Trust Management Model and Architecture for Context-Aware Service Platforms

Ricardo Neisse[1,*], Maarten Wegdam[1], Marten van Sinderen[1], and Gabriele Lenzini[2]

[1] CTIT, University of Twente, The Netherlands
{R.Neisse, M.Wegdam, M.J.vanSinderen}@utwente.nl
[2] Telematica Instituut, Enschede, The Netherlands
Gabriele.Lenzini@telin.nl

Abstract. The entities participating in a context-aware service platform need to establish and manage trust relationships in order to assert different trust aspects including identity provisioning, privacy enforcement, and context information provisioning. Current trust management models address these trust aspects individually when in fact they are dependent on each other. In this paper we identify and analyze the trust relationships in a context-aware service platform and propose an integrated trust management model that supports quantification of trust for different trust aspects. Our model addresses a set of trust aspects that is relevant for our target context-aware service platform and is extensible with other trust aspects. We propose to calculate a resulting trust value for context-aware services, which considers the dependencies between the different trust aspects, and aims to support the users in the selection of the more trustworthy services. In this calculation we target two types of user goals: one with high priority in privacy enforcement (privacy concerned) and one with high priority in the service adaptation (service concerned). Based on our trust model we have designed a distributed trust management architecture and implemented a proof of concept prototype.

1 Introduction

Context-aware services use context information to adapt themselves to the current situation. Adaptive service provisioning offers compelling business opportunities (e.g., personalization of offers and control of the quality of service) and new technological challenges, such as, for example, the management of context information in order to not violate the user's privacy preferences.

In order to reach a widespread success, context-aware services must be trustworthy. The trustworthiness of a context-aware service depends on the trust relationships among the entities, such as, service, identity and context providers, that cooperate during the service provisioning. For example, users of context-aware services may not accept that privacy sensitive [1] context information is released if they do not trust the service providers receiving the information; service providers

* Supported by CNPq Scholarship – Brazil.

R. Meersman and Z. Tari et al. (Eds.): OTM 2007, Part II, LNCS 4804, pp. 1803–1820, 2007.

may, in turn, demand trustworthy context providers in order to ensure that the context information has the minimum required quality for service adaptation [2]; finally, context providers may request trustworthy identity providers to ensure that the retrieved context information corresponds to the correct identity. The trust of a user in the context-aware service depends on all these trust relationships, and the trust relationships also depends on each other, e.g., the trust in the context provider depends on the trust in the identity provider that identifies the context provider.

Existing trust models propose special-purpose solutions that are not easily portable to our context-aware domain because they either specify incomplete trust relationships related to at most one trust aspect (e.g., enforcement of access control procedures [5], integrity of identities [6], or the enforcement of privacy policies [7]) or make no distinction between different trust aspects because users need to trust a centralized service as a whole, for instance, in the way it is done by e-bay [8].

We define a new trust management model for context-aware service platforms that explicitly addresses trust relationships for different trust aspects and their interdependencies. We identify and analyze a set of interconnected trust relationships related to specific trust aspects that satisfy the trust requirements of our target context-aware service platform (the Freeband AWARENESS service platform [9]). Our trust management model, or trust model for short, addresses a basic set of trust aspects related to *identity provisioning*, *privacy enforcement*, and *context provisioning* activities. This list is not exhaustive and can be extended with other trust aspects when needed. Our model supports both *direct trust* resulting from direct experience and *indirect trust* derived from trust calculations, for example, based on recommendations from other entities.

Our trust model evaluates the trust users have in a context-aware service by taking into account the interdependencies between the three different trust aspects that we consider. Based on specific user goals, the trust values in the privacy enforcement and context provisioning aspects have different weights in the resulting trust in the service. Based on [10] we address two types of user goals: one demanding with higher priority the enforcement of his/her privacy rules and the second one demanding with higher priority the service adaptation. With the calculation of a resulting trust value from the trust values for different trust aspects we want to assist users in the selection of more trustworthy context-aware services.

In our target context-aware service platform [9], it is not acceptable that one central entity is responsible for the management of the trust relationships for all other entities, because different administrative domains may be involved. Each and every administrative domain has its own components and management infrastructure and, for this reason, we also propose a distributed trust management architecture. Our trust management architecture instantiates our trust model and is currently implemented in a peer-to-peer prototype using JXTA [11]. We present the current proof-of-concept implementation of our trust model which uses the Subjective Logic [12] API for trust calculations. In the prototype the user can select his goal (privacy enforcement or service adaptation) and see the resulting trust value for the available context-aware services.

This paper is structured as follows: Section 2 gives an overview of our target context-aware service platform identifying entities, roles, and trust relationships. Section 3 presents our trust management model and an algorithm for the combination

of trust values regarding different trust aspects. Section 4 presents our architecture for distributed trust management and our prototype implementation. Section 5 compares our work to the state of the art on trust for distributed, pervasive, and context-aware service platforms. Section 6 summarizes our conclusions and provides a discussion on future work.

2 Trust Relationships in a Context-Aware Service Platform

Figure 1 presents our target context-aware service platform [9] and illustrates the five roles we distinguish in it, namely *users*, *context owners*, *identity providers*, *context providers*, and *service providers*.

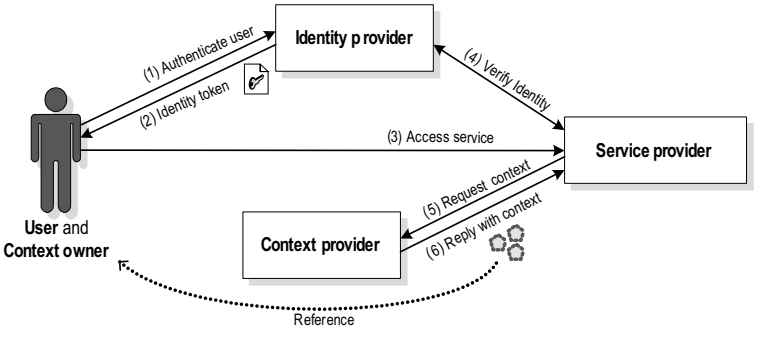

Fig. 1. Roles in a context-aware service platform and their interactions when a user accesses a service provider. User and context owner roles are played by the same entity.

The arrows in Figure 1 indicate the basic interactions between the roles when a user accesses a service provider. First the user authenticates with an identity provider (1) and receives an identity token (2). After the authentication is done, the user can access the service provider (3), where the service will verify the identity token of the user (4). To be able to adapt the service to the user's context, the service provider retrieves context information about the user (here also the context owner) from a context provider (5 and 6). This information can be, for instance, the current activity or location of the user; however, it can also include context information about other entities that are relevant for the context-aware service being used.

In our context-aware service platform roles are dynamically assigned during the service provisioning. In a particular scenario, it is possible that multiple entities play the same role, and that one entity plays more than one role; for example, a person holding a GPS device might play at the same time the user, the context owner and context provider roles.

2.1 Analysis of Trust Aspects

In our service platform trust is a critical issue. The context owner must trust the context provider and the service provider, because they are going to manage his/her

context information. The context owner will demand its context information to be released only when his privacy policies allow such, and he will only accept his context information to be managed if he trusts that both context providers and service providers are able and willing to adhere to his/her privacy policies.

In addition, the user and the service provider trust the context provider regarding the provisioning of context information. This trust aspect is important to guarantee that this information is provided with the required quality characteristics and consequently resulting in the expected context-aware service adaptation. Trust in the context provider from the service provider point of view is also required in case dynamic security policies based in context information are used (e.g. [13]), which may require additional security verifications in case untrustworthy context information is received. An example of additional security verifications could be, for example, redundant check of context obtained from different context providers.

Finally, all the entities have trust relationships with an identity provider because they present and receive credentials (issued by the identity provider) in order to identify themselves to other entities in the service platform. Even though Figure 1 presents (for simplicity) only user authentication and identity verification, with only one identity provider (arrows 1 and 4), also context owners, service providers, context providers, and identity providers themselves should provide identity credentials when interacting with other entities. Even so, it is not required for all the entities to be authenticated with the same identity provider.

Figure 2 depicts the trust relationships among the user, the context-owner, the identity provider, the service provider, and the context provider, namely *identity provisioning, privacy enforcement* and *context provisioning* trust relationships. These relationships are interpersonal relationships where each of them has an entity that sets up the trust relationship with another entity, called respectively the Trustor and the Trustee [3]. This set of trust relationships is by no means exhaustive; other trust relationships targeting different aspects can be identified if different scenarios would be considered. Our objective here is to propose a basic set, based on our target service platform, and motivate the definition of different trust relationships for different trust aspects, including their dependencies.

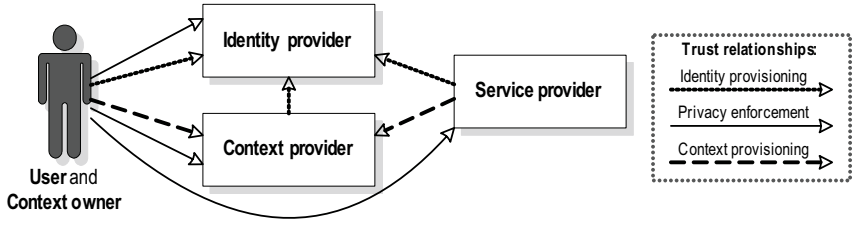

Fig. 2. Trust relationships in context-aware service platforms for different aspects

For each type of trust relationships presented in Figure 2 it is possible to establish a trust value according to certain aspect-specific metric. The following subsection present metrics for obtaining trust values related to identity provisioning, privacy enforcement and context information provisioning.

2.2 Metrics for Obtaining Trust Values

This section discusses existing metrics that can be used to quantify the amount of trust for each type of trust relationship.

Identity Provisioning. One metric that influences the identity provisioning trust is the authentication method. Identity providers that use very strong biometric authentication should be more trusted than others that use only username/password authentication. It is also possible to associate the identity provisioning trust value with a specific session, according to the type of authentication used for that session, in case the identity provider supports more then one type of authentication method. The user registration policy also influences the identity provisioning trust. Identity providers that allow users to freely register without verifying the identity of the user (e.g. Google and Yahoo) may not be trusted as much as identity providers that do not allow free registration, such as a university or a bank.

Privacy Enforcement. Trust in privacy enforcement depends upon the existence of privacy policies in the context provider and service provider (e.g. P3P policies [7]), which state how the context owner's data will be handled. These privacy policies should be compared with the context owner's privacy preferences and, in case they match, it is assumed that the privacy expectations will be followed. The following metrics have also been proposed by [14] and [7] to calculate trust values regarding privacy enforcement aspects: user interest in sharing, confidentiality level of the information, number of positive previous experiences, number of arbitrary hops, a priori probability of distrusting, and service popularity in search engines. The number of arbitrary hops is related with identities issues and the chain of certificate authorities between the source and the target of the information. Privacy enforcement trust values can be also obtained from trusted third parties specialized in privacy protection issues. Privacy protection organizations take care of privacy policies certification in the same way identities are certified today by certification authorities [15]. We foresee that privacy recommendations will be provided by informal organizations such as virtual users' communities and customer protection organizations.

Context Information Provisioning. The trust in the context providers can be evaluated, for example, through cryptographic mechanisms based on PKI (identity coupled) and through the following metrics and mechanisms: reputation of context provider, statistical analysis of context information provided from the source, and context aggregators that compare redundant information from different sources in order to increase trustworthiness. It is also possible to evaluate the trust of the context information based in the trustworthiness of the quality aspects [2] of one particular instance of context, or in the method used to obtain the information. One example is location information, which trustworthiness may vary depending on how the information is obtained: from outlook calendars, user personal GPS position, or position of the GSM/WiFi base station to which the user is connected.

3 Trust Management Model for Context-Aware Service Platforms

After motivating the need for the different trust aspects in our context-aware service platform in the previous section, this section discusses an algorithm to measure and combine trust for each trust aspect relevant in our architecture. Well-known concepts like trust establishment, direct and indirect trust, and recommendations are instantiated to match the trust requirements of our service platform. We show how the trust values related to different aspects can be combined into an overall trustworthiness evaluation of the context-aware service from the user point of view. Here, we restrict our analysis to two user-profile perspectives: the first one with higher priority on privacy enforcement and the second one with higher priority on the service adaptation. This section ends with a discussion on the integration of trust recommendations in our trust model.

3.1 Formalization of Aspect-Specific Trust Relationships

Many models for trust management exist (e.g., see [3][4][16] and Section 5 of this paper). Most of these models refer to a specific application domain and, as such, propose special-purpose solutions that are not easily portable to other domains: our context-aware domain requires a specific formalism of combining trust aspects we have not found treated appropriately in the literature. Despite we have not researched in this direction, we do not exclude that existing formalisms for trust (e.g., [17]) can be extended to express and combine multiple trust aspects as it is required by our domain.

As widely accepted, we formalize trust as a relationship between two entities, the Trustor and the Trustee [3]. In its more general definition [16], a trust relationship represents a subjective measurement of belief from a Trustor concerned with a certain Trustee *behavior* and focused on a certain trust *aspect*. For example, Bob (Trustor) may trust at a high degree Alice (Trustee) for what concerns her competence in coding in Java. The Trustee's *behavior* is part of the social perspective of trust. Trustors can perceive or interpret the Trustee's behavior as an isolated or combined measurement of, for example, honesty, competency, reputation, usability, credibility and reliability. In this paper we consider behavior as "honesty, competence, and reliability". Other behaviors are also important and will be considered in our future work. A list of possible trustee behaviors and their correlations based in user studies can be found in [18]. The trust aspect models different scopes that can be tackled by the trust relationships. As motivated in Subsection 2.1, for our target context-aware service platform we address the following aspects: *identity provisioning*, *privacy enforcement*, and *context information provisioning*. The metrics presented in Subsection 2.2 are examples of how to obtain trust values for these aspects.

Regarding the choice of the domain of trust values, existing trust models have different proposals. Some authors quantify trust as a real numeric value (e.g., between -1 and 1), a discrete value (e.g., trust or distrust), or a combination of both where each element in the discrete set has a numeric equivalent (e.g., values in (0, 1] mean trust, values in [-1, 0) denote distrust, and 0 means unknown).

Our proposal is independent from any particular solution; we assume a generic domain *TValue.* As a matter of example we instantiate *TValue* in the set of opinions

of the Subjective Logic (in short, SL) [12], which supports uncertainty and provides operators to deal with trust opinions calculations, for example, discount, addition and consensus. Accordingly to the SL theory, trust in a certain proposition is expressed with a triple $(b, d, u) \in [0, 1]^3$ that represents respectively the Trustor's subjective belief (b), disbelief (d), and uncertainty (u). With

$$A \xrightarrow[v]{*;\ a} B$$

we indicate a trust relation between A (the Trustor) and B (the Trustee) that tackles on the trust aspect a and that has degree v (see also Figure 1). B here can also represent a category. "$*$" is a place-holder for classes of trust relation. In this paper we will consider two classes of trust relations: *direct functional* (*df*) and *indirect functional* (*if*), so $* \in \{id, if\}$. Direct trust originates from A's direct experiences or evaluations of B. Indirect trust originates when A's resorts to indirect evaluating B's trust, for example, by combining trust values or asking for recommendations from other entities (see also [4]). In our formalism A and B are entities' identities which belongs to a set *ID*. Aspect a ranges over identity provisioning, privacy enforcement, and context information provisioning, that is $a \in \{idp, pe, cip\}$.

Identities are assigned to different roles in different instances of our platform. We consider the set of roles $R = \{US, CO, IP, CP, SP\}$ from our context-aware service platform (Section 2), namely, *user, context owner, identity provider, context provider* and *service provider*. The function $role: ID \rightarrow R$ returns the role that, in the present moment, a given entity identified by an identity ID plays; running this function is of exclusive competence of identity providers, but it can be invoked by any entity that has registered its identity (see Figure 1, arrow 1).

3.2 Trust Evaluation

Abstracting from the actual trust metric evaluation that will be applied in an instance of our platform (for details on metrics for the different trust aspects see Subsection 2.2), we assume that entities can access a set of functions that calculate, respectively, the direct trust value from a Trustor to a Trustee based on the evaluation of its privacy enforcement (*pe*), identity provisioning (*idp*), and context information provisioning (*cip*) qualities. These functions receive as input the Trustor and Trustee identities ($ID \times ID$) and return the trust value for the specific trust aspect:

$$\begin{aligned} trust\_PE:&\ \ ID \times ID \rightarrow TValues \\ trust\_IDP:&\ \ ID \times ID \rightarrow TValues \\ trust\_CIP:&\ \ ID \times ID \rightarrow TValues \end{aligned}$$

For example, *trust_PE(Alice, Bob)* is the evaluation of Bob's honesty, competence, and reliability in its privacy enforcement aspect, from the Alice's view point. Considering the metrics in Subsection 2.2, it is easy to image that Alice provides a trustworthiness profile against which Bob qualities are compared and evaluated. Here we assume a trusted-third party role, the *Trust Provider* whose task is to run those functions on demand and on behalf of Trustors. These functions are our starting point for trust evaluation; on their output we can establish the degree of trust between the

Trustor and the Trustee. If we specify our reasoning in term of an inference system, i.e., in terms of axioms and deductive rules of the form premises/conclusion_the functions we have identified in this section can be used, at a meta-level, to define our set of axioms. In all the following rules, which express our algorithm, we assume that *role(A) = US*, that is the Trustor *A* is a user.

$$\frac{[trust\_PE(A,B)=v]}{A \xrightarrow[v]{df\,;pe} B} \quad role(B) \in \{CP, IP, SP\}$$

$$\frac{[trust\_IDP(A,B)=v]}{A \xrightarrow[v]{df\,;idp} B} \quad role(B)=IP \qquad \frac{[trust\_CIP(A,B)=v]}{A \xrightarrow[v]{df\,;cip} B} \quad role(B)=CP$$

For example, in the first rule when *trust_PE* is invoked with parameters *A* and *B* it returns a value *v*, which states that A has degree *v* of (direct) trust in *B*, with respect the aspect *pe* (privacy enforcement). This aspect is significant when Trustee *B* is a context provider, an identity provider and a service provider. In the following we use the deductive style formalization to depict the main characteristic of our algorithm of trust evaluation and composition.

As we have seen in the previous section, it is the responsibility of the identity provider to provide the identity of the entity that plays a certain role. Moreover, we have defined trust as a relation between identities. We therefore conclude that trust in an identity is influenced by the trust (regarding the trust aspect *idp*) in the identity provider that has provided that identity. The trust value associated with the provider or issuer of the trustee identity influences all the trust values associated with that identity. This reflects the case that it is not possible to trust the trust values associated with some identity that is not trusted. This inter-relation between trust in identities and trust in identity providers is synthesized by the following inference rule for indirect trust:

$$\frac{A \xrightarrow[v]{df\,;a} B \qquad A \xrightarrow[v']{df\,;idp} C}{A \xrightarrow[v' \otimes v]{if\,;a} B} \quad role(C)=IP,\ C \text{ provides } B\text{'s identity}$$

The previous rule express the following: if *A*'s direct trust degree in *B* regarding aspect *a* is *v*, and if the identity of *B* is provided by identity provider *C*, and if *A*'s indirect trust in *C* for aspect identity provisioning is *v′*, then *A*'s indirect trust in *B* regarding aspect *a* is $v' \otimes v$, which represents the value *v* discounted by the value *v′ (e.g.* $v' \otimes v \leq v$*).*In the SL domain set, the ⊗ can be mapped onto the discounting operator.

Once the user has established a trust relationship with all the entities playing the context provider and service provider roles, (trust that as we explained has been influenced by the trust the user has in the identity providers), the user deduce its trust in the role itself. This passage is a generalization step, quite important in our framework, because the user is willing to evaluate its trust in the service considering that the context provide and the service provider roles may be played by more than one entity. The following rules express this generalization step for the context

provider role (*CP*). The rules for the generalization step concerning the service provider role (*SP*) are similar.

$$\frac{A \xrightarrow[v]{if;a} C, \quad [role(C) = CP]}{A \xrightarrow[v]{if;a} [CP, \{C\}]} \qquad \frac{A \xrightarrow[v]{if;a} C \quad A \xrightarrow[v']{if;a} [CP, \boldsymbol{S}], \quad [role(C) = CP]}{A \xrightarrow[v \oplus v]{if;a} [CP, \boldsymbol{S} \cup \{C\}]} \; C \notin S$$

Here $a \neq idp$, because identity provisioning has been already in place. The rule on the left says that *A*'s trust in the *CP* role, can be initiated with the trust *A* has on one members of the *CP* role. The rule on the right says that new members can contribute to the A's trust in the *CP* role; so if *A*'s trust in the role *CP* is *v,* and if *A*'s trust in the member C is *v'*, then the new *A*'s trust in the role is $v \oplus v'$. Here $v \oplus v'$ expresses a "fair" combination of the two trust values as, for example, SL consensus operator.

We are now ready to the final step of our algorithm, which consist in evaluating the user's trust in a context-aware service. It depends on trust he/she has on both the roles *CP* and *SP* regarding privacy and context provisioning aspects, and where the context provisioning aspect is only influenced by *CP*. We assume two different user profiles, the first one with higher priority in *the privacy enforcement* and which will accept to have less service adaptation, and the second one with higher priority on *context-aware service adaptation* even if his/her privacy is not respected [10]. We name these two profiles privacy focused and service focused users. The rule that express how to calculate *A's* (user) trust on a service provider *B*, when context provider role is played by entities in $\boldsymbol{S}$, is formalized as follows:

$$\frac{A \xrightarrow[v]{if;pe} B \qquad A \xrightarrow[v']{if;cip} [CP, \boldsymbol{S}]}{A \xrightarrow[f(v,v')]{if;pe \times cip} B \times [CP, \boldsymbol{S}]} \; role(B) = SP$$

Here the user combines his trust in the service in the privacy enforcement aspect, and the trust he has in the context provider role in the context provisioning aspect. Function *f* expresses a particular way of aggregating trust, which depends on the user profile. In the following we are going to consider two use profiles, the first one focusing on privacy enforcing aspect and the second one on service provisioning aspect.

In order to give an example of *f*, and for illustration purposes, we map *TValues* into the ordered set {VT, T, U, VU} whose elements model judgment of user perspectives: *very untrustworthy* (VU), *untrustworthy* (U), *trustworthy* (T), and *very trustworthy* (VT). We assume that VT > T > U > VU. Figure 3 depicts an example of how to identify user judgments in our domain of reference. An opinion *v* whose belief is higher than disbelief, is considered trustworthy (i.e., $v \in$ T) if it has uncertainty not lower than 1/3 and very trustworthy (i.e., $v \in$ VT) otherwise. One opinion *v* whose belief is not higher than disbelief, is considered untrustworthy (i.e., $v \in$ U) if it has uncertainty not lower than 1/3 and very trustworthy (i.e., $v \in$ VU) otherwise.

An example of function *f* can be obtained by first applying π to *v* and *v'*, then applying one of the function of Figure 4, and then by mapping back each user category onto a "representative" opinion of that category. For example a representative opinion of VT can be the triple (0.75; 0.01; 0.24), of T can be (0.50;

0.01; 0.49), and so on. To the best of our knowledge, functions with the properties sketched in Figure 4 cannot be obtained by composing existing SL operators with π.

Informally, Figure 4 shows the resulting trust in the service when the trust expectation in the service provider regarding the privacy enforcement aspect and the trust expectation in the context provider regarding the context information provisioning increase. The best case scenario for both user profiles is the one where the trust expectations for both privacy enforcement and context information provisioning trust aspects at least trustworthy.

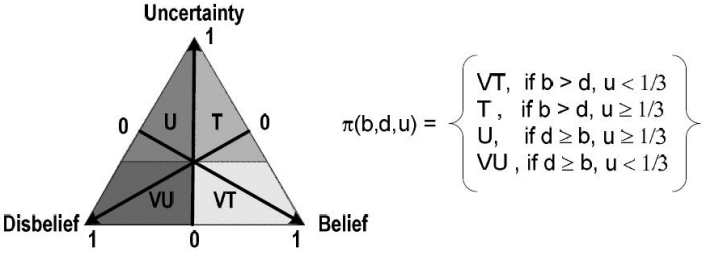

Fig. 3. The function π: $[0, 1]^3 \rightarrow$ {VT, T, U, VU} that maps a SL opinion onto the set of user judgments

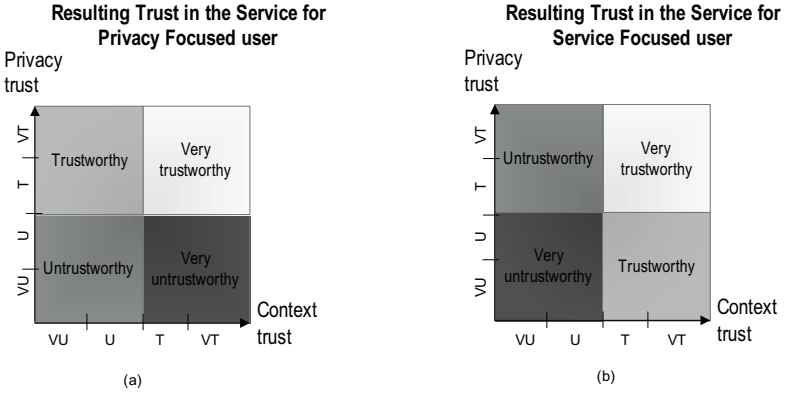

Fig. 4. Resulting trust in the service accordingly to a user profile that focuses on privacy (a) and on a user profile that focuses on service (b)

Accordingly to Figure 4 (a), for the privacy focused profile the best cases are when the privacy enforcement is at least trustworthy. The worst cases is when the privacy enforcement is untrustworthy, because is more likely that trustworthy context information about the user will be under a privacy risk. For the service focused profile (Figure4 (b)) the best cases are when the context provisioning is at least trustworthy where it is even better when the privacy is also enforced. The worst cases are when the context information is not trustworthy, which results in a bad service adaptation, however in this case it is preferable to have privacy enforcement if possible. We

assume here that a context-aware service receiving untrustworthy context information is more likely to adapt wrongly to the current user situation. From this discussion we support the conclusion that for both user profiles the best case is when trust in the context information and privacy enforcement is high, however, depending in the profile the worst case scenario is not the same.

3.3 Extension of the Basic Algorithm: Recommendations

Since users may interact with entities that are unknown (or whose features are unknown) and with which they have had no previous experiences, we support recommendation management in trust relationships in our model in a similar way to the approach adopted by [19]. By using recommendations (indirect) trust can be established based on information received from other entities. Each entity can have an a priori trust value regarding the recommendation aspect about other entities in the system, stating a level of trust in the recommendations received from that entity.

In order to merge the recommendations received from many entities, for example, we can use the solution proposed in [27]. Here the SL consensus operator is used to merge considering uncertainty in a "fair" way and if entities receive conflicting recommendations this increases the uncertainty in the trust values. This is slightly different from the proposal of [19] where an average function is used and where conflicting recommendations may result in a lack of information about trust. One major drawback of the approach done by [19] is that they do not consider uncertainty, which may result in less accurate trust results when conflicting opinions are combined.

Our recommendation algorithm requires a Trust Provider role (TP), to which identities ask for recommendations. Note that, as discussed in the previous section, TP is expected to receive feedbacks from identities regarding their trust on others, and it is responsible to the synthesis of an overall recommendation. More advanced algorithms to calculate trust from indirect knowledge are presented in [20]. We leave as a future work the formalization and evaluation of trust recommendations exchange in our trust model using the SL consensus operator.

4 Distributed Trust Management Architecture

A context-aware service platform is typically a distributed system without a unique central point of control. In such a system, in some cases implemented in a fully ad-hoc configuration, multiple administrative domains may exist. To illustrate this, consider a weather service which provides for mobile phone users the local weather forecast based on the latitude/longitude of the GSM cell they are in. In this case, the weather service provider, the mobile phone operator, and the user personal devices are examples of different administrative domains controlled by different administrative entities.

In this multi administrative domain scenario it is not possible to have a centralized trust provider responsible for the management of all trust relationships due to privacy and scalability reasons. In order to support distributed management of trust we designed a distributed trust management architecture, which is presented in Figure 5.

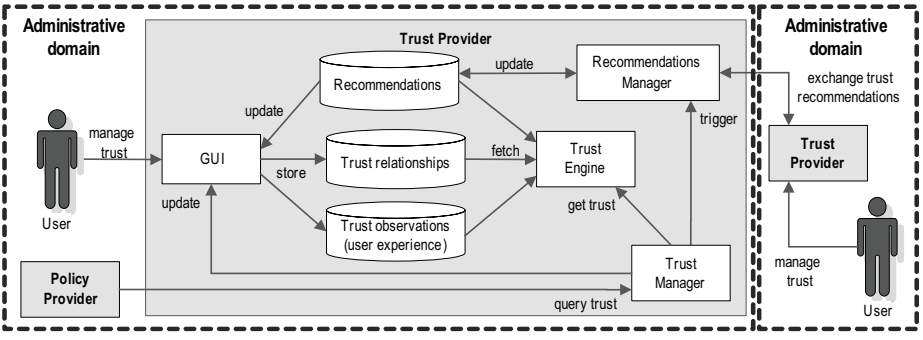

Fig. 5. Distributed Trust Management Architecture

Our architecture supports distributed management of trust considering that each administrative domain has its own trust provider. Using the graphical user interface (GUI) users can visualize and change their trust relationships and also provide feedback of their experiences by informing trust observations. The objective of this trust database is to manually support users in the selection of more trustworthy context-aware services and also provide input for automated policy components where decisions can be automatically taken based on policies that use the trust values in their conditions.

In case trust evidence is not available in one administrative domain, our architecture support the propagation of recommendations requests to other domains, for example, using existing social network connections such as buddy lists. The following section presents our prototype implementation where our trust model and management architecture is currently implemented as a proof of concept.

4.1 Prototype Implementation

We have implemented our trust model and architecture in a proof of concept prototype using the JXTA peer-to-peer library [11] and the Subjective Logic API [12] for trust calculations based on opinions. Figure 6 presents the user agent screen of our prototype where users can visualize in different tabs the context aware services, context providers, and identity providers available in the network. Users can also see their current identity and selected the context providers they want to use from the list of available context providers in the respective tab.

For each entity the interface displays the identity description and a colored representation of the calculated trust. The colors range from dark to light green for trustworthy entities, grey for uncertain, and dark to light red for untrustworthy entities. The colors represent the trust value regarding role specific aspects, for example, in the context providers tab the trust value displayed is the value for the context provisioning trust aspect. We calculate the resulting trust value for each trust aspect considering the dependency with the identity provider that provides the identity of each entity, following our trust model formalism described in Section 3. We use for that the SL discount operator.

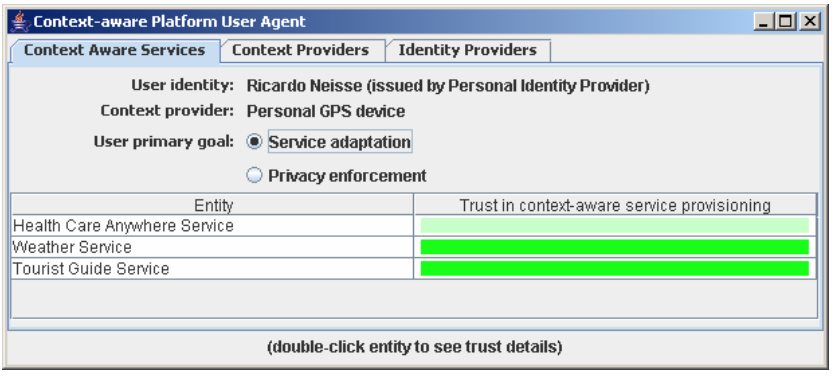

Fig. 6. Visualization of trust for users with high priority in service adaptation

Figure 6 and 7 are examples of the same "context aware services" tab after the user changes his primary goal respectively from "Service adaptation" in Figure 6 to "Privacy enforcement" in Figure 7. In Figure 6, for the user goal "Service adaptation" the resulting trust in the "Health Care Anywhere Service" is trustworthy because the trust in the context provider "Personal GPS device" is very trustworthy. In Figure 7, when the user changes his goal to "Privacy enforcement" the resulting trust in the service became very untrustworthy, because the trust value for this service regarding the privacy enforcement trust aspect is untrustworthy (see Figure 7). The resulting trust in the service changes to very untrustworthy following exactly the function we presented in Figure 4 Section 3.2). This function states that untrustworthy services receiving trustworthy context information are a major privacy risk for privacy concerned users.

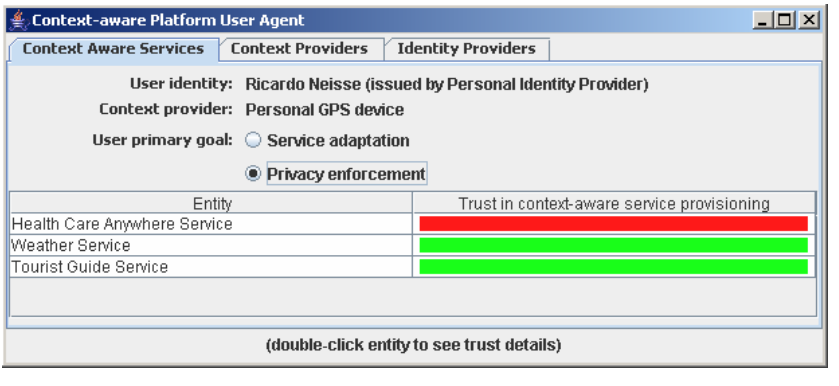

Fig. 7. Visualization of trust for users with high priority in privacy enforcement

Figure 8 presents the "Trust details" screen where users can see and change, after a double-click in an entity, the detailed trust information. In this screen it is possible to see the name of the identity provider that identifies the identity and details about the trust values (including the colored scale). In this screen we do not present the triple

belief (b), disbelief (d), and uncertainty (u) from SL for each trust value. For simplicity we decided to show the SL expectation value, which is a linear representation from 0 to 1 more easily understandable for users of a trust value.

Our prototype uses the JXTA peer-to-peer communication model for publishing and discovering entities in the network however we do not support in the current prototype implementation the exchange of trust recommendations nor user experience reports displayed in our trust architecture. Our next step is to implement the exchange of trust recommendation requests and responses using the SL consensus operator (as described in Subsection 3.3) to merge the trust recommendations responses.

In the current prototype we have also not implemented any metric for direct trust calculation presented in Subsection 2.2, we have arbitrarily defined initial trust values for each aspect and entity in order to illustrate the usefulness of our model. More details about our next research steps are presented together with the conclusions of this paper in Section 6.

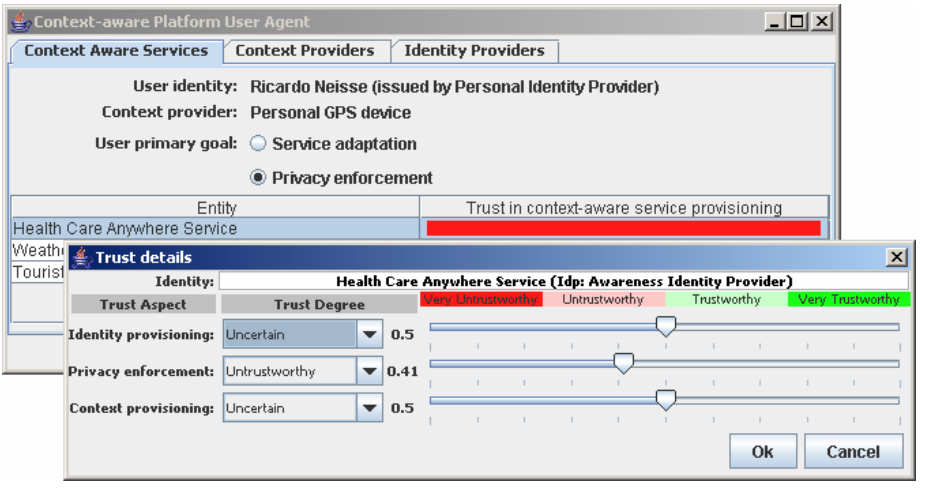

Fig. 8. Visualization of know identities and trust details

5 Related Work

The research on trust can be approached from the *social*, *informational*, and *technical* points of view [10]. For each of these perspectives there are different trust issues that should be addressed, for instance, how users perceive the trust in the system [21] (*social*), what are the concepts and semantics of trust mapped into the system (*informational*), and how secure is the encryption technology used (*technical*). In this paper we are especially interested on the informational level.

Grandinson and Sloman [22] propose a trust specification and analysis framework for internet applications called SULTAN. In SULTAN trust levels are defined from a Trustor perspective for different allowed Trustee actions. SULTAN does not support the combination of different trust levels and does not consider trust for different aspects like identity provisioning.

The Pervasive Trust Model (PTM) of Almenárez et al. [19] applies the concept of trust degrees in the definition of access control policies. They support in their work direct trust by previous knowledge and indirect trust based on recommendations. The final trust degree for an entity is calculated as the average of the recommendations and only recommendations from trusted identities are processed. They do not explicitly support trust quantification for identities and also do not target specifically context-aware service platforms.

A specific approach for trust definition and management for context-aware applications is proposed by Daskapan et al. [13]. In their approach they target privacy aspects and provide a heuristic model to evaluate trustworthiness of context consumers, in order to influence user privacy policy decisions. If the evaluated trust is under a certain threshold then user consent is required, otherwise, the context provider decides automatically, on behalf of the user, whether the context information should be provided or not based on the computed trust value. For Dakaspan et al. trust is a function of the number of previous experiences, the number of hops, and the a priori probability of distrusting the Trustee. Kolari et al. [7] also proposes trust for privacy where trust values are associated with privacy policies in the *Platform for Privacy Preferences* (P3P) format.

Liberty Alliance [23] and MSN passport are examples of identity federation and single sign-on solutions. When using these approaches the authentication task is delegated to trusted identity providers. The authentication information is then communicated through assertions to other entities in the system. These approaches are usually based on Public Key Cryptography and, in spite of being target only to identity issues, are sometimes wrongly applied for other trust aspects. If the identity of some entity is certified this does not mean that the privacy policies or context information provided by this entity can also be trusted.

The relation between context-awareness and trust can also be carrier of new opportunities. Proposals where context information is used as input for trust evaluation can be found in [24][25]. Here, the inference of different levels of trustworthiness of a piece of data depends upon also the currently active context. In [26], the context is explicitly modeled in the trust relationship that might exist between two agents; as such the trust relationship that results is formally contextualized; contextual data, when available, are thus used to guide the process of trust establishment whilst values of trust are assigned to each deduced relationship depending on the availability and on the quality of the context. We consider our approach in this paper as a complimentary solution in comparison with these solutions.

6 Conclusions and Future Work

We have proposed a new trust management model and architecture that supports the quantification of trust for the different trust aspects relevant for our target context-aware service platform. Our model is extensible and considers trust aspects related with identity provisioning, privacy enforcement, and context information provisioning. We identify the dependencies between these trust values and develop a formalism to combine these different trust aspects in order to evaluate the resulting

trust users have in a context-aware service. We address two different resulting trust calculations considering privacy enforcement and service provisioning concerned user goals.

Our contribution in the area of context-aware computing is a trust model that quantifies trust relationships regarding essential trust aspects of our context-aware service platform and calculates the resulting trust users have in a context-aware service by taking into account the interdependencies between these trust relationships. Our trust model is extensible with other trust aspects. In addition, we have also designed and implemented a proof-of-concept prototype of our distributed trust management architecture which implements our model and assists context-aware service users in their trust decisions and selection of more trustworthy context-aware services.

As future work we plan to use context information to improve the recommendation process. For example, context can be used to determine the suitable target entities to request recommendations from. This will allow anonymous and still useful recommendations exchange. Context can also be used to dynamically adapt the user goals. In certain context situations (e.g. health care service) users may not have privacy as first goal when they need the best service adaptation (e.g., to send an ambulance to their current trustworthy location).

Furthermore we will research specific challenges in modeling trust between trust providers from different administrative domains and evaluate the usability and usefulness of our trust model and architecture through user studies in the Freeband AWARENESS project [9] using our prototype implementation. This evaluation will enable us to validate and fine tune our trust model.

Acknowledgements

This work is part of the Freeband AWARENESS project. Freeband is sponsored by the Dutch government under contract BSIK 03025. G. Lenzini has been partially supported by both the Freeband AWARENESS project and the ITEA project Trust4All.

References

[1] Lahlou, S., Langheinrich, M., Röcker, C.: Privacy and Trust Issues with Invisible Computers. Communications of ACM 48(3), 59–60 (2005)

[2] Sheikh, K., Wegdam, M., van Sinderen, M.J.: Middleware Support for Quality of Context in Pervasive Context-Aware Systems. In: Workshop Proc. of the Fifth IEEE Percom (2007)

[3] Grandinson, T., Sloman, M.: A Survey of Trust in Internet Applications. IEEE Communications Surveys (2000)

[4] Jøsang, A., Keser, C., Dimitrakos, T.: Can We Manage Trust? In: iTrust. the Proceedings of the Third International Conference on Trust Management (2005)

[5] Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust management. In: Blaze, M., Feigenbaum, J., Lacy, J. (eds.) IEEE Conference on Security and Privacy, Oakland, CA (May 1996)
[6] Jøsang, A., Fabre, J., Hay, B., Dalziel, J., Pope, S.: Trust Requirements in Identity Management. In: Australasian Information Security Workshop 2005, Newcastle, Australia (January-February 2005)
[7] Kolari, P., et al.: Enhancing P3P Framework through Policies and Trust. UMBC Technical Report, TR-CS-04-13 (September 2004), Available at http://ebiquity.umbc.edu/get/a/publication/ 118.pdf
[8] Trust and Safety in eBay, Available at: http://pages.ebay.com/help/newtoebay/resolving-concerns.html
[9] van Sinderen, M.J., van Halteren, A.T., Wegdam, M., Meeuwissen, H.B., Eertink, E.H.: Supporting Context-aware Mobile Applications: an Infrastructure Approach. IEEE Communication Magazine (September 2006)
[10] Berendt, B., Günther, O., Spiekermann, S.: Privacy in E-Commerce: Stated Preferences vs. Actual Behavior. Communication of the ACM (CACM) 48(3) (2005)
[11] JXTA Java peer-to-peer API. Available at: http://www.jxta.org
[12] Jøsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
[13] Neisse, R., Wegdam, M., van Sinderen, M.J.: Context-Aware Trust Domains. In: Havinga, P., Lijding, M., Meratnia, N., Wegdam, M. (eds.) EuroSSC 2006. LNCS, vol. 4272, pp. 234–237. Springer, Heidelberg (2006)
[14] Daskapan, S., Ali Eldin, A., Wagenaar, R.: Trust in Mobile Context Aware Systems. In: 5th IBIMA. International Business Information Management Conference, Cairo, Egypt (2005)
[15] Pons, A.: Biometric marketing: targeting the online consumer. Communications of ACM Magazine 49(8), 61–65 (2006)
[16] Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: HICSS33. Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii (2000)
[17] Nielsen, M., Krukow, M.: Towards a Formal Notion of Trust. In: PPDP 2003. Proceedings of the 5th ACM SIGPLAN international conference on Principles and Practice of Declarative Programming, ACM Press, New York (2003)
[18] Quinn, K., O'Sullivan, D., Lewis, D., Wade, V.P.: Trust Meta-Policies for Flexible and Dynamic Policy Based Trust Management. In: POLICY 2006, Canada (June 2006)
[19] Almenárez, F., Marín, A., Campo, C., García, C.R.: A Pervasive Trust Management Model for Dynamic Open Environments. In: First Workshop on Pervasive Security and Trust in MobiQuitous, Boston, USA (2004)
[20] Toivonen, S., Lenzini, G., Uusitalo, I.: Context-aware Trustworthiness Evaluation with Indirect Knowledge. In: SWPW 2006. proc. 2nd International Semantic Web Policy Workshop, Athens, USA (2006)
[21] Mayer, R.C, Davis, J.H, Schoorman, D.F.: An Integrative Model of Organizational Trust. The Academy of Management Review 20(3), 709–734 (1995)
[22] Grandinson, T., Sloman, M.: Trust Management Tools for Internet Applications. In: Nixon, P., Terzis, S. (eds.) iTrust 2003. LNCS, vol. 2692, Springer, Heidelberg (2003)
[23] Liberty Identity Federation Framework Architecture Overview, Version 1.2, Liberty Alliance Project. Available at: https://www.projectliberty.org/resources/

[24] Toivonen, S., Denker, G.: The Impact of Context on the Trustworthiness of Communication: An Ontological Approach. In: Workshop on Trust, Security, and Reputation on the Semantic Web at the 3rd ISWC, Japan (November 2004)
[25] Toivonen, S., Lenzini, G., Uusitalo, I.: Context-aware Trust Evaluation Functions for Dynamic Reconfigurable Systems. In: MTW 2006. Workshop on Models of Trust for the Web MTW 2006, Edinburgh, Scotland (2006)
[26] Hulsebosch, R.J., Salden, A.H., Bargh, M.S., Ebben, P.W., Reitsma, J.: Context sensitive access control. In: SACMAT 2005, Stockholm, Sweden (June 2005)
[27] Jøsang, A., Hayward, R., Pope, S.: Trust Network Analysis with Subjective Logic. In: ACSC 2006. Proceedings of the Australasian Computer Science Conference, Hobart (January 2006)