

Context Management Framework for MAGNET Beyond

M. Bauer, R.L.Olsen, M. Jacobsson L. Sanchez, J. Lanza, M. Imine, N. Prasad

Abstract—In this paper we discuss architectural issues that are especially relevant in a context management system for Personal Networks. We identify the needs and requirements for a solution, based on a set of use cases, which can manage context information within one Personal Network (PN) or a federation of PNs. We describe the architectural challenges that have to be tackled to enable context aware services and applications in PNs and propose components for an efficient context management framework. In this paper we put special focus on the use of context information from and within the communication stack. We also address security and privacy issues that arise, if context information is to be shared between users in different PNs

Index Terms—Context management, Personal Networks, use cases, high level requirements, security

I. INTRODUCTION

Personal Networks are a new network paradigm introduced in [1]. This type of network encompasses a user's personal area network (in our notation, the Private-PAN or P-PAN), but also his/her personal devices in remote locations like his/her office or his/her home. Figure 1 shows the conceptual structure of a Personal Network with the user's Private PAN as its core. An interconnecting structure connects the P-PAN with the user's personal devices and nodes at his/her home cluster, office cluster or whatever is currently interesting for the user to interact with. What is specific about this concept is that it provides the user with an easy and secure way of interacting with nodes, services and applications which have a personal relation to the user. This concept has been under development and research in the MAGNET Project, and is extended in the MAGNET Beyond project to also incorporate cooperative/federated networks of this type, e.g. if a user wishes to share services, resources or other information in a work relation or with his/her family.

The Personal Network accompanies and supports the user wherever he/she goes. As the network is private, there is great potential in personalizing not just single applications, services or devices, but the whole computing environment to the user's needs. As the focus of the mobile user may be on real world tasks, proactive computer support becomes important. This proactive behaviour pertains to automatically adapting the

Personal Network and the computing environment as a whole to the dynamically changing environment, but also to alerting the user when something of interest happens. These types of actions require information regarding the current situation of the user, which is typically referred to as *context* information. Application and services using context to adapt their behaviour are called *context-aware*.

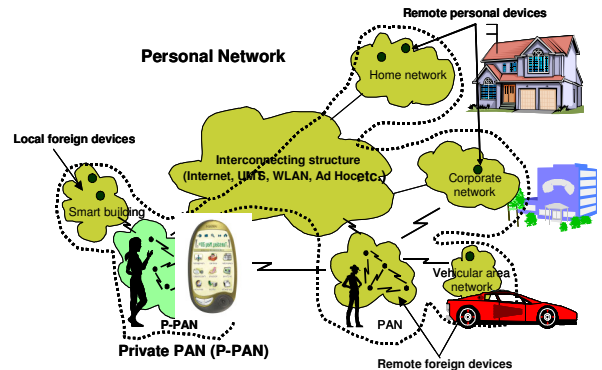


Figure 1 A conceptual illustration of a Personal Network

The main challenges for management of context information in Personal Networks are the potential size, dynamicity and heterogeneity of the network. The dynamicity pertains to the changing availability of nodes, e.g., due to the movement of the user and the limited battery power, but also the changing availability of networks that affect connectivity. These changes in the network also affect the availability of context sources and the nodes that can process and manage context information. Therefore, a highly flexible and adaptive context management framework is necessary which provides efficient access to all relevant context information that is available at any time.

In this paper we present a high level description of a context management framework that will allow services, applications and other networking components in PNs to be context aware. First we provide a small set of use cases of how context will be used in MAGNET Beyond. Second we provide a set of requirements to the context management framework, and third we give a high level description of a secure context management framework, and finally we conclude with a perspective of our coming work.

II. RELATED WORK

Early context aware applications were often designed as standalone applications, each managing its own context information, e.g. [2]. This approach limits the reuse of components. Sharing context between applications is often practically impossible.

Middleware like the Context Toolkit [3] introduced generic components that could be reused. Still, the composition of components is closely tied to the application and the particular

The work presented in this paper has been carried out under the auspices of the IST-027396 **MAGNET Beyond** project

Martin Bauer is with NEC Europe Ltd., Network Laboratories, Kurfürsten-Anlage 36, 69115 Heidelberg, Germany (e-mail: martin.bauer@netlab.nec.de)

Rasmus L. Olsen, Mohammed Imine and Neeli Prasad is with the Center for TeleInfrastruktur, Aalborg University, Niels Jernes vej 12, 9220 Aalborg, Denmark (e-mail: {[rlolmilnp](mailto:rlolmilnp@kom.aau.dk)}@kom.aau.dk).

Luis Sanchez and Jorge Lanza are research engineers at the Network Planning and Mobile Communications Lab, University of Cantabria, Dept. Ingeniería de Comunicaciones, Avda de los Castros s/n, 39005, Santander, Spain (e-mail: {lsanchez, jlanza}@tmat.unican.es).

Martin Jacobsson is a PhD candidate at the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) of Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands (email: m.jacobsson@ewi.tudelft.nl).

environment, making it difficult to build applications that can automatically adapt to different environments using the context sources available there.

To alleviate this problem, context management frameworks were introduced that decoupled the application from the context sources, making it possible to share context between applications and allowing applications to seamlessly work in different environments, e.g., [4],[5]. These frameworks are typically based on a server infrastructure, so permanent access to this server infrastructure is required.

In personal networks the access to a fixed infrastructure cannot be guaranteed due to changing connectivity and availability of resources. Also, limitations in bandwidth and battery power require reducing the communication overhead as much as possible, so context should only be communicated if required. Therefore, we need a context management framework that takes the special requirements of a Personal Network into account.

III. THE USE OF CONTEXT IN PERSONAL NETWORKS

The themes that are under investigation in the MAGNET Beyond project are called *nomadic@work* and *magnet.care*. In this section, we provide a small subset of use cases to outline what we expect of the system, more thoroughly described use cases can be found in [6].

A. *Nomadic@work*

1) *Context Aware Service Discovery*

A journalist is carrying a portable device, which incorporates his/hers contextual information (interests, age, personal characteristics, current position and preferences etc.) The journalist is required to make a trip in order to cover a specific story, and wants to find out what services are available at the place he/she is going to.

1. The journalist initiates by starting a GUI and inputs a search request along parameters such as travel destination, time and estimated budget.
2. The system notices his/hers preferences and current position to initiate a service discovery request.
3. The system returns with a list of available services with relevance to the user.

2) *Context aware service provisioning*

A journalists carrying one or more MAGNET enabled personal devices (PDA, notebook) with WiFi communication capability is streaming video through her notebook at the office premises and needs to go.

1. The user leaves the office and goes into her car.
2. The system notices that the battery of the notebook is running low, and notifies her.
3. The journalist initiates a stream switch, so that the video is now streamed to a PDA.
4. The system renegotiates QoS according to the new situation, i.e. the PDA's capability, bandwidth etc.
5. The stream at the notebook is terminated as the stream starts at the PDA.

B. *Magnet.care*

1) *Emergency call for a diabetic*

A diabetic is having problems with his/her level of blood sugar and needs to input a certain dose of insulin into his/her body.

1. A Blood Glucose Meter has been set to frequently report the user's blood glucose level to a specific diabetes application
2. At some point in time, the level drops to either a very low or high value, and the system sends a warning to an application.
3. The application needs to find an appropriate screen to popup a notification window and asks the system to find the appropriate screen service.
4. The application acquires the level of blood glucose, and is now able to send recommended settings for injection dose to a special injector pen.

2) *Remote monitoring of the diabetes*

The diabetic may want to maintain control over his/her history and even having his/her personal doctor to be able to look into the history of the user's blood glucose level.

1. The user instructs the system to keep updates and history line of the blood glucose level available remotely and on which devices
2. The system does this at the predefined time interval. Optionally, prior to update, ask for permission from the user.

C. *Short summary of used context*

Based on these use cases and others from [6] we list below a set of information that is considered useful to the project:

- **User context:** User's geographical position, time and date, age of user and various user preferences and settings, identity of the user, user's blood glucose level
- **Network context:** Various information about the network condition e.g. link and end-to-end delay, bandwidth etc., available networks.
- **Environmental context:** Local weather information

D. *Usage of context information history*

Some of the context information changes over time and for many applications, it is interesting to see how this information changes. Certain patterns can be learned in order to optimize or predict future behaviour. The context aware service provisioning system can for instance remember that certain services are available at certain circumstances and proactively hand over to another server when the observable context changes. It is therefore also interesting to store some part of the context information for future use or for machine learning.

IV. REQUIREMENTS TO CONTEXT MANAGEMENT FRAMEWORK

In this section we provide an overview of the most important requirements that have to be met in order for the context management framework to operate properly in the settings of (federated) Personal Networks. We assume a client of the context information to be a person (owner of the PN), an application, a service or some other networking component.

Requirement 1. Adding, maintaining and removing context information

A client should be able to:

1. Request the context managing entity to provide a particular context information
2. Maintain the value of particular context information.
3. When not needed anymore, it should be possible to remove the information again.

The framework must also be able to keep updated values of the information by whatever means available or appropriate. Old information may also be stored for future use. This requirement is based on use case B-1.1.

Requirement 2. Efficient access to context information

It is the task of the system to provide efficient access to context information. The client does not have to know the potential sources of the context information. It should, however, also be possible for the client to specifically inform the context management framework where the information comes from and what communication means is to be used for maintenance. This requirement is common to all use cases.

Requirement 3. Adaptation to changing context sources

The system has to adapt to changes in the available context sources and always provide the client with the context information that fits his/her request. This requirement is based mainly on use case A-2.2.

Requirement 4. Scalability

The context management framework must not introduce exceptionally long response delays or show lack of functionality considering the increase of

- The number of clusters and nodes in the PN or in the federated PNs.
- The amount of information that is/becomes available in the system.

This requirement is at most relevant for Scenario B-2.2.

Requirement 5. Ability to handle missing and ambiguous context information

In some cases, context information may not be present for various reasons, or it may be overrepresented which may lead to ambiguities. If there are many users to the same information, it is in most cases, desired for them to share the same view of context. The context management framework must therefore have a clear and well-defined way to handle such situations. This is not directly required by any of the illustrated use cases, but is required in the sense that the system outcome otherwise might be inappropriate to the user.

Requirement 6. Support of context triggered events

Some applications may depend on knowledge whether a certain situation or event has occurred or not. The context management framework must be able to support a mechanism signalling required to do this. This is required by use case B-1.2, and could be a potential automation in A-2.3.

Requirement 7. Standardised data formats

Since context can be much information and as such, can be described in many fashions, the supported context information must be described using one or more standardised data format.

This is not a requirement directly from the use cases, but rather a requirement based on the diversity of the information and context sources, that is to be used within the use cases.

The following requirements are defined to protect the privacy of the user when allowing access to/from context sources within and outside the PN. This also covers federated PN's.

Requirement 8. Privacy of the user

The distribution and sharing of context information should be governed by rules and policies which ensure that the user's privacy is not endangered.

Requirement 9. Authentication

If context is accessed to/from outside the PN, the system must ensure that the context provider/client is identified and authenticated prior to the access.

Requirement 10. Data integrity and confidentiality

If context is accessed to/from outside the PN, the system must ensure that data integrity and user confidentiality is kept.

Requirement 11. Data freshness and non repudiation

If context is accessed to/from outside the PN, the system must be able to detect the *freshness* of data and ensure that data access cannot be repudiated.

V.CONTEXT MANAGEMENT FRAMEWORK

Using context information can be described by four steps:

- 1) Retrieve/capture context information
- 2) Process and store context information
- 3) Distribute the information
- 4) Use the information

In the following sections we introduce the high level architecture that allows the system to perform step 2) and 3).

A. Conceptual architecture

provides a conceptual view of the high-level functional components that the context management framework has to provide.

There will be a large variety of sources that provide very different input using different formats and access protocols, e.g. context from sensors, user preferences or information from the communication stack. In order to allow a unified handling of the context information, we have introduced a sensor abstraction layer. Through the sensor abstraction layer, the context management framework gets uniform access to context information in its standard context representation format.

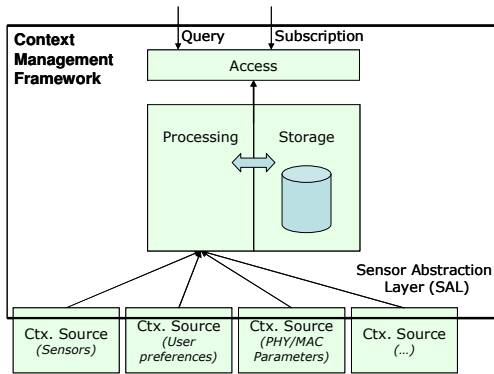


Figure 2 Conceptual view of Context Management Framework

The access component of the context management framework provides two interaction modes:

- **Query/Response:** A synchronous interaction node that allows an application to query for context information.
- **Subscribe/Notify:** An asynchronous interaction node that allows an application to subscribe for context information or a situation. The subscription can provide updates based on
 - a periodic time interval.
 - an absolute threshold violation
 - a relative threshold violation

The processing component allows the placement of context operators for deriving higher levels of context information. Context operators can provide functionalities for filtering, aggregating and combining context information into context situations, as well as reasoning and learning capabilities.

The context information provided by the context sources as well as derived context can be stored by the storage component in order to provide efficient access to context information.

B. Context management framework within the PN

The structure of the PN, shown in Figure 1, also forms the basis for the context management framework.

Each node that is part of the context management framework has to provide as a minimum the access component to allow uniform access to context information. Nodes with context sources will also provide the sensor abstraction layer, integrating the provided context information into the context management framework. Computationally stronger nodes may also provide processing and storage functionalities.

In order to allow efficient access to context information within a cluster, there is a designated context management node (CMN) that has information about which context information is available on which node, i.e. it stores index information, not necessarily the potentially much more dynamic context information itself. This requires that new context nodes register the context information they can provide with the context management node. To initialize the context management node, or locating unknown context sources, a discovery protocol may necessary. The existing service discovery framework developed in the MAGNET

project, offers the required discovery functionality, which makes it a strong candidate for this.

To allow context access on the PN level, the context management nodes in the clusters have to interact in a peer-to-peer fashion. We will investigate different options ranging from no replication of index information, so all context management nodes have to be queried, to partially and fully replicated index information, so that either the cluster or even the individual node providing the context information is known by every context management node.

C. Context in the OSI communication layers

As seen in the use cases, network context is an important context class for applications in personal networks, especially because of the dynamic changes in the available networks due to the mobility of the user. This context information may be provided by different layers in the OSI stack.

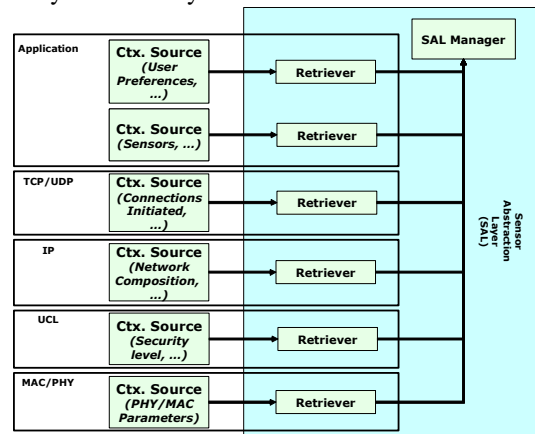


Figure 3 Context sources in different networking layers

Error! Reference source not found. shows local context sources providing information coming from different network layers. In the following we will look at context that can be provided in the network and universal convergence layer, and also how it can be used for communication purposes.

1) Network layer

The knowledge of any statistics or information of the link that makes it possible the communication of a node with its surroundings is shown to be very important when providing a service to the user. In this sense, it is clear that the context information must flow up and down in the network stack in order to supply services to the user in the best conditions.

The purpose of the network layer is to provide good quality communication between nodes and context information can be used to enhance this. The more information that is available about alternative paths, the better the selection will be. The information related to the selected and/or alternative paths can further be communicated to higher layers in order for other components, services or applications to better adapt to the current situation.

The network layer needs to gather information about neighbouring nodes, their communication capabilities and whether they are routers connected with the Internet, etc. In addition, user profile information is used in order for the user to be able to influence the decisions.

2) Universal convergence layer

Some of the possible metrics that can be retrieved at link layer and bound to context decision are: *Characteristics* of the underlying wireless network technologies (number of them, bandwidth, retry limit, ...), *SNR* and other channel metrics as *number of lost packets*, *Node population* of the surroundings and *detection of new neighbours* and *nearby foreign PN*, *Security associations* with neighbour peers

The Universal Convergence Layer (UCL) [7] is located just above link layer of the underlying network interfaces and hides the complexity of the different link layers to the network layer. Its privileged position in the protocol stack makes it the perfect candidate to gather all context information from the controlled wireless interfaces.

D. Secure distribution and sharing of context information

The problem of user data privacy and the user anonymity are of paramount importance in the PN framework. The context manager has to be able to ensure that security and privacy requirements, such as privacy and authentication (Requirement 8-11), are met considering both updates of changing contexts but also the sharing of sensitive information with PN federation members. Static settings and/or updates are done when the user performs them through his/her secured GUI. Indirect dynamic updates to/from foreign nodes are only allowed when the user trusts and has authorized the nodes and running applications/service involved in the process. Security is divided in three different levels based on the context:

- *Security Level (Low, Medium, High)*: Security levels that only the user is allowed to setup for communication with other personal/foreign nodes
- *Preference Level (Yes or No)*: Preferred levels that the user is allowed to setup for accessing services
- *Access Control Level (Low, Medium, High)*: Allowed levels that the user/foreign 's services are allowed to be accessed with
- *Trust Level (Unknown, Untrusted, Trusted)*: Trust levels that the user is allowed to setup other nodes/services

Additional four privacy attributes *Always*, *Check*, *Ask*, and *Never* are proposed to be incorporated with the user's sensitive data directly or through a data abstraction model. A new algorithm (called Always-Check-Never-Ask) is proposed to implement the safeguard mechanism for handling the user's sensitive data. Each time a request is made to the context manager to provide any of the user's sensitive data that is part of the user profile, the context manager makes a request to the privacy safeguard algorithm to filter the sensitive data based on the user's privacy flags setting before it is provided to the requesting party.

VI. CONCLUSION AND OUTLOOK

In this paper we have presented a number of example use cases and some resulting requirements for the architecture of a context management framework for Personal Networks. We have then proposed a high-level architecture with a focus on

the use of context from and within the communication stack. Finally we have addressed some of the security and privacy issues related to the management of context information in Personal Networks.

We will now refine and implement the presented approach so that it can be validated within the MAGNET Beyond platform, based on the pilot services that will be implemented on the basis of the use cases. Special emphasis will be on extending the architecture towards PN federations.

ABOUT MAGNET BEYOND:

MAGNET Beyond is a continuation of the MAGNET project (www.ist-magnet.org). MAGNET Beyond is a worldwide R&D project within Mobile and Wireless Systems and Platforms Beyond 3G. MAGNET Beyond will introduce new technologies, systems, and applications that are at the same time user-centric and secure. MAGNET Beyond will develop user-centric business model concepts for secure Personal Networks in multi-network, multi-device, and multi-user environments. MAGNET Beyond has 32 partners from 15 countries, among these highly influential Industrial Partners, Universities, Research Centres, and SMEs.

REFERENCES

- [1] I.G.G. Niemegeers and S.M. Heemstra de Groot, From Personal Area Networks to Personal Networks: A User Oriented Approach, Kluwer Journal, Personal Wireless Communication, May 2002
- [2] S. Long, R. Kooper, G.D. Abowd, C.G. Atkeson, "Rapid prototyping of mobile context aware applications: the Cyberguide case study." Proceedings of the Second Annual International Conference on Mobile Computing and Networking, White Plains, NY, ACM Press, 1996, pp 97-107.
- [3] D. Salber, A. Dey, G. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications", Proceedings of the Conference in Human Factory in Computing Systems (CHI), 1999, pp. 434-441.
- [4] P. Floréen, M. Przybyski, P. Nurmi, J. Koolwaaij, A. Tarlano, M. Wagner, M. Luther, F. Bataille, M. Boussard, B. Mrohs, S. Lau, "Towards a Context Management Framework for MobiLife", IST Mobile & Communications Summit, 2005, Dresden, Germany.
- [5] M. Grossmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas, T. Schwarz, "Efficiently Managing Context Information for Large-Scale Scenarios", Third IEEE International Conference on Pervasive Computing and Communications (PerCom) 2005., pp. 331 – 340
- [6] H. Olesen, N. Schultz, K. E. Skouby, L. Sørensen, S. Bessler, D. Kyriazanos, and C. Z. Patrikakis, "Scenario Construction and Personalization of PN Services based on User Profiles and Context Information", to be published in the proceedings of the
- [7] Luis Sanchez, Jorge Lanza, Luis Muñoz, Julian Perez, "Enabling Secure Communications over Heterogeneous Air Interfaces: Building Private Personal Area Networks", 8th International Symposium on WPMC - Aalborg, September 2005, pp. 1963-1967.
- [8] Information security glossary and reference: http://www.yourwindow.to/information_security/gl_glossaryandreference.htm
- [9] Information security reference: <http://www.thefreedictionary.com/>