Privacy-Aware Modelling and Distribution of Context Information in Pervasive Service Provision

Ioanna Roussaki[1], Maria Strimpakou[1], Carsten Pils[2], Nikos Kalatzis[1], Martin Neubauer[3], Christian Hauser[3], Miltiades Anagnostou[1]
[1] *School of Electrical and Computer Engineering, National Technical University of Athens, Greece*
{nanario,mstrim,nikosk,miltos}@telecom.ntua.gr
[2] *Telecommunications Software & Systems Group, Waterford Institute of Technology, Ireland*
cpils@tssg.org
[3] *Institute of Communication Networks and Computer Engineering (IKR), University of Stuttgart, Germany*
{neubauer,hauser}@ikr.uni-stuttgart.de

Abstract

Context awareness is an essential cornerstone in future pervasive computing systems. It has the potential to greatly reduce the user attention and interaction bottlenecks, to give humans the impression that services fade into the background, and to support intelligent personalization features. Nevertheless, in order to create such an environment, a growing amount of personal information has to be provided to the system, either manually or automatically. Hence the digital trace and representation users have in the system is getting dangerously detailed, thus stressing the need for privacy protection.

DAIDALOS[1] is a European research project in the area of 3G and beyond, which aims to combine heterogeneous networks in a transparent and seamless way, and develop on top of this a pervasive environment for applications and end-users. This paper describes the main models and mechanisms that have been established to provide federated context-aware services and protect the privacy of their users.

1. Introduction

The vision of a pervasive computing world [1] is gradually gaining momentum and is expected to constitute a worldwide common shared computing paradigm for a plethora of new advanced telecom services. It is the "*third paradigm*" computing, after mainframes and personal computing, where technology recedes into the background, and computing takes place invisibly, everywhere and every time, for everyone, enhancing quality of life in all its arenas.

A pervasive computing environment is saturated with computing and communication capabilities so gracefully integrated with users and their environment that they disappear into the fabric of everyday life [2]. Pervasive systems need to be aware of their environment and associated resources (their context), and must be able to detect changes in the environment and to adapt their functionality and behaviour accordingly [3]. They need to be minimally intrusive and exhibit inherent proactiveness and dynamic adaptability to the current conditions and user preferences & environment. Thus, they have to be context-aware.

Context awareness (CA) [4] provides pervasive environments with the ability to adapt the services or content they provide, by implicitly sensing and automatically deriving the users' needs from the context that surrounds them. CA distinguishes context-aware from traditional applications, in the sense that it makes them more attentive, responsive, predictive and

[1] This work has been partially supported by the Integrated Project DAIDALOS ("Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services"), which is financed by the European Commission under the Sixth Framework Programme. However, this paper expresses the authors' personal views, which are not necessarily those of the DAIDALOS consortium.

1-4244-0237-9/06/$20.00 ©2006 IEEE

aware of the user desires and environment. Here, context is any information that can be used to characterize the situation of an entity [5]. User context may include a wide variety of data collected via sensors such as the current temporal and spatial location, the weather or even the user's biological state, and manually entered information such as user preferences and identity details.

DAIDALOS [6] is an Integrated Project that aims to bring together mobile and broadcast communications and deliver ubiquitous end-to-end services across heterogeneous technologies. DAIDALOS provides a universal and open service platform that can offer pervasive services to application developers in such a way that the underlying network technology becomes fully transparent. One of the main parts of the DAIDALOS Pervasive Service Platform (PSP) [7] is the context management system that establishes the CA functionality of the PSP. Nevertheless, in order for this to be achieved, a considerable amount of dynamic and static personal information needs to be monitored and stored by the system. This situation leads to an increase of users' privacy threats, as personal data is disclosed to unknown providers and a lot of sensitive personal information is to be handled by services and the platform [8]. The latter is especially dangerous if the user's identity is also known. This privacy threat grows stronger as the context information known by the pervasive computing system increases.

This paper is concerned with the CA and the protection of user privacy aspects in pervasive computing environments, and the mechanisms that establish such functionalities within DAIDALOS. The rest of the paper is structured as follows. Section 2 focuses on the formulation of an efficient context model adequate for ambient context management systems in support of pervasive provision. Section 3 elaborates the distribution and federation of context data, while section 4 presents issues that threaten the user's privacy in context-aware environments. Section 5 describes a privacy protection approach that is based on context pseudonymization and is adequate for distributed context management systems. In section 6, an overview of existing privacy protection approaches for context-aware systems is presented. Finally, in section 7 conclusions are drawn, while an outline of the current status and future plans is provided.

2. The context model

Establishing the CA functionality in pervasive service provision is a very challenging task, as it needs to be accomplished in a highly dynamic physical and computing environment. On the one hand, a context management system must be designed such that it addresses the requirements of pervasive computing environments. On the other hand, as there are many different types and natures of context information that are vital for the realization of a fully pervasive system, a flexible, scalable and interoperable context information model needs to be established that supports efficient representation, interpretation, management and dissemination of context data. In this section, the context model, which has been established in the DAIDALOS context management architecture [9], along with its design rationale are described in detail.

The support for adaptive and context-sensitive service provision to users places unprecedented demands for the underlying context framework. One of the key issues for successfully introducing a global context management framework in the service provision chain is the adoption of a clear and consistent context model. Such a model should enable efficient management of context information and allow for feasible context taxonomy and formalism that will fuel context reasoning mechanisms.

In general, a well-designed context model is a key factor in developing context-aware systems. It is a fact that context-aware application development requires significant and careful modelling efforts to ensure that context information is appropriately represented in the target context management system and that applications are able to perform valid manipulations on it [4][10]. This section focuses on providing the basis for constructing a distributed, scalable, extensible and well performing context model in support of fully integrated context-aware services in large-scale pervasive environments. Subsequently, the concepts of the context model designed and developed are introduced.

The main items of the proposed model are the Entity, Attribute, Directed and Undirected Association [11]. These classes formulate the core context model that is further enriched by additional classes that in principal address context management requirements and do not contribute to the formalisation of context information. Thus, their description is out of the scope of this paper. In general, the proposed context model is built upon the notion of an *Entity*, which corresponds to an object of the physical or conceptual world. Entities may demonstrate various properties, e.g., "height", "color", "address", "location", etc., which are represented by Attributes. The *Attribute* class identifies an entity's status in terms of its static and dynamic properties and therefore, it captures all the context information that will be used to characterize the situation of the owner entity. An Entity may be linked to other Entities via *DirectedAssociations*, such as "owns", "uses", "located in", "student of", etc, or *UndirectedAssocations*, such as "friends", "teammates", etc. Directed associations are relationships among entities with different source and target roles. Each directed association originates at a single entity, called the parent entity, and points to one or more entities, called child entities. This modelling concept serves for declaring special associations between entities, where the parent and the children of the association need to be explicitly defined since otherwise the association would be meaningless. Undirected associations do not have an owner entity specified, but form generic associations among peer entities. Each undirected association has at least two participants. All Entities, Attributes and Associations are marked with a timestamp indicating their most recent update time. Our modelling concepts are based on a relational approach and address the notion of entities and their interrelations. For communicating the context model between peer context management systems a XML (de)serialization mechanism is exploited, which provides from-XML and to-XML functionality for entities, attributes and associations.

As already stated, the context types that characterize an entity, attribute or association should be consistent throughout the PSP, no matter where the context is generated or updated. To achieve this, a functional building block has been introduced that acts as a registry containing all valid context types, i.e., all context types that characterize the context knowledge of the pervasive environment.

The proposed relational context model is inspired by the object-oriented and graphical models categories [12], and is implemented as a location-based model [13]. In general, location-based models are used to define spatial relations between locations. In this framework, locations can be determined by a symbolic identifier or by a geometrically defined location. Choosing a suitable location model for the spatial structure of the context model objects is important for distributing, synchronizing and managing context information in an integrated pervasive system. The location-based hierarchical structure selected for the application of the presented context model is described in detail in the next section.

3. Distribution and federation of context data

Virtually any conceivable information could be considered to be context information. Therefore, it is apparent that scalability is crucial for designing context databases in context management systems. We attempt to meet this requirement by making the assumption that context information is, in general, location centric. That is, typically, a context client requests just the context information, which is directly related to its current location. Context clients will rarely access information, like for example outdoor temperature or restaurant menus, of fairly remote areas. To exploit this property, the context management comprises database servers distributed all over the network; each of them being responsible for collecting and maintaining context data that is related to a certain domain, i.e., a preconfigured geographical area or an organization. For example, an organization can be responsible for managing context of its members. According to this precondition, it is assumed that most access requests to a context database originate from its domain and hence, the system meets the scalability requirement.

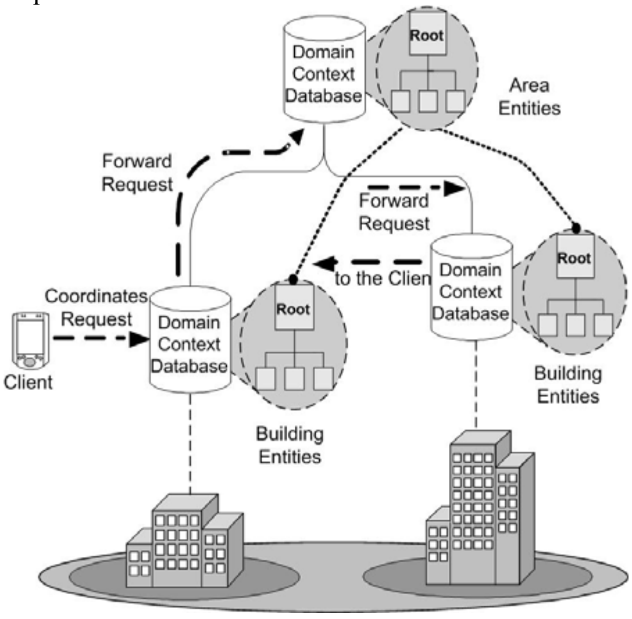
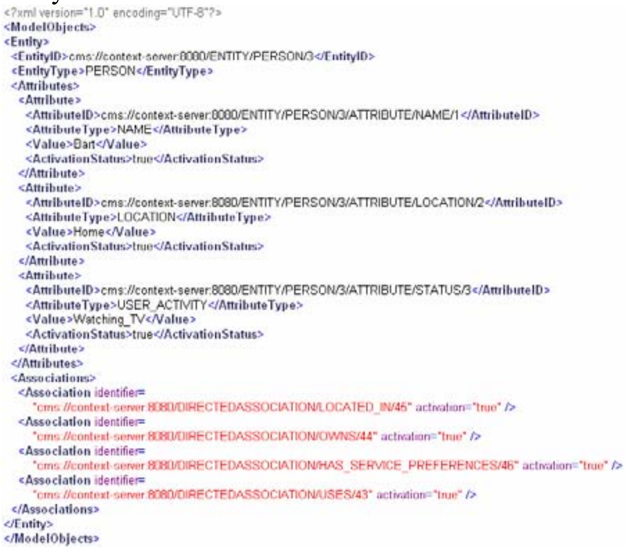


Figure 1. Context database hierarchy

The precondition and the deployment of context databases are also reflected in the logical structure of the databases. In the proposed solution, each context database contains a hierarchy of entities, each of them representing an area of the covered domain. The root node of this hierarchy describes the covered area of the domain. Leaf entities describe either places of minimum granularity, or point to other context databases. Figure 1 shows an example hierarchy. When a client requires an entity that matches a certain coordinate or address, i.e., an entity representing a place that contains the coordinate or address provided, it dispatches its request to the local domain context database. On receiving the request, the database first checks whether the coordinates are within the domain. If not the database forwards the request to a database server, the domain of which covers a larger geographical area. Thus, in a DNS like fashion [14], requests are routed through the hierarchy until they reach a server, the domain of which contains the coordinate provided. From here the request is handed down to the sub-domain server that stores the searched entity.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<ModelObjects>
<Entity>
 <EntityID>cms://context-server:8080/ENTITY/PERSON/3</EntityID>
 <EntityType>PERSON</EntityType>
 <Attributes>
  <Attribute>
   <AttributeID>cms://context-server:8080/ENTITY/PERSON/3/ATTRIBUTE/NAME/1</AttributeID>
   <AttributeType>NAME</AttributeType>
   <Value>Bart</Value>
   <ActivationStatus>true</ActivationStatus>
  </Attribute>
  <Attribute>
   <AttributeID>cms://context-server:8080/ENTITY/PERSON/3/ATTRIBUTE/LOCATION/2</AttributeID>
   <AttributeType>LOCATION</AttributeType>
   <Value>Home</Value>
   <ActivationStatus>true</ActivationStatus>
  </Attribute>
  <Attribute>
   <AttributeID>cms://context-server:8080/ENTITY/PERSON/3/ATTRIBUTE/STATUS/3</AttributeID>
   <AttributeType>USER_ACTIVITY</AttributeType>
   <Value>Watching_TV</Value>
   <ActivationStatus>true</ActivationStatus>
  </Attribute>
 </Attributes>
 <Associations>
  <Association identifier=
    "cms://context-server:8080/DIRECTEDASSOCIATION/LOCATED_IN/45" activation="true" />
  <Association identifier=
    "cms://context-server:8080/DIRECTEDASSOCIATION/OWNS/44" activation="true" />
  <Association identifier=
    "cms://context-server:8080/DIRECTEDASSOCIATION/HAS_SERVICE_PREFERENCES/46" activation="true" />
  <Association identifier=
    "cms://context-server:8080/DIRECTEDASSOCIATION/USES/43" activation="true" />
 </Associations>
</Entity>
</ModelObjects>
```

Figure 2. A person context model example object

Both, entities and associations are identified by URLs. Here a URL contains the address of the responsible Home Manager (HM) that stores the entity. Therefore it is straightforward to retrieve an entity or association when the identifier is known. The format of context identifiers is as follows: cms://hostname:port/[scope]/modelType/type/number. It is built on the following components:

- *Hostname* and *port* specify the host name and port number of the Home Manager that stores the master copy of the specific context model object.
- The *Scope* is valid only for Attribute objects in order to encapsulate information about the owner entity of the specified attribute.
- *ModelType* describes the type of the context model object, and can take one of the following values {empty, entity, attribute, directed association, undirected association}.
- *Type* is the context type used to characterize a context model object.
- *Number* is a unique number identifying an individual context object.

In a nutshell, following the GSM principles, the HM concept has been used to indicate the location of the master context information instance [15]. But the HM acts not only as an entity or association provider, it is also responsible for updating and synchronizing model objects. Basically, each context database can store a copy of a model object. Yet, modifications to replicas must be synchronized with the master copy.

An instance of the context model described in Section 2 that represents a person entity and uses the aforementioned identification scheme is depicted in Figure 2. It uses the XML language and is dynamically translated to the corresponding context model object.

4. Privacy considerations in context-aware systems

The goal of context-aware systems is to support each user best in all situations with specially tailored functionality. Thus, the following two requirements must be addressed. First of all, the system has to recognize the situation the user is in. Second, the system must know the user very well in order to behave according to his wishes. In other words, the system must have a whole set of preferences and attributes of the user. Both requirements above lead to the necessity of personal information (such as the user's situation, his preferences, his service usage history, etc.) to be stored and tracked by the system. Therefore, the user is obliged to disclose personal and hence privacy-sensitive data to the system. This information should per se not be generic so that the user could "hide" in a group of users with same attributes/preferences and situations (see the definition of anonymity set in [16]), but highly specialized for each user in order to tailor the functionality best.

The intended disclosure of personal data is one difference of pervasive computing systems with regard to mobile telephony systems in which user data is concealed. A second difference is that a pervasive system will be in need of a magnitude of providers. This is because providers have to specialize themselves as well as because different providers have to interoperate in order to share expensive infrastructure and in order to offer complex rich services to users.

This situation leads to an increase of threats to the privacy of users. Personal information is disclosed to unknown – thus often not trustworthy – providers and a lot of sensitive personal information is to be handled by services and the platform. The latter is especially dangerous if the user's identity is also known. Roughly stated, the privacy threat grows stronger as the amount of known context information increases.

This is the point, where the privacy protection approach gets a grip. The huge set of personal (context and other) data will be partitioned in smaller sets with much lower sensitivity. For this, a user is enabled to use several pseudonyms, thus building several virtual identities (VIDs) in the system. A VID [8] [17] is a kind of user identity consisting of an artificial name (pseudonym) for the user, augmented by a set of attributes under which the user appears in the system. Each VID only comprises a partition of the user's overall data set, thus resembling a restricted view on the user presented to external parties. This information set is a partial view on the user's overall context hierarchy. Figure 3 shows an example of this approach.

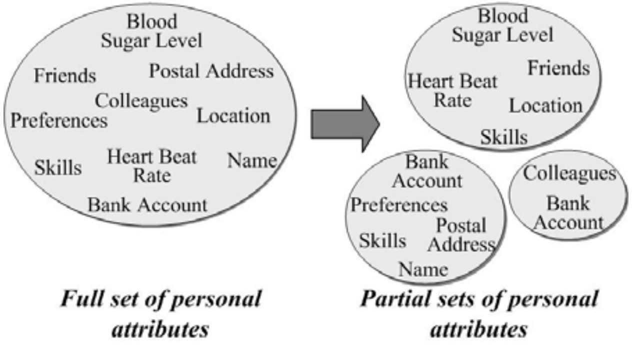


Figure 3. Partitioning of personal information

Another advantage of controlling access to VIDs rather than to plain data is the fact that escalated revealing of knowledge about the real user identity is supported. For example, if people are involved in public affairs, they often need to trade anonymity for a service. This also means that the disclosure of private data is proportional to the number of parties that share the awareness of one only real identity. In our VID approach, involving personal information in public affairs does not necessary mean trading anonymity. This is due to the fact that, in case a virtual identity is carefully selected, then the possibility of revealing a true identity by inspecting the disclosed information is drastically reduced or even eliminated. Of course, each VID still must be accountable for the user's actions – for charging purposes as well as for the case of misuse by the user. How this is realized is described in [18]. It should be mentioned here that the multiple identity approach adopted supports the principle of data minimisation of the European Parliament and Council [19][20], which requires the set of personal data disclosed to a service to exactly match the amount of data the service needs.

As this partitioning of personal data is the user's goal, an attacker – who wants to gain as much information about a user as possible – aims at combining several such data sets, i.e., he wants to link several VIDs of a user in order to merge the associated data. Thus, each component of the pervasive system must be secured against this attack. In the following section, a protection approach against such privacy attacks will be discussed for the context management described in Sections 2 and 3.

Prior to designing a protection approach, the attacker must be determined, against which the system shall be secured. Here, this could basically be the context management itself and/or clients querying for context data. Context management is supposed to be a rather large system serving many users, which will be built up over years in order to collect large amounts of context information about these users. Thus, abuse of its users' data would cause a huge damage to its reputation. It is likely, that such a misuse would disruptively stop the business of the context management system operators. On the other hand, the more users are managed within the context management the bigger is the possible group each user can hide in. This means that the anonymity set [16] increases and the effort for an outside attacker, e.g., to link multiple VIDs to one user, would also increase.

This situation is comparable to today's mobile telephony operators, which also know a considerable amount of personal information about their users, e.g., location, and postal address, and which are large, well-known parties, that would loose their customers in case of a privacy scandal. Those operators are trusted a) not to abuse the personal data and b) not to reveal this data to 3rd parties. There are no special technical security measures in place to protect users against the mobile telephony operators. Due to the organizational similarities, we chose to adopt the same approach and trust the context provider as a large, known player in the pervasive computing world.

So the goal of our proposed approach is to protect the user's VIDs against third parties, which want to access context information from the context management. Thereby, the user is assumed to allow each third party only access to a restricted amount of personal data (the respective partition being visible with a certain VID). Our main goal is to prevent such third parties from linking several VIDs using management information accessible in the context management subsystem and not from the actual values of context items.

5. Privacy considerations in distributed context management

The privacy protection mechanisms applied in the DAIDALOS context management system fall mainly into two categories. On the one hand, an access control mechanism has been established that requires authentication and authorisation verification of the party requesting context information. Once the necessary access rights are in place, the system delivers the requested context data. On the other hand, pseudonymization of context identifiers (used to access the actual context) is applied, the enabling mechanisms of which are the main concern of this section.

Albeit, access control mechanisms can protect context information, they fail in disguising relationships within the context structure. That is, since entities are identified by URLs, clients can use URLs to determine the relation of an entity with other entities in the hierarchy. For example, a client may observe a certain context URL joining a "located-in" association of an entity representing a street. When this association is removed from the street's list of associations the client can still analyse the URL and browse to neighbouring entities. Finally, the client can inspect whether the URL is added to their "located-in" associations. Thereby, the client is able to determine the embedding context structure of this URL and figure out which URLs are related to each other.

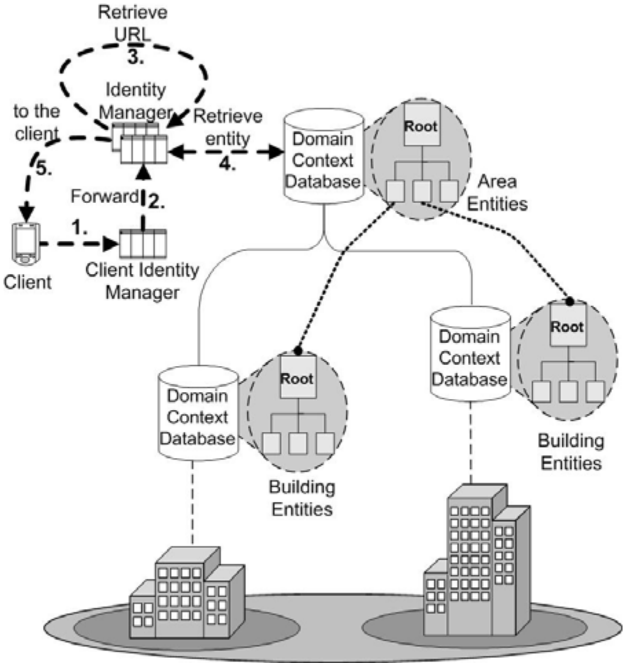


Figure 4. VID/TID-based entity retrieval

To protect context from this kind of threat, URLs are disguised by identifiers. DAIDALOS distinguishes between two kinds of identifiers: pseudonyms –Nyms for short– and temporal identifiers (TIDs). Basically both have the same format, yet in contrast to Nyms that are permanent, TIDs have a short validity period. In addition, Nyms are reserved for representation of persons via the corresponding VID, while a TID may temporarily identify an aircraft a person has boarded. While the person is on the plane, it can access the entity representing the plane by providing this TID.

Consequently, these TIDs are issued and managed by a new functional building block (TID Manager), which provides the handling of the TIDs independently of the handling of VIDs. This is a logical separation, because VIDs are focused on users and do have a longer lifetime than TIDs. In contrast, TIDs are in principle about arbitrary context and their lifetime is shorter. For the sake of simplicity, in this paper it is henceforth assumed that the TID Manager is integrated into the DAIDALOS' Identity Management [8]. Thus, whenever in the remaining the Identity Manager (IM) is referred to as the component processing TIDs or URLs in combination with TIDs, the processing is actually performed by the TID Manager.

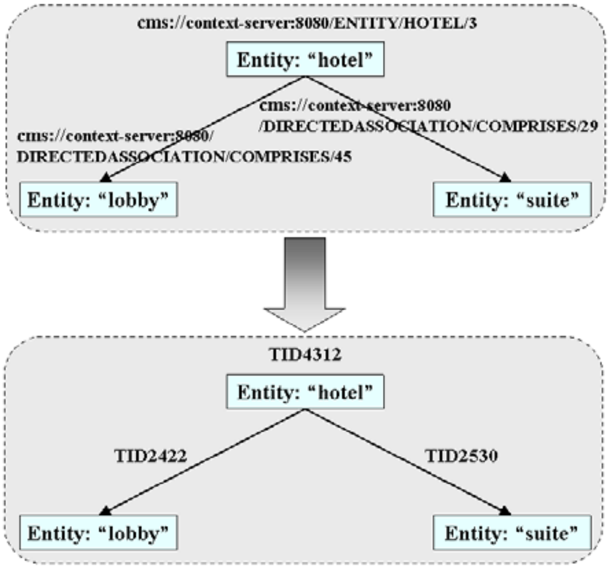

cms://context-server:8080/ENTITY/HOTEL/3
Entity: "hotel"
cms://context-server:8080 /DIRECTEDASSOCIATION/COMPRISES/29
cms://context-server:8080/ DIRECTEDASSOCIATION/COMPRISES/45
Entity: "lobby"
Entity: "suite"
TID4312
Entity: "hotel"
TID2422
TID2530
Entity: "lobby"
Entity: "suite"

Figure 5. Replacement of URLs by TIDs

The retrieval of context entities based on Nyms and TIDs is shown in Figure 4. Because the task of managing Nyms and TIDs is part of Identity Management, context management does not process these. Hence, a client that wants to access an entity must provide the Nym or TID respectively to the Identity Management. Since the IM maintains a database, which maps the Nyms and TIDs to the actual context URLs, retrieving entities from Context Management is straightforward to them. On receiving the request, the IM inspects the identifier, performs the mapping to the URL and forwards the request to the Context Management. Yet, as raw entities still contain context URLs, they must be processed before being returned in a response to a client. To this end, the IM processes URLs and replaces them by TIDs, which are, in case no TID already exists, created dynamically. Finally, the issuing IM passes the processed entity back to the requesting client. Figure 5 illustrates this processing in a simplified form.

The processing of location-based queries, as depicted in Figure 6, is similar: The client passes the queried coordinates to an IM. The latter forwards the request without any processing to the domain Context Database, which triggers the request routing through the context hierarchy. When the context Home Manager, i.e., the one storing the searched entity, receives the query, it fetches the entity from its store and passes it on to the (trusted) IM. The IM replaces the URLs by TIDs as described in the Nym/TID retrieval case. Finally, the processed context entity is forwarded to the client.

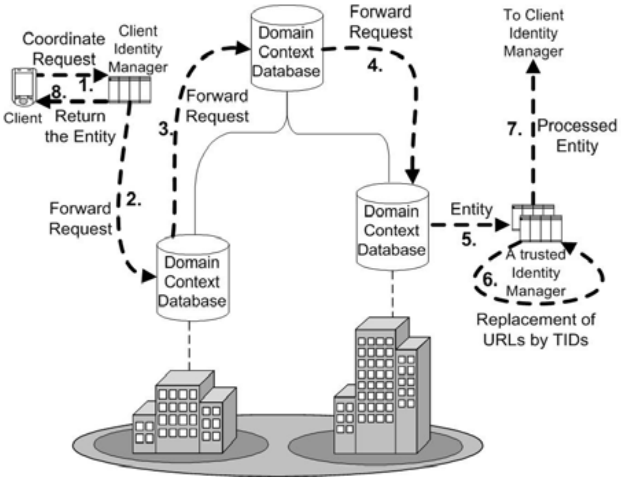


Figure 6. Location-based entity retrieval

Since both Nyms and TIDs are required to encapsulate the address of their issuer, a context manager must have more than one trusted IM. Alternatively, IMs must not serve only a single context database. Otherwise a malicious client would again be able to identify relationships between entities, simply based on the fact that Nyms and TIDs are issued by the same instance. Therefore, the same party should not operate the context databases and IMs.

Typically, TIDs have a short expiration time. Hence, clients cannot record their browse history based on TIDs. After a while all visited TIDs become invalid and cannot be used any longer. However, when a client requires frequent entity updates, the IM may renew the contract or create a new TID. Since the IMs create TIDs for every request, none of the clients share TIDs. Thus, a malicious client cannot detect relationships between entities by comparing the issued TIDs. Even when an attacker intercepts the delivery of entities, the TID comparison attack fails. Lacking the knowledge of the semantics of browsing history, clients cannot determine relationships between entities as they are disguised by TIDs.

6. Overview of privacy protecting approaches in context-aware systems

Protecting personal privacy is a critical requirement for the successful deployment of pervasive computing systems in the real world. Numerous research initiatives have designed and established various privacy protecting schemes that exploit traditional ones (e.g. policies, role-based access control, etc.) adequately enhanced in order to address the needs of a pervasive computing environment. Although the results produced in this area have been quite promising, the vast majority of current initiatives deals with the security issue as a separate scientific discipline and do not consider the impact on context management systems. Thus, even in cases where the privacy schemes exhibit prominent advantages, they seem to fall short in important aspects like securely caching and distributing context data, or employing a context representation and hierarchy scheme adequate for both management and security purposes. Furthermore, another critical issue that is not appropriately addressed in most privacy protecting approaches is the user's anonymity. Anonymity is about hiding your real identity during your transactions. Although it is not a new term, it becomes of critical importance in a pervasive computing environment. In the remainder of this section a brief review of the most important research work on privacy protecting frameworks is presented, while an attempt is made to evaluate each presented initiative.

The Context Toolkit project [21], developed by Georgia Institute of Technology, is one of the first architectures that attempted to address privacy concerns. The Context Toolkit is based on abstract components named context widgets, interpreters and aggregators. These components are responsible for gathering, processing and propagating context data to appropriate applications. A special kind of widget is a server that collects, stores, and interprets information from other widgets. A server acts like a gateway between applications and widgets therefore can be used to encapsulate a privacy manager. A user could implement access restrictions on sensitive information stored on his personal server. On the other hand it is reported that this privacy mechanism is not robust and that, sensitive contextual information could be accessible without any restriction [22].

CoBrA (Context Broker Architecture) [22] is a broker-centric agent based architecture. The Context Broker process context data collected from agents, devices, and sensors in order to provide context-aware functionality. Among others, Broker functions include security mechanisms for addressing user privacy protection issues. In this respect, the Broker is responsible for enforcing user defined policies in order to protect the privacy of users when context information is shared with other agents in the community. For achieving this purpose, CoBra imports and extends the SOUPA policy ontology (http://pervasive.semanticweb.org) so that users can define customized policy rules to permit or forbid access to their private information. Additionally, CoBra implements a metapolicy-reasoning mechanism suggested by the SOUPA ontology in order to complement the user-defined policy with a global policy and let users adjust their information's granularity. The results produced from the alliance of SOUPA and CoBra aim to standardize an ontological framework for defining privacy policies. Nevertheless, the whole scheme is totally dependent on a stationary environment and is used in a rather restricted manner, while no proof is provided about its potential for handling distributed context information.

Owl [23][24] is a middleware infrastructure for context collection and dissemination, realized as a Context Service. Among Owl's primary design goals is the protection of user's privacy through the implementation of a role-based access control (RBAC) [25] mechanism. In this respect, the Privacy Engine component is realized as part of the Context Service framework, and its main purpose is to specify, store, and retrieve the established privacy policies. The design principle of the privacy management system is that users preserve control of their context information, but taking as granted that users trust the system in which the identity of all participants is known. The Privacy Engine offers the ability to specify policies for groups of users as well as individuals. This feature in combination with RBAC reduces the administration overhead. Owl's privacy scheme displays interesting features, but does not take into consideration the need for distributing and securing context information across multiple administration domains. On the contrary, our approach deals explicitly with this perspective.

UbiCOSM Middleware (Ubiquitous Context-based Security Middleware) [26] has developed an access control model built upon the concept of context as the first-class design principle to rule access to resources. Unlike traditional access control models, where permissions are associated with user identities or roles, UbiCOSM model access permissions are determined with regards to the user's context. In a nutshell, this model supports associating access control permissions with context where users operate and users acquire/lose their permissions, when entering/leaving a specific context. The access permissions are expressed in a RDF-based [27] standard format. Although access control to resources is crucial, its potential is rather restricted for addressing the need for securing all kind of context data capturing, refinement and dissemination during pervasive services' provision. A similar approach to the one of UbiCOSM model is adopted by the context based secure resource access architecture [28] and the access control for Active Spaces [29].

Privacy Awareness System (pawS) [30] facilitates an interesting approach on privacy protection mechanisms. It provides pervasive computing environment users with a privacy-enabler and not a privacy protector. By implementing a platform for the P3P [31] based privacy model, PawS offers tools that allow data collectors and processors to communicate their privacy policy. Upon agreement on the privacy policy presented by the service and the desired privacy policy of the user, the service is granted access to user's sensitive data. Moreover a privacy aware database (pawDB) combines and stores the collected data and their privacy policies into a single file, thus assisting the system to better handle them according to their usage policy. PawS system is a general-purpose framework for protecting users' privacy in a pervasive environment that is based on privacy proxies distributed among various nodes. Although, it offers compelling benefits, it still remains unclear how linkage between user's context data is prevented (i.e. how user's anonymity is achieved), as well as how the established privacy policy mechanisms could integrate with a specific context model.

Emapp (Encapsulated Mobile Agent-based Privacy Protection) [32] is a privacy protection technology similar to pawS, with additional security procedures. The model tries to address pawS's potential risk of sensitive data misuse after the data has flow out from its original location. It introduces the concept of a privacy capsule, where personal data is encapsulated together with the associated user's preferences stored in user's personal device. Data inside the capsule were not accessible by the outside world without the user's preferences check. Mobile agents migrate into the privacy capsule and after the execution of appropriate processes, decisions are made concerning the data accessibility by external actors. In contrast to pawS, Emapp deals with context data sets storage (with the use of privacy capsules). However, this approach implies that all context consumers must send their agents in a specific node (i.e. where the capsule resides) and wait until their request is processed.

The Mist [33] privacy protocol has the primary goal to preserve the privacy of users while they are communicating in a ubiquitous computing environment. The Mist protocol protects the users' physical location information from all other parties and preserves their anonymity, even from the system it self. This is achieved by implementing a routing protocol combined with strong public key cryptography that allows the system to detect the presence of users in a place, but not to positively identify them. The Mist system claims to have achieved location privacy, anonymous connections and confidentiality during communications, but it is not clear how disclosure of other personal data, apart from location information, is controlled and performed to interested parties.

Mix zone [34][35] is a middleware that provides 3rd party applications with anonymized user location information. The model introduces the concept of application zones and mix zones. Application zones are geographical spaces, in which (untrusted) 3rd parties can provide their context-aware services. Mix zones are certain areas, where no location information is available for the application providers. Each time a user enters an application zone, he/she is assigned with a new pseudonym. A 3rd party application provider receives this pseudonym, and not a traceable user identity associated with user's location. Once a user enters a mix zone, his/her identity is mixed with all other users in the mix zone. The communication between the user and the context-aware services is accomplished only through the middleware, in order to prevent linking of pseudonym and user identity. Mix zone system has provided significant results in supporting user anonymity in location-aware services, but focuses solely on the location information privacy in pervasive computing applications.

7. Conclusions

The abundance of commercial sensing technologies and the prevalence of powerful networked devices are bringing the pervasiveness vision closer. Nevertheless, before this vision can be realized, context-awareness coupled with protection of user privacy has to be established. Context-awareness implies a growing amount of privacy sensitive (context) information of users to be tracked, stored and distributed in the system. Therefore, while designing our context management system for the DAIDALOS services platform, privacy protection means are considered from the beginning. This ranges from adhering to the approach of multiple virtual identities, i.e., pseudonymous usage of services while maintaining accountability, chargability and non-repudiation, integration of user management as well as considerations of compatibility with classical authentication and authorization schemes.

The research presented in this paper focuses on the design of a context model for distributed pervasive computing environments that addresses major privacy concerns. We have recently finalized the implementation of the pervasive service platform prototype, which supports the provision of secure context-aware services. This prototype has been built on an OSGi Service Platform [36]. The OSGi™ specifications define a standardized, component-oriented computing environment for networked services. For remote communication SOAP [37] is used, while the discovery of services and components is based on the SLP [38]. The final demonstration of the developed pervasive service platform took place in November 2005. It is currently being evaluated against performance, user-friendliness, efficiency, scalability, usefulness and commercial criteria, over a blend of heterogeneous technologies encompassing multi-role domains. Its validation is expected to further contribute to the integration of pervasive systems, while this work will hopefully make a step towards the introduction of pervasive service provision in the wide market.

8. References

[1] M. Satyanarayanan, "Pervasive computing: vision and challenges", *IEEE Personal Communications Magazine*, Vol. 8, No. 4, Aug. 2001, pp. 10-17.

[2] M. Weiser, "The computer for the 21st century", *Scientific American*, Vol. 265, No. 3, Sep. 1991, pp. 94-104.

[3] J. Sun, "Mobile ad hoc networking: an essential technology for pervasive computing", *Int. Conf. on Info-tech & Info-net*, Beijing, China, Oct. 2001.

[4] S. Xynogalas, M. Chantzara, I. Sygkouna, S. Vrontis, I. Roussaki, and M. Anagnostou, "Context Management for the Provision of Adaptive Services to Roaming Users", *IEEE Wireless Communications*, Vol. 11, No. 2, Apr. 2004, pp. 40-47.

[5] A. Dey, "Providing Architectural Support for Building Context-Aware Applications", *Ph.D. thesis*, College of Computing, Georgia Institute of Technology, Atlanta, USA, Feb. 2000.

[6] IST Daidalos Research, URL: http://www.ist-daidalos.org.

[7] B. Farshchian, J. Zoric, L. Mehrmann, A. Cawsey, H. Williams, P. Robertson, and C. Hauser, "Developing Pervasive Services for Future Telecommunication Networks", *IADIS Int. Conf. WWW/Internet* (ICWI 2004), Madrid, Spain, Oct. 2004.

[8] J. Clarke, S. Butler, C. Hauser, M. Neubauer, P. Robertson, I. Orazem, A. Blazic, H. Williams, and Y. Jang,

"Security and Privacy in a Pervasive World", *Eurescom Summit 2005*, Heidelberg, Germany, Apr. 2005.

[9] M.H. Williams, I. Roussaki, M. Strimpakou, Y. Yang, L. MacKinnon, R. Dewar, N. Milyaev, C. Pils, and M. Anagnostou, "Context-Awareness and Personalisation in the Daidalos Pervasive Environment", *IEEE Int. Conf. on Pervasive Services* (ICPS 2005), Santorini, Greece, Jul. 2005.

[10] S. Xynogalas, I. Roussaki, M. Chantzara, M. Anagnostou, "Context Management in Virtual Home Environment Systems", *Journal of Circuits, Systems, and Computers*, Special Issue on Mobile and Wireless Networking, Vol. 13, No. 2, Apr. 2004, pp. 293-311.

[11] M. Strimpakou, I. Roussaki, C. Pils, P. Robertson, M. Angermann, and M. Anagnostou, "Context Modelling and Management in Ambient-aware Pervasive Environments", *Int. Workshop on Location and Context-Awareness* (LoCA 2005), Oberpfaffenhofen, Germany, May 2005.

[12] T. Strang, C. Linnhoff-Popien, and K. Frank, "CoOL: A Context Ontology Language to enable Contextual Interoperability", *4th IFIP WG 6.1 Int. Conf. on Distributed Applications and Interoperable Systems* (DAIS 2003), Paris, France, Nov. 2003.

[13] C. Becker, F. Dürr, "On location models for ubiquitous computing", *Personal and Ubiquitous Computing*, Vol. 9, No. 1, Jan. 2005, pp.20-31.

[14] P. Albitz, C. Liu, "DNS and BIND", Publisher: O'Reilly Media, Inc., 4th Edition, Apr. 2001.

[15] M. Mouly, and M.B. Pautet, "The GSM System for Mobile Communications", *Telecom Publishing*, Jun. 1992.

[16] A. Pfitzmann, M. Hansen, "Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management – A Consolidated Proposal for Terminology", Version 0.26, 13th Dec. 2005, URL: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

[17] J. Porekar, K. Dolinar, "Identity Management and Privacy issues in DAIDALOS pervasive environment", 15th Wireless World Research Forum (WWRF) Meeting, Paris, France, Dec. 2005.

[18] B. Weyl, P. Brandao, A.f. Gomez Skarmeta, R.M. Lopez, P. Mishra, C. Hauser, H. Ziemek: "Protecting Privacy of Identities in Federated Operator Environments", *14th IST Mobile & Wireless Communications Summit 2005*, Dresden, Germany, Jun. 2005.

[19] Directive 95/46/EC of the European Parliament and of the Council of 25 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (Official Journal L 281, 23/11/95, pp. 31-50).

[20] 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 (Official Journal L 201, 31/07/2002, pp. 37-47).

[21] D. Salber, A.K. Dey and G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications", *Conference on Human Factors in Computing Systems 1999* (CHI 1999), May 1999.

[22] H. Chen, T. Finin, A. Joshi, and L. Kagal, "Intelligent Agents Meet the Semantic Web in Smart Spaces", *IEEE Internet Computing*, Vol. 8, No. 6, Nov.–Dec. 2004, pp. 69-79.

[23] H. Lei, D.M. Sow, J.S. Davis II, G. Banavar, M.R. Ebling, "The design and applications of a context service", *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 6, No. 4, Oct. 2002, pp. 45–55.

[24] M. Ebling, G.D.H. Hunt, and H. Lei, "Issues for Context Services for Pervasive Computing", *Workshop on Middleware for Mobile Computing* (Middleware 2001), Heidelberg, Germany, Nov. 2001.

[25] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-based Access Control Models", *IEEE Computer*, Vol. 29, No. 2, Feb. 1996, pp. 38-47.

[26] A. Corradi, R. Montanari, and D. Tibaldi, "Context-based access control management in ubiquitous environments", *3rd IEEE Inte. Symposium on Network Computing and Applications* (NCA 2004), Aug. 2004.

[27] Resource Description Framework (RDF), URL: http://www.w3.org/RDF/

[28] A. Tripathi, T. Ahmed, D. Kulkarni, R. Kumar, and K. Kashiramka, "Context-based secure resource access in pervasive computing environments", *2nd IEEE Annual Conf. on Pervasive Computing and Communications Workshops* (PERCOMW 2004), Florida, USA, Mar. 2004.

[29] G. Sampemane, P. Naldurg, and R.H. Campbell, "Access control for active spaces", *18th Annual Computer Security Applications Conference* (ACSAC 2002), Las Vegas, USA, Dec. 2002.

[30] M. Langheinrich, "A privacy awareness system for ubiquitous computing environments", *Int. Conf. on Ubiquitous Computing* (UbiComp 2002), Goteborg, Sweden, Sep. 2002.

[31] P3P Project, URL: http://www.w3.org/P3P/

[32] N. Huda, S. Yamada, E. Kamioka, "Privacy Protection in Mobile Agent Based Service Domain", *3rd Int.l Conf. on Information Technology and Applications* (ICITA'05), Sydney, Australia, Jul. 2005.

[33] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, "Routing through the mist: Privacy preserving communication in ubiquitous computing environments", *Int. Conf. of Distributed Computing Systems* (ICDCS 2002), Vienna, Austria, Jul. 2002.

[34] A. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services", *2nd IEEE Annual Conf.e on Pervasive Computing and Communications Workshops*, Florida, USA, Mar. 2004.

[35] A.R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing", *IEEE Pervasive Computing*, Vol. 2, No. 1, Jan.-Mar. 2003, pp. 46–55.

[36] OSGi Alliance, URL: http://www.osgi.org/.

[37] Simple Object Access Protocol (SOAP), URL: http://www.w3.org/TR/soap/.

[38] E. Guttman, "Service Location Protocol: Automatic Discovery of IP Network Services", *IEEE Internet Computing*, Vol.3, No. 4, Jul. 1999, pp. 71-80.