] have demonstrated several conditions using graph theory that must be satisfied when calculating reputation in order for reputation models to be resilient to sybil attacks. The only drawback of this kind of approaches is that they usually need a particular and specific way to calculate reputation ratings about an individual. Thus, this approach cannot be applied to reputation models that follow other approaches for managing reputation ratings.

Magentix2 follows a different approach, it bases on completely avoiding the possibility for agents to change their identity. In this way, agents can only hold one PPI for a given system. Therefore, when trust and reputation are established trough PPIs, agents cannot get rid of the trust and reputation ratings they got from other agents in the system.

## 9. Conclusions

In this paper, we present the support that the Magentix2 AP provides for enhancing privacy in the applications built on top of it. This support also allows agents to avoid identity-related vulnerabilities of trust and reputation models. Moreover, this support avoids the lack of accountability of the principals involved. All these features are crucial for encouraging principals' trust in agent-based applications.

We experimentally show that the support that Magentix2 provides can be used for agents to minimize information processing. Moreover, we demonstrate that the efficiency cost of changing a RPI is absolutely bearable. Finally, we also demonstrate that current trust and reputation models can be implemented in Magentix2 by implementing some of them.

Agents running on Magentix2 can use the support it provides to enhance privacy while preserving accountability and avoiding identity-related vulnerabilities of trust and reputation at will depending on their principals' needs. An agent can create as many RPIs as needed to avoid information processing. Otherwise, an agent can use a PPI if it is interested in building trust and reputation. Thus, other agents can trust in this agent while being sure that it cannot perform whitewashing and sibyl attacks because PPI cannot be changed.

As future work, we would like to explore the possibility of agents automatically deciding whether or not they change their RPI for their next interaction. This decision may be based on the privacy that is to be lost if the RPI is not changed (e.g., an estimation of the model that other agents could have about their preferences) and the possible benefits of not changing the RPI. In particular, the decision of whether or not changing a RPI could be based on disclosure decision making models that consider a tradeoff between the privacy that is lost and the benefit of doing so, such as privacy-utility tradeoff models [72] and privacy-intimacy tradeoff models [73].

## 10. Acknowledgments

This work has been partially supported by CONSOLIDER INGENIO 2010 under grant CSD2007-00022, and project TIN2009-13839-C03-01.

## References

[1] M. Wooldridge, An Introduction to MultiAgent Systems, Wiley, 2002.

[2] J. M. Alberola, J. M. Such, A. Garcia-Fornes, A. Espinosa, V. Botti, A performance evaluation of three multiagent platforms, Artificial Intelligence Review 34 (2010) 145–176.

[3] G. Piolle, Y. Demazeau, J. Caelen, Privacy management in user-centred multi-agent systems, in: G. O'Hare, A. Ricci, M. O'Grady, O. Dikenelli (Eds.), Engineering Societies in the Agents World VII, Vol. 4457 of LNCS, Springer Berlin / Heidelberg, 2007, pp. 354–367.

[4] J. M. Such, A. Espinosa, A. Garcia-Fornes, A survey of privacy in multi-agent systems, Knowledge Engineering Review (2012) In press.

[5] M. Fasli, On agent technology for e-commerce: trust, security and legal issues, Knowledge Engineering Review 22 (1) (2007) 3–35.

[6] C. Sierra, Agent-mediated electronic commerce, Autonomous Agents and Multi-Agent Systems 9 (3) (2004) 285–301.

[7] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, D. Zabowski, Experience with a learning personal assistant, Communications of the ACM 37 (7) (1994) 80–91.

[8] E. Weitnauer, N. Thomas, F. Rabe, S. Kopp, Intelligent agents living in social virtual environments bringing max into second life, in: Intelligent Virtual Agents (IVA), Springer Berlin / Heidelberg, 2008, pp. 552–553.

[9] M. Montaner, B. López, J. De La Rosa, A taxonomy of recommender agents on the internet, Artificial intelligence review 19 (4) (2003) 285–330.

[10] S. Warren, L. Brandeis, The right to privacy, Hardvard Law Review 4 (5).

[11] A. Acquisti, S. Gritzalis, C. Lambrinoudakis, S. di Vimercati (Eds.), Digital Privacy: Theory, Technologies, and Practices, Auerbach Publications, 2008.
[12] A. Westin, Privacy and Freedom, New York Atheneum, 1967.
[13] K. Rannenberg, D. Royer, A. Deuker (Eds.), The Future of Identity in the Information Society: Challenges and Opportunities, Springer Publishing Company, Incorporated, 2009.
[14] B. Schermer, Software agents, surveillance, and the right to privacy: a legislative framework for agent-enabled surveillance, Amsterdam Univ Pr, 2007.
[15] S. Fischer-Hübner, H. Hedbom, Benefits of privacy-enhancing identity management, Asia-Pacific Business Review 10 (4) (2008) 36–52.
[16] S. Garfinkel, Database nation: the death of privacy in the 21st century, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.
[17] J. Borking, B. Van Eck, P. Siepel, D. Bedrijf, Intelligent software agents: Turning a privacy threat into a privacy protector, Registratiekamer, The Hague, 1999.
[18] H. Taylor, Most People Are "Privacy Pragmatists" Who, While Concerned about Privacy, Will Sometimes Trade It Off for Other Benefits., Harris Interactive, 2003.
[19] D. Hoffman, T. Novak, M. Peralta, Building consumer trust online, Communications of the ACM 42 (4) (1999) 80–85.
[20] D. Solove, A taxonomy of privacy, University of Pennsylvania Law Review 154 (3) (2006) 477–560.
[21] W. Stallings, Network security essentials : applications and standards, Prentice Hall, 2010.
[22] B.-J. Koops, R. Leenes, Identity theft, identity fraud and/or identity-related crime, Datenschutz und Datensicherheit - DuD 30 (2006) 553–556.
[23] L. Bilge, T. Strufe, D. Balzarotti, E. Kirda, All your contacts are belong to us: automated identity theft attacks on social networks, in: Proceedings of the 18th international conference on World wide web (WWW), ACM, New York, NY, USA, 2009, pp. 551–560.
[24] M. Petkovic, W. Jonker (Eds.), Security, Privacy and Trust in Modern Data Management (Data-Centric Systems and Applications), Springer-Verlag, 2007.
[25] M. Stamp, Information Security: Principles and Practice, Wiley-Interscience, 2006.
[26] C. Neuman, T. Yu, S. Hartman, K. Raeburn, The Kerberos Network Authentication Service (V5), no. 4120 in Request for Comments, IETF, 2005.
[27] A. Frier, P. Karlton, P. Kocher, The secure socket layer, Tech. Rep. MSU-CSE-00-2, Netscape Communications (1996).
[28] T. Dierks, C. Allen, The tls protocol version 1.0, RFC 2246 (1999).
URL `http://www.ietf.org/rfc/rfc2246.txt`
[29] S. Spiekermann, L. F. Cranor, Engineering privacy, IEEE Transactions on Software Engineering 35 (1) (2009) 67–82.
[30] M. Hildebrandt, S. Gutwirth, Profiling the European Citizen: Cross-Disciplinary Perspectives, Springer Publishing Company, Inc., 2008.
[31] A. Odlyzko, Privacy, economics, and price discrimination on the internet, in: Proceedings of the 5th international conference on Electronic commerce (ICEC), ACM, New York, NY, USA, 2003, pp. 355–366.
[32] S. Spiekermann, Individual price discriminaton - an impossibility?, in: Proceedings of the Workshop on Privacy and Personalization held with the International Conference for Human-Computer Interaction (CHI), 2006, pp. 47–52.
[33] H. J. Smith, S. J. Milberg, Information privacy: measuring individuals' concerns about organizational practices, MIS Quarterly 20 (1996) 167–196.
[34] D. Chaum, Security without identification: transaction systems to make big brother obsolete, Commun. ACM 28 (1985) 1030–1044.
[35] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, v0.34 (Aug. 2010).
[36] D. Chaum, A. Fiat, M. Naor, Untraceable electronic cash, in: Proceedings on Advances in cryptology(CRYPTO), Springer-Verlag New York, Inc., New York, NY, USA, 1990, pp. 319–327.
[37] E. Aïmeur, G. Brassard, F. Onana, Secure anonymous physical delivery, IADIS International Journal on WWW/Internet 4 (1) (2006) 55–59.
[38] A. Bhargav-Spantzel, J. Camenisch, T. Gross, D. Sommer, User centricity: A taxonomy and open issues, J. Comput. Secur. 15 (2007) 493–527.
[39] M. Bishop, Computer Security: Art and Science, Addison-Wesley, 2002.
[40] S. Chopra, L. White, Artificial agents - personhood in law and philosophy, in: Proceedings of The 13th European Conference on Artificial Intelligence (ECAI), IOS press,, 2004, pp. 635–639.
[41] T. Balke, T. Eymann, The conclusion of contracts by software agents in the eyes of the law, in: Proc. of The 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), IFAAMAS, 2008, pp. 771–778.
[42] M. Fasli, Agent Technology For E-Commerce, John Wiley & Sons, 2007.
[43] M. Hansen, P. Berlich, J. Camenisch, S. Clau, A. Pfitzmann, M. Waidner, Privacy-enhancing identity management, Information Security Technical Report 9 (1) (2004) 35 – 44.
[44] I. Pinyol, J. Sabater-Mir, Computational trust and reputation models for open multi-agent systems: a review, Artificial Intelligence Review (2011) In press. DOI 10.1007/s10462–011–9277–z.
[45] J. Sabater, C. Sierra, Review on computational trust and reputation models, Artificial Intelligence Review 24 (2005) 33–60.
[46] E. Carrara, G. Hogben, Reputation-based systems: a security analysis, ENISA Position Paper (2007).
[47] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decis. Support Syst. 43 (2) (2007) 618–644.
[48] A. Jøsang, J. Golbeck, Challenges for Robust Trust and Reputation Systems, in: Proceedings of the 5th International Workshop on Security and Trust Management (STM), 2009, pp. 1–12.
[49] E. J. Friedman, P. Resnick, The social cost of cheap pseudonyms, Journal of Economics and Management Strategy 10 (1998) 173–199.
[50] J. M. Such, A. Espinosa, A. Garcia-Fornes, V. Botti, Par-

tial identities as a foundation for trust and reputation, Engineering Applications of Artificial Intelligence 24 (7) (2011) 1128–1136.

[51] M. Luck, P. McBurney, O. Shehory, S. Willmott, Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing), AgentLink, 2005.

[52] S. Vinoski, Advanced message queuing protocol, IEEE Internet Computing 10 (6) (2006) 87–89. doi:http://doi.ieeecomputersociety.org/10.1109/MIC.2006.116.

[53] FIPA, FIPA ACL Message Structure Specification, FIPA (2001).
URL http://www.fipa.org/specs/fipa00061/

[54] S. Clau$\beta$, D. Kesdogan, T. Kölsch, Privacy enhancing identity management: protection against re-identification and profiling, in: Proceedings of the workshop on Digital identity management (DIM), ACM, New York, NY, USA, 2005, pp. 84–93.

[55] R. Aydoan, P. Yolum, Learning opponents preferences for effective negotiation: an approach based on concept learning, Autonomous Agents and Multi-Agent Systems (2010) 1–3710.1007/s10458-010-9147-0.
URL http://dx.doi.org/10.1007/s10458-010-9147-0

[56] E. Serrano, M. Rovatsos, J. Botia, Mining qualitative context models from multiagent interactions (extended abstract), in: Proceedings of the tenth international joint conference on Autonomous agents and multiagent systems (AAMAS), IFAAMAS, 2011, pp. 1215–1216.

[57] J. Urbano, A. P. Rocha, E. Oliveira, Computing confidence values: Does trust dynamics matter?, in: EPIA '09: Proceedings of the 14th Portuguese Conference on Artificial Intelligence, Springer-Verlag, 2009, pp. 520–531.

[58] T. D. Huynh, N. R. Jennings, N. R. Shadbolt, An integrated trust and reputation model for open multi-agent systems, Autonomous Agents and Multi-Agent Systems 13 (2) (2006) 119–154.

[59] JADE Board, Jade security guide, http://jade.tilab.com (2005).

[60] J. M. Such, J. M. Alberola, A. Espinosa, A. Garcia-Fornes, A Group-oriented Secure Multiagent Platform, Software: Practice and Experience 41 (11) (2011) 1289–1302.

[61] T. B. Quillinan, M. Warnier, M. Oey, R. Timmer, F. Brazier, Enforcing security in the agentscape middleware, in: Proceedings of the workshop on Middleware security (MidSec), ACM, New York, NY, USA, 2008, pp. 25–30.

[62] S. Ugurlu, N. Erdogan, An overview of secmap secure mobile agent platform, in: Proceedings of Second International Workshop on Safety and Security in Multiagent Systems, 2005.

[63] H. Xu, S. M. Shatz, Adk: An agent development kit based on a formal design model for multi-agent systems, Journal of Automated Software Engineering 10 (2003) 337–365.

[64] A. E. Newman, Cougaar developers' guide, http://www.cougaar.org (2004).

[65] V. Roth, M. Jalali-Sohi, Concepts and architecture of a security-centric mobile agent server, in: Proceedings of the Fifth International Symposium on Autonomous Decentralized Systems (ISADS), IEEE Computer Society, Washington, DC, USA, 2001, pp. 435–444.

[66] Recursion Software Inc., Voyager security guide, http://www.recursionsw.com/ (2008).

[67] M. Warnier, F. Brazier, Anonymity services for multiagent systems, Web Intelligence and Agent Systems 8 (2) (2010) 219–232.

[68] H.-H. Lee, M. Stamp, An agent-based privacy-enhancing model, Inf. Manag. Comput. Security 16 (3) (2008) 305–319.

[69] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Comput. Surv. 42 (2009) 1:1–1:31. doi:http://doi.acm.org/10.1145/1592451.1592452.
URL http://doi.acm.org/10.1145/1592451.1592452

[70] H. Yu, M. Kaminsky, P. B. Gibbons, A. Flaxman, Sybilguard: defending against sybil attacks via social networks, in: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), ACM, New York, NY, USA, 2006, pp. 267–278.

[71] A. Cheng, E. Friedman, Sybilproof reputation mechanisms, in: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, P2PECON '05, ACM, New York, NY, USA, 2005, pp. 128–132. doi:http://doi.acm.org/10.1145/1080192.1080202.
URL http://doi.acm.org/10.1145/1080192.1080202

[72] A. Krause, E. Horvitz, A utility-theoretic approach to privacy and personalization, in: Proceedings of the 23rd national conference on Artificial intelligence (AAAI), AAAI Press, 2008, pp. 1181–1188.

[73] J. M. Such, A. Espinosa, A. Garcia-Fornes, C. Sierra, Self-disclosure decision making based on intimacy and privacy, Information Sciences (2012) DOI: 10.1016/j.ins.2012.05.003.