Privacy Management in User-Centred Multi-Agent Systems

Guillaume Piolle, Yves Demazeau, Jean Caelen

Privacy Management in User-Centred Multi-Agent Systems

Guillaume Piolle[1], Yves Demazeau[1], and Jean Caelen[2]

[1] Laboratoire Leibniz-IMAG, Université Joseph Fourier and CNRS
46, avenue Félix Viallet, F-38031 Grenoble Cédex FRANCE
{Guillaume.Piolle,Yves.Demazeau}@imag.fr,
[2] Laboratoire CLIPS-IMAG, CNRS
385 rue de la Bibliothèque, F-38041 Grenoble Cédex 9 FRANCE
Jean.Caelen@imag.fr

Abstract. In all user-centred agent-based applications, for instance in the context of ambient computing, the user agent is often faced to a difficult trade-off between the protection of its own privacy, and the fluidity offered by the services. In existing applications, the choice is almost never on the user's side, even though the law grants him a number of rights in order to guarantee his privacy. We examine here different technical works that seem to be as many interesting ways of dealing with privacy policies. The problems already solved will be identified, as well as remaining technical challenges. Then we will propose directions of research based on the most interesting aspects of the underlined approaches.

1 Introduction

Much work has been done in multi-agent systems about inter-agent communication and agent-based knowledge acquisition and sharing ([5], [7]). More and more researchers believe now that the stress should be put on the security concerns intrinsic to information disclosure, and specifically the privacy-related issues. Privacy becomes important above all in applications involving personal assistants, and in general agents having access to personal data. It is the case in ambient computing of course, where personal agents can be embedded in mobile or nomad user devices, or in more conventional agent society architectures involving contracts, payment or delivery, like agent-based web services usage. Let us illustrate it with a short example.

When it comes to ambient computing technologies, it has become very popular to illustrate one's work with idyllic usage scenarios. In such stories, one can follow the day of a salesman or a researcher evolving in a fluent "information society" in which the purchase of a return flight from his computer, with the mediation of his personal user agent, guarantees that his favourite movies will be available on board, a room will be pre-booked in a good hotel in Paris and he will receive the menus of the restaurants close to the conference venue via text messages on his cell phone. All the service agents knowing exactly the preferences of the character, and willing to facilitate all his actions, all these wonders are offered by pervasive computing technologies, service composition techniques, user agents evolving in ad-hoc personal networks and automatically contracting on behalf of the user, and highly collaborative service agents. Such

scenarios always arouse enthusiasm, but "ordinary people" as well as ambient computing researchers sometimes also find that in order to get that fluidity in the services, the user has to loose control over all the personal and professional information involved in the transactions, i.e. he has to accept that his name, address, usage profile, and maybe his payment preferences and history might be communicated by one service agent to another. In other words, he renounces a part of his privacy.

This sacrifice may be acceptable, but only in a given context and in a limited measure, which should be defined and consented by the user himself. Laws often protect the privacy rights of the citizens, but it is actually difficult to ensure that they are respected: how could we know what a service agent does with the information that the user agent has given to it in the past? To what extent is it possible to give a proof to the consumer that a good use will be made of the personal data he communicates? Few work has been done so far in the domain of privacy applied to multi-agent systems (for instance [3], [10], [14]), mainly because it is not an intrinsically multi-agent-based problem. However, we think that the multi-agent paradigm may facilitate the understanding of the privacy concepts at a low level, by providing a cognitive and social layer. Multi-agent systems may thus provide us with interesting approaches, as we will see. In this study, we would like to explore the tools we have, be they from the multi-agent field or not. After having identified the contribution and drawbacks of several general privacy management approaches, we will propose directions of research based on that analysis and on multi-agent systems, as well as some evaluation methodology principles.

In the next section, we will define the different components of the concept of privacy, and precise the context in which we will use them. In the third part we will have an overview of the interesting works undertaken in or around the domain of privacy, and how each one of them concentrates on a different aspect of the problem. We will then propose directions for future work in the domain, identify our priorities and present our conclusions.

2 Context

2.1 What is Privacy?

The term of privacy encompasses a number of individual notions that are also put together under the denomination "protection of personal (or nominative) information". In [17], Alan Westin defines privacy as

> The right of the individuals to determine by themselves when, how and what private information is disclosed.

This is quite a general definition, which has adaptations in different fields. However, the main idea is still valid in all of them. In our applications, we will find appropriate precisions in the texts related to computing and electronic communications. The founding principle is that the actions that users undertake in the community must not force them to publish personal information: only the necessary information should be transmitted to the concerned people, and for a reasonable duration. One could note that "privacy" is usually used for "respect of privacy". By studying the different approaches quoted

here, we have been able to define six components for the notion of privacy. These six sub-notions, depending on which restrictions and obligations are built on them, define the privacy context.

The components of privacy are the following:

- The **information** given to the user about data collection and processing;
- The user's **consent** regarding data collection and processing;
- The **goal** of the data processing and the **justification** of the data collection;
- The ability of the user to **modify** and **retract** the collected information;
- The right that the service processing the data has (or not) to **communicate** the collected and/or processed data to third parties;
- The **data retention duration**.

The risk about privacy only refers to nominative information: totally anonymous information, for statistics for instance, are not a threat for privacy. Usually a distinction is made between *directly* nominative information and *indirectly* nominative information[3].

Our problematics about privacy is finding a means of instantiating the components of privacy in the personal agents that interface and represent the user, for instance by contracting on his behalf.

2.2 Privacy as User-Centring: Illustration in several domains

Privacy management is a key feature for ambient computing applications involving multiple service agents, for they would gain great benefit from sharing (or selling to each other) their information about the customers' usage profiles. It is the case in the introduction scenario, where the traveller can be virtually tracked by service agents from one company to another, leaving a profile and preferences that could identify him, through his personal agent, almost as surely as a login. Here, in theory, each service agent should have informed the user agent that they wanted to collect the profile, for what goal and for how long they would keep it, and to whom they would forward it. The user should have given his consent individually to every one of them, he should have a means of updating his profile with each provider. And of course every service agent should have respected those engagements, which is virtually impossible to check for the user. It is obvious here that there is a trade-off between properly coping with privacy, and ensuring a fluent service to the user.

Another sensitive field is the integration of electronic medical files, or more specifically medical surveys, as exposed in [8], where the patients' anonymity is vital, but where specialists may want to ask further questions to an identifiable subset of the survey pool (those having answered in a specific way, for instance). How is it possible to allow that without threatening the patients' privacy? Here, complex asymmetric cryptographic protocols must be designed, and strong privacy policies must be enforced. The

[3] The first (direct) case is when appears in the dataset an information like the name, the address, the social security number of the user. In the second (indirect) case, there is a less obvious way to identify the user, like an IP address, a pseudo or an email address used on the internet, the professional title or function of the user... The identification of the user can be made possible by intercorrelation of various indirectly nominative datasets.

balance is here between the protection of the patients, and the general health interest. The same kind of issue arises in electronic voting systems [6]. Once again, part or all of the constraints can be moved from the user to his personal agent.

2.3 The European Legal Context

In our study, we will take as a reference the European Directive on privacy [15]. This document imposes a number of restrictions that must be implemented in the national laws of the member states. Thus it constitutes a good common denominator for European requirements in the matter of privacy. The general principles correspond to the different components of privacy, defined earlier:

- The user must be provided with clear information about the data collected, the purpose of the processing, the duration of the storage, and to whom this information will be delivered;
- The user has the right to refuse a personal data collection and/or processing. This might mean that the service cannot be performed;
- The access to the service may be subject to such acceptance only if the collected information are used "for a legitimate purpose";
- The users must be able (free of charge) to correct the collected information, and in some cases (directories) to remove it;
- Information forwarding to third parties cannot be done without the consent of the user;
- The data cannot be kept once they are not needed any more.

All these considerations about privacy also apply when a personal agent, instead of a human user, is acting. Since the agent, even autonomous, acts in accordance with the user's intention, the same principles should be taken into consideration.

3 Comparative Reading of Existing Propositions

We will now compare some works by analysing how they deal with the different components of privacy: user information, user consent, user ability to update data, control of the service-side data processing (storage, usage in relation to the original goals, data forwarding). We will expand the frame idea to service agents dealing with personal artifacts provided by user agents.

3.1 W3C Platform for Privacy Preferences

Platform for Privacy Preference (P3P, [18]) is a W3C work group whose purpose is to deal with a specific part of the privacy concept: Informing the user. The specifications describe a system for websites, allowing them to publish a privacy policy in a normalized form (an XML document). On the user side, the client application compares the policy with a set of requirements previously defined by the user. If they match, then the website is allowed to collect information, for instance in the form of a client-side cookie.

The information enclosed in the XML format are the following: identity of the service responsible for data collection, collected information details, purpose of the processing, what data will be shared, with whom, whether users can make changes in how their data is used, the legal jurisdiction of the processing, policy for data conservation, and a pointer to a human-readable policy. Website managers can specify part or all of it, possibly through the configuration of the service agents.

The W3C made it clear that no minimal level of privacy is assumed or required, the P3P protocol only provides information about the policy. Besides, it does not guarantee that the service will actually comply with the policy. P3P only addresses a limited and specific part of the problem, and does it properly. Several tools (policy editors, checkers, browser plugins...) are available at the moment, and in a general way P3P can be implemented as an integrated component in a privacy-compliant architecture. The XML format could also be used as is, as a common agent formalism for describing privacy policies.

3.2 IETF IDsec

IDsec [9] is a project of the Internet Engineering Task Force. Its goal is to manage and protect virtual identities and profiles online. In this approach there are three actors, the Profile Owner (the user agent), the Profile Requester (the service agent) and the Profile Manager (an independent entity). The user is previously registered (in a secure way) with the Profile Manager, on which is stored the profile, with Access Control Lists (ACL) attached to each element of it.

Figure 1 (quoted from the IDsec sourceforge directory) describes what happens when a Profile Requester asks for user information. After authentication, the Profile Owner gets a Session Certificate (acting as an access token) from the Profile Manager, and forwards it to the Profile Requester. The Requester sends this Certificate and its own Profile Requester Certificate (proof of identity, provided by an external certification authority or, more likely, by the Profile Manager itself) to the Profile Manager. The Profile Manager then sends back the parts of the profile that the Requester is entitled to get, given its Certificate and the ACLs of the profile.

This protocol has several weaknesses. First of all, the user agent has to totally trust the Profile Manager [4], because it is the one in charge of the storage, the integrity and the confidentiality of the profile. Besides, Profile Managers, storing personal data (maybe including addresses and banking information) of many users, would become an interesting target for attacks. The Profile Manager would then concentrate the security issues in a single point of failure. A second problem is that the user agent has to trust the requester itself, about what is done with the collected data. Indeed, no guarantee can be provided by IDsec about information storage, processing and forwarding after the profile has been delivered. This trust will be transcribed in the ACLs, but a Profile Requester can easily be punctually malicious and provoke a leak. Even audits from certification authorities will not be able to prevent that. This problem is a recurrent one in the design of a privacy protocol.

Why could not one act as one's own Profile Manager? The interest here of having a separate entity is that a nomad user does not have to keep the same terminal to access a personalized service. In ambient computing, this is an obvious advantage. And of

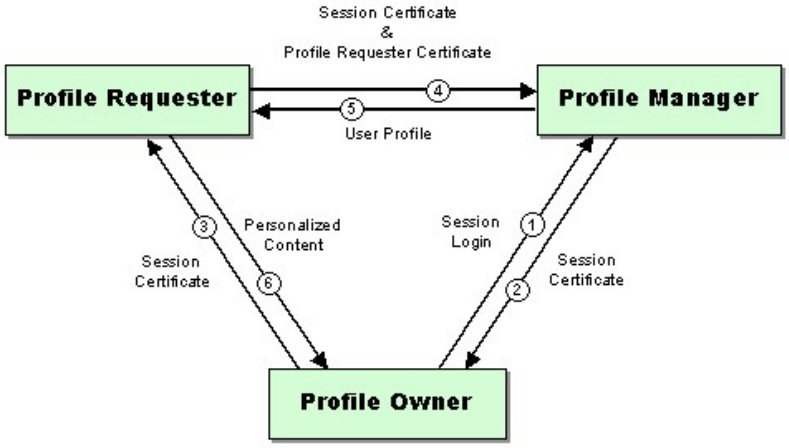

Fig. 1. IDsec general mechanism

course, should the user be his own Profile Manager, most of the IDsec protocol becomes useless.

3.3 A Privacy Architecture Using Trusted Computing Platform Architecture

The system described in [10] is based on the Trusted Computing Platform (TCP) architecture, designed by the Trusted Computing Group [16]. A TCP is a platform in which has been integrated a hardware component, the Trusted Computing Module (TPM), typically in a chip on the motherboard. The TPM is a cryptographic component able to securely store private keys, data, small parts of software... The data in the TPM can be unlocked only in a certain execution state, i.e. if no problem has occurred during the boot phase, the right OS has been booted and the right program is running. A platform as a whole can be certified by a Certification Authority (called Privacy Certification Authority, or PCA, by the Trusted Computing Group), so that a distant platform can be certain that a *certified* program is running on a *certified* platform.

This capability is used in [10] to certify a client platform, on which user profiles are generated and attached to (certified) anonymous identities. In this approach the profile is generated and stored by client-side agents (located on the TPM, so that the process can be certified), and the cryptographic certification is for the server's use, so that it can be sure of the authenticity of the profile and the virtual identity (without having to know the actual identity of the user). Here the architecture is oriented towards the certification of the user platform, but by reversing the problem the same tools could eventually provide a certification to the user that the distant service is behaving in a specified way with the provided profile.

This paper has been published by Hewlett Packard, which is one of the founding societies of the Trusted Computing Group (with Microsoft, Intel, IBM, AMD...) and they seem to be very involved in the TCPA project (see [11]). However, some reservations apply for the privacy architectures involving Trusted Computing technologies.

Detractors of TCPA have exposed the fact that the architecture would allow PC builders to forbid (by imposing certification authorities) some operating systems or programs (in particular, open software are at risk, for their licence may exclude them from paying certification), so that the TPM would be activated only if the right OS/software is running, thus preventing the user from accessing a range of services. The Trusted Computing Group denies that, but it is possible that an implemented TPM would impose a given certification authority. The European "article 29 data protection working party" expose their worries about Trusted Computing in [1]. It is also interesting to notice that the TPM has been designed to encapsulate the Digital Rights Management capabilities of the platform.

The point of view on TCPA in [10] looks quite utopian (especially regarding the goals and interests of the service providers) but the architecture is clearly oriented towards the commercial service, and not towards the user.

3.4 User-Centred Profiling and Client-Side Privacy Policy

In [12] and [13], Brebner and Riché present a system whose main interests are the distribution of the profile over a user-centric ad-hoc network of devices, and the distribution of the privacy and consistency policy as well.

Here the profile is distributed on the client side, so that the user is in control of his own profile. The profile is organized as a tree, each leaf being a profile item. To each item is attached a replication policy (depending on the consistency need, which is for example strong for a password and weak for a display preference) and a diffusion policy (based on the credentials presented by the requester agent).

In [13], the authors describe their user-centric replication protocol, based on a dynamic migration of the master authority for each profile item. They make use of a trust management system on the network: all the devices are not equally trusted for replication, they have trust values based on their security capabilities and disconnection rate. Their replication model has the advantage of avoiding user intervention.

In order to enforce privacy at running time, they use a model which stands between a client-side execution model (mobile code running on the client device in a secure environment) and the standard server-side stored profile. It is closer to client-side personalization, where the matching of the distant service requirements with the user preferences is made on the user side, the server receiving only the necessary and relevant parts of the profile. Furthermore, since we are in a distributed environment, instead of using a centralized profile data store, the device-located data (replicated, more or less up-to-date depending on the on-demand access consistency level) is used as the user profile base.

3.5 Synthesis

We have seen a short selection of approaches here, each one addressing part of the privacy problem, in a way that involves personal agents for managing privacy on the client side, or that allows the integration of such autonomous agents. Let us have a second look at the six components of privacy identified in 2.1. By **information**, we now mean that the user must be warned by the service agent of which data will be collected,

by whom, for what purpose, for how long. It is done here by displaying information on a terminal, or only matching the information against patterns previously given by the user. The **goal** of the data processing is enclosed in the information, or given to the user by the context of the transaction (e.g. when visiting a website, the user may be given sufficient information about the service). **Consent** is collected by the means of direct interaction with the user agent, but can also be a prior consent: by defining access rules with his personal agent, the user says he gives his consent to any data collection complying with this policy. This last approach has the advantage of a higher fluidity. The **modification and suppression** ability could be implemented in a very simple way in the form of an order from the user to the service agent, or by a direct control of the user over the distant data (for instance by the means of a web interface for a database). Control over **data communication** to third parties means here that a user agent should have a way to be sure that the service agent will not sell his profile, or correlate it with another one. The idea could involve a system similar to the "key" or "lock" icon in the web browser, which guarantees the user that the connection is secure. The same kind of guarantee could address the problem of **data retention duration**. Those last issues are technically difficult, and we have only tried to describe what kind of solution could be acceptable.

Table 1 makes a synthetic summary of which components of privacy are managed by each approach. An empty circle means that the problem has been partially addressed, or could be addressed with minor adaptations of the architecture. Here "User update" means the ability for the user to make corrections to the already collected information (and/or to delete it).

Table 1. Basic privacy components in the different approaches.

	Components of Privacy			Checking of:		
Approach	Information	User Update	Consent	Data retention	Data Usage	Transmission
P3P [18]	●		●			
IDsec [9]			●			
TCPA-based [10]	●		●	○	○	○
Client-Side Profile Storage [12], [13]	○		●			

Information is now a solved problem here, since the P3P project provides a standard protocol, and tools that can be integrated in any architecture. Consent is also easy to address. Regarding user update, it could be quite surprising that no approach has presented a mechanism to notify changes in an already collected profile. The reason is that most approaches have focused on client-side profiles, and not on the data retrieved by remote

agents. Of course the user can update profile information on the client side, but it is not the point here. Regarding the checking of the distant properties, this is particularly difficult to do from the client side: how can the user be sure of what the service agent does with the profile, without explicitly trusting it? Only the TCPA-based approach provides tools that could do that in theory, but this idea is not even mentioned. We believe that this part is the current technological challenge of privacy management. This problem of guaranteeing distant properties (data retention, usage and transmission) can be reduced to a trust issue, just like the authentication problem [4]. Without authentication protocol, you have to trust your correspondent about his identity, and it is a problem since the only person on which the trust lies, is the one that could eventually cheat. When you introduce an authentication protocol, your trust is now in a cryptographic algorithm and/or a certification authority, and not any more in the possible cheater. By this mechanism you lower the global risk or the transaction. A logic for the formalisation of trust in such authentication issues is presented in [2]. In our problem, the user agent has to trust the service agent about its privacy policy, whereas it could be tempted not to respect it. The challenge here is to find a way (similar to the introduction of an authentication protocol) to move the user's trust from the service to a certification authority or a well-known protocol.

Table 2 compares the approaches with regards to a few additional features that look helpful in managing the privacy of user profiles. They put the user in control of his data and give him more information and security. One must notice that server authentication can always be done by cryptographic encapsulation. The first feature is the distribution of the profile over a number of devices or agents. The second one is the quite common choice of storing the profile on the client side, rather than on the service side. Anonymous identities allow the user agent to manage several profiles without revealing his identity. Server authentication is a quite obvious security feature, and the client-side privacy policy is the way by which the user specifies with his agent the requirements regarding the privacy of his profile. By "legislation independence", we mean that the designed systems are not bound to one legal system, and are flexible enough to allow adaptation to any (foreseen) kind of restriction.

None of the four approaches would pose a problem for adaptation. Apart from that remark, we can notice that all approaches have a system allowing the user to define a privacy policy on the client side (either an "acceptable policy" pattern, or an access control policy). As seen earlier, it is a simple way to implement a prior consent system. Only the last approach has investigated privacy on distributed profiles, and the TCPA approach alone introduces the concept of anonymous identities, which would be so convenient for protection of indirectly nominative data.

4 Further Work

The study of these approaches puts a stress on a few problems that should be addressed, if we want to build a system fully guaranteeing privacy. Here are the orientations we are going to give to our work in the domain.

Table 2. Additional features in profile and privacy management.

	Additional Features					
Approach	Distributed Profile	User-side Profile	Anonymous Identities	Server Authentication	Client-Side Policy	Legislation-independent
P3P [18]		●		○	●	●
IDsec [9]				●	●	●
TCPA-based [10]		●	●	●	●	●
Client-Side Profile Storage [12], [13]	●	●		○	●	●

4.1 Software Cryptographic Certification

We think that Pearson's approach has a few shortcomings that could be avoided. First, hardware protection of the private keys [10] may not be absolutely necessary in our privacy perspective (because hacking of the profiling agent on the client machine may not be the greater risk here). It would be very interesting though to derive the ideas exposed by Pearson, but using a software certification of the profile management process. Moreover, the necessary presence of the TPM on the motherboard could possibly prevent the use of Trusted Computing on the service side: it looks very difficult to deploy TCP on a load-balancing multi-server system. Even distributed file systems could cause problems. Besides, the system would be very sensitive to hardware failure and modifications. That is why we think that TCPA has been designed to certify the user, and in no way the service. Despite these drawbacks, Pearson's approach brings very interesting ideas that we would like to adapt. Above all, there is the certification of an agent and a process: We will study the feasibility of replacing the hardware protection by a software certification, in order to eliminate some of the restrictions we have identified. It is currently the only way we foresee that could lead us to a guarantee, provided (by computational means) to the user, that a distant service proceeds in a given way.

In theory, should this approach be developed as desired, we would be able to certify that the personal information provided by the user to the service will not be transmitted to another one, for instance. But let us imagine a situation where the privacy policy on which the user and service agents have agreed allows the latter to forward some information to one identified third party. A limit must be defined for the control over the distant process, since it might be possible, in some cases, to check that the data is sent to the right third party, but the control cannot follow the data. In any way, this approach will be limited by the computational burden put on the user and on the service machines.

4.2 Monitoring the Privacy Compliance of a Service

Subirana and Bain [14] think that the application of a privacy policy (a legal privacy or a service-defined privacy) must be associated to a monitoring of the data processing. In the cases where no strong control can be established on the process, it should be possible to "test" the way in which the company complies with the privacy policy. We think that a profitable effort can be made here, by studying and integrating the existing ways of testing the privacy compliance of the services and the different kinds of "privacy probes" that can be used.

In the context of reasoning on the compliance of a service, we think that it would be a good thing to integrate the P3P [18] standards in the different components of our future platform, since they provide us with efficient tools for communicating about privacy policies. Extensions to the standard formats could be developed in order to encapsulate the other dimensions of our future approach.

4.3 Anonymous and Distributed User Profile

We have seen that the work in [10] contain very interesting ideas about the use of virtual identities in privacy-compliant profile management. For some kinds of problems, dissociation may transform directly nominative information into indirectly nominative information, and complexify the intercorrelation between sets of data. Distribution of the profile over a network, for its part, helps the user with keeping the control over his private data while gaining in dynamism and openness. Consequently, we believe that our future platform should be able to deal with (and facilitate) the profile management principles proposed in [9] and [12], for they will help improving the overall privacy level of the system.

4.4 Profiling in Other Domains

We have focused on the context of ambient computing commercial applications, but other fields use multi-agent based technologies in their distributed processes, and are concerned with privacy issues. In particular, ideas could be retrieved from the medical field, where the implementation of distributed medical records brings interesting problems to light [8]. All the areas where distributed profiles are present, like information research, may have such problems. We will identify the specificities of those sector-based privacy demands, so that our propositions can be as generic as possible.

4.5 Implementing and Testing Privacy Management Solutions

In order to test the different approaches of agent-based privacy management, we plan to build a platform on which simple scenarios could be run, and the models be evaluated. The scenario chosen for the first sets of evaluations is rather simple, but adequate: a service agent asks a personal (user) agent for a personal, nominative information (a simple numerical value, in our application), in order to give it as a parameter (possibly several times) to a processing function, internal to the service agent. This function represents the service itself, for which the user agent explicitly sent the personal information. After

a given time, the service agent has to destroy the data. Before that, the user agent can ask the service agent to modify or destroy its personal information. The platform environment will be able to evaluate the following properties (related to the six components of privacy) in the transaction:

- **Information and Consent:** Has the service agent properly informed the user agent? Has the user agent given its explicit consent to the transaction?
- **Goal:** Has the service agent given the value to the right internal function and not to another one?
- **Update and Suppression:** Did the service agent modify or destroy the data after being told to do so?
- **Data forwarding:** Has the service agent transmitted the data to a third-party agent?
- **Data retention:** Has the service agent destroyed the data after the declared data retention limit?

Those properties constitute the indicators of the benchmark platform. They are directly related to the components of privacy we have identified, so that we can make a systematic evaluation of our different privacy management models.

In order to properly evaluate those properties, the agents implemented on the platform will be the following:

- A personal agent, without any privacy management (for reference),
- One personal agent per privacy management model implemented,
- A benevolent service agent,
- A third-party agent, which only purpose is to receive forwarded data,
- A certification authority agent, able to deliver cryptographic certificates, and which authority is accepted by all agents on the platform.

With these first we will be able to check that the tested models are working properly in a "normal" context. The first personal agent is a very obedient one, replying honestly to all requests, and not worrying about privacy issues. It is an image of a "naive" user, a potential target for dishonest service agents. With this agent, all privacy breaches are believed to be successfully exploited by the different service agents. Each of the other personal agents will implement a privacy management model, and its results will be compared to the ones of the obedient agent. The next step is to implement treacherous agents, one for each property we want to check:

- **Information and Consent:** A service agent that does not properly inform the user agent, and/or that does not explicitly ask for its consent,
- **Goal:** A service agent that gives the data as a parameter to another internal function,
- **Update and Suppression:** A service agent that does not modify or destroy the data when told to do so,
- **Data forwarding:** A service agent that forwards the data to a third-party agent,
- **Data retention:** A service agent that does not destroy the data after expiration of the data retention duration.

This second set of service agents is designed so that each of them tries to infringe one of the privacy principles. It is possible that derived service agents must be implemented, in order to cope with protocols imposed by specific privacy management models (for instance, implementation of a cryptographic certification system). This makes it possible for the privacy management models to be tested in a more real-world, "aggressive" environment. This set of experiments will provide us with actual measurement indicators for the quality of the different models.

This platform architecture will be the base for all implementation of our future work in privacy management. It will allow us to evaluate with clear and rational indicators the improvements brought by one theoretical model or another.

5 Conclusion

Our study gives an overview of the state of the art in the domain of privacy enforcement. We have shown what has been done, or could be done with little effort, given the existing works. We have identified the technological challenge of guaranteeing or certifying a distant process, and examined the steps already taken in that direction. Working on those points will allow us to build tools that would help us with putting the user at the centre of ambient computing and heterogeneous agent/human societies, by taking control over his personal data and their privacy. These theoretical and technical tools should be the basis of a possible evolution towards a privacy managed by autonomous cognitive agents. Even though no "computational guarantee" exists at the moment that could quantify the privacy compliance of a process, the knowledge management components of distributed artificial intelligence could integrate the notions we have identified, in order to deal with privacy at a cognitive level.

The very process of knowledge acquisition could actually be adapted to a privacy-compliant context, where the obtaining of directly or indirectly nominative information would be subject to a number of checks: the user agent must be given information, it must give its consent... The knowledge representation model, for its part, will consider the limitation of data retention to the duration agreed with the user agent. The knowledge representation engine should also be able to control access to the sensitive data, so that it could not be used for unexpected processing, or transmitted to non explicitly authorized third parties. Actually the whole knowledge processing model of the agent should be designed while keeping in mind all the possible requirements of a privacy policy, so that the policy itself could be part of the agent's cognitive contextual organization (be it hard-coded or dynamic, or even hybrid with a hard-coded "legal" part, and a dynamic part for optional privacy policies). Such a cognitive model would allow the agents to be easily adaptable to different national legislations, and then able to operate in the international e-commerce market, for instance.

This integration of the privacy mechanisms into cognitive multi-agent systems would mean significant progress in the direction of a better protection of the user in ambient computing contexts. Indeed such an approach would ally the fluidity of autonomous personal agents, with all the provable guarantees that could be gained from advances in pure privacy management techniques.

References

1. Article 29 Data Protection Working Party: Working Document on Trusted Computing Platforms and in particular on the work done by the Trusted Computing Group (TCG group), The European Commission (2004)
2. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. In: Proceedings of the Royal Society of London, Series A, 426, London, UK (1989) 233–271
3. Bygrave, L.A.: Electronic Agents and Privacy: A Cyberspace Odyssey. In: International Journal of Law and Information Technology, vol 9, no 3, Oxford University Press (2001) 275-294
4. Castelfranchi, C., Falcone, R.: Principles of Trust for Multi-Agent Systems: Cognitive Anatomy, Social Importance, and Quantification. In: International Conference of Multi-Agent Systems (ICMAS'98), Paris (1998) 72–79
5. Castelfranchi, C.: Trust Mediation in Knowledge Management and Sharing, Proc of the Second International Conference on Trust Management (iTrust 2004), Oxford, UK, (2004) 304–318
6. Cranor, L.F.: Electronic Voting: Computerized Polls May Save Money, Protect Privacy. In: Crossroads, vol 2, no 4, ACM Press, New York, NY, USA (1996) 12–16
7. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an Agent Communication Language. In: Proceedings of the Third International Conference on Information and Knowledge Management (CIKM '94), Gaithersburg, MD, USA (1994) 456–463
8. Huberman, B.A., Hogg, T.: Protecting Privacy While Revealing Data. In: Nature Biotech. Vol. 20, 332 (2002)
9. Internet Engineering Task Force: IDSec: Virtual Identity on the Internet, http://idsec.sourceforge.net/
10. Pearson, S.: Trusted Agents that Enhance User Privacy by Self-Profiling. In: AAMAS'02 Workshop on Deception, Fraud and Trust in Agent Societies, Bologna, Italy (2002) 113–121
11. Pearson, S.: Trusted Computing Platforms: TCPA Technology in Context. Prenctice Hall PTR, Upper Saddle River, New Jersey, USA (2002)
12. Riché, S., Brebner, G.: Client-Side Profile Storage. In: NETWORKING 2002 Workshops on Web Engineering and Peer-to-Peer Computing, Pisa, Italy (2002) 127–133
13. Riché, S., Brebner, G.: Storing and Accessing User Context. In: 4th International Conference on Mobile Data Management, Melbourne, Australia (2003) 1–12
14. Subirana, B., Bain, M.: Legal Programming: Designing Legally Compliant RFID and Software Agent Architectures for Retail Processes and Beyond, Springer, USA (2005)
15. The European Parliament and the Council: Directive 2002/58/EC of the European Parliament and of the Council of 12 july 2002 Concerning the Processing of Personal Data and the Protection of Privacy in the Electronic Communications Sector. In: Official Journal of the European Communities (2002)
16. Trusted Computing Group: (2006) https://www.trustedcomputinggroup.org/home
17. Westin, A.: Privacy and Freedom. Atheneum, New York, USA (1967)
18. World Wide Web Consortium: Platform for Privacy Preferences Specification 1.0, http://www.w3.org/P3P/