Adv. Radio Sci., 9, 231–235, 2011
www.adv-radio-sci.net/9/231/2011/
doi:10.5194/ars-9-231-2011

Advances in
Radio Science

An efficient architecture for the integration of sensor and actuator networks into the future internet

J. Schneider, A. Klein, C. Mannweiler, and H. D. Schotten

University of Kaiserslautern, Institute for Wireless Communication and Navigation, Paul-Ehrlich-Straße – Building 11,
67663 Kaiserslautern, Germany

Abstract. In the future, sensors will enable a large variety of new services in different domains. Important application areas are service adaptations in fixed and mobile environments, ambient assisted living, home automation, traffic management, as well as management of smart grids. All these applications will share a common property, the usage of networked sensors and actuators. To ensure an efficient deployment of such sensor-actuator networks, concepts and frameworks for managing and distributing sensor data as well as for triggering actuators need to be developed. In this paper, we present an architecture for integrating sensors and actuators into the future Internet. In our concept, all sensors and actuators are connected via gateways to the Internet, that will be used as comprehensive transport medium. Additionally, an entity is needed for registering all sensors and actuators, and managing sensor data requests. We decided to use a hierarchical structure, comparable to the Domain Name Service. This approach realizes a cost-efficient architecture disposing of "plug and play" capabilities and accounting for privacy issues.

1 Introduction

Today, most sensors are restricted to a local platform and cannot be accessed from external systems or applications. This drawback causes high redundancy in terms of hardware and hence high capital expenditures. In this paper, we present an architecture for integrating different sensors and actuators into the Internet. Our concept enables efficient management and access to sensors, actuators, and additional meta information via open and standardized interfaces, and provides means for ID and access management, while using the Internet as comprehensive transport medium. Important requirements and objectives, that need to be addressed, are:

- Privacy and Security
- Manufacturer Independence
- Dynamic System Adaptations
- Scalability
- Plug and Play Capability
- Open and Standardized Interfaces
- Low implementation Costs

All these issues are related to the question: "How will such a concept be accepted by users?". In order to gain the users' confidence, the architecture must take care of all these aspects. The market acceptance is not only dependent on users' confidence, moreover the system needs to be composed of mass market products to achieve low-priced products and increase the system's attractiveness. Since our selected transport medium is the Internet, the users benefit from low installation costs and high reusability. Our proposed architecture is not limited to any special application. Moreover, we introduce an architecture which is able to deal with several application fields like home automation, ambient assisted living, home security, smart grid, etc. The paper is organized as follows: Sect. 2 briefly lists the state of the art. In Sect. 3, we present the system architecture and sensor integration. Finally, the paper concludes with a summary and an outlook on future work.

2 State of the art

Today, related to sensors and sensor networks a lot of research work has been done in the field of energy management (Wang, 2006; Emmert and Staehle, 2007; Wang and

Correspondence to: J. Schneider
(schneider@eit.uni-kl.de)

Published by Copernicus Publications on behalf of the URSI Landesausschuss in der Bundesrepublik Deutschland e.V.

Yang, 2007). In particular, wireless sensor networks have become a popular target for several energy management concepts. Although, our proposed concept has the potential to reduce energy consumption in wireless sensor networks, the focus of our approach is on easily integrating different sensors and actuators in the present and future Internet structure. The project C-Cast (C-Cast, 2010) introduced a centralized context management architecture. This architecture consists of Context Providers (e.g. wireless sensor networks), Context Consumers (e.g. actuators for system adaptations) and a Context Broker. The Context Broker acts as a central agent. Therefore, all Context Providers have to send a registration to the Context Broker with their capabilities (available sensor values). However, this architecture does not account for privacy issues. Zuniga (Zuniga and Krishnamachari, 2003) proposed two different solutions to integrate sensors into the Internet. In the first approach, a direct connection is used, i.e. each sensor is directly connected to the Internet and can be reached via its IP address. In the second approach, an indirect connection is used; this means a gateway is responsible for providing an interface between the Internet and each sensor. Especially for home automation, a special EIB/KNX (KNX, 2010) bus is used to connect different actuators with each other. Communication is performed via a serial bus and each actuator is assigned a unique address. Main drawback is the enclosed system communication, since bus specification is restricted to KNX Association members and license fees are expensive. For managing sensor data, a distributed Context Service Architecture has been published in (Mannweiler, 2010) and describes the design as well as the implementation of a highly scalable solution for a context service registry in mobile environments. This overcomes the bottleneck of centralized context management systems. Additionally, this system has successfully been tested with five small sensor networks, each disposing of three to five individual sensors. Our designed architecture also aims at cooperating with such context service architectures and service delivery platforms.

3 System architecture and sensor integration

Sensor networks are an emerging research topic and have a wide range of potential applications. Hence, we present a comprehensive architecture to cover multiple application fields. Our introduced system is capable of supporting applications and services which use sensor data for controlling actuators and system adaptations. Services are able to access sensor information of different SCPs via open and standardized interfaces, that also enable to integrate new sensors, irrespective of manufacturer. Thus, the system is extensible in a “Plug and Play” manner. Figure 1 illustrates the generic system structure.

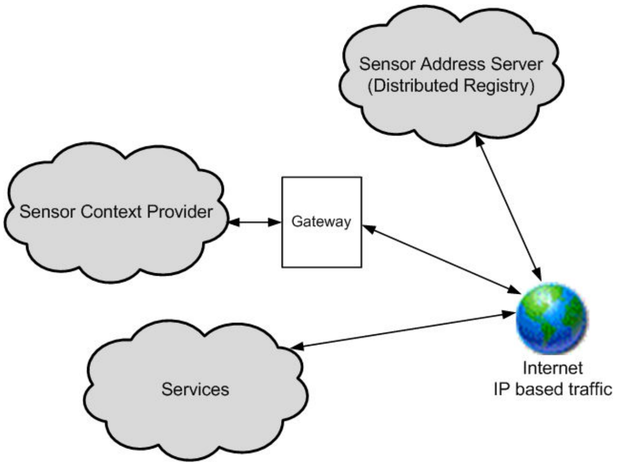

Fig. 1. Generic system structure.

Our system consists of the following functional components:

- Sensor Context Providers (SCP): These entities provide sensor context information which is gathered by sensors or sensor networks.
- Gateway: The gateway implements the interface between SCP(s) and the Internet.
- Sensor Address Server (SAS): All SCPs are registered at this entity and the SAS will choose an appropriate SCP on sensor data requests.
- Service: These entities are able to request and subscribe for sensor data. Logical functions are used for combining these values and exploiting them e.g. for system adaptations. All services need to be advertised to a node of the registry located in the Local Area Network or to the distributed registry of the Wide Area Network, respectively.
- Actuators (AC): These entities are used to control external devices like heaters, light, etc.

To achieve a high reusability of deployed hardware, each SCP and AC requires a network connection to a Local Area Network (LAN). Multimedia data, sensor context data as well as other Internet services can use the same infrastructure. Additionally, installation costs for large buildings will be reduced by using wireless LAN. For sensor data abstraction and exchange, our concept applies a three layer sensor context model as described in Table 1.

Raw sensor data (layer 0) is directly gathered by sensors and provided without any data processing. Layer 1 context data is derived from one or more layers, i.e. layer 0 and/or layer 1. The minimum requirement for a SCP is to provide sensor context data of layer 0. In the following subsection,

Table 1. Three layer context model.

Context layer	Description
0	Raw sensor context data without any processing
1	Abstracted context data with additional meta information
2	Service-level abstraction

more detailed information on SCPs will be depicted. In contrast, services (layer 2) use sensor context data from layer 0, 1 and 2 for their actual purpose.

3.1 Sensor Context Provider (SCP)

The Sensor Context Provider (SCP) can be seen as a sensor context source, since this entity is responsible for gathering and provisioning raw sensor data. A SCP is not restricted to deliver only values of one sensor, rather most of these entities will provide a set of sensor values and parameters. However, each SCP must be capable of supporting the respective bus or hardware interface of its attached sensors and sensor networks. For example, a sensor network can also be seen as a SCP and directly connected to a gateway. For provisioning of sensor data, we decided to use an indirect connection, i.e. a gateway is responsible for implementing an interface between the Internet and each SCP. This solution provides more flexibility, since sensor data processing is accomplished in the gateway.

3.2 Gateway

The gateway implements an open interface for providing raw or abstracted sensor data to external services and applications via the Internet. By means of a standardized XML-derivative, different access rights, and security concepts, sensor data information can be reliably exchanged, taking privacy issues into account. Each gateway must provide the following functionalities:

- Registration of all attached sensor nodes with their capabilities (available values, accuracy, sensor location, etc.)
- Detection of arrival and departure of sensors
- Announcement and request of sensor values to the Sensor Address Server via corresponding node
- Provision of sensor data (Context Layer 0)
- Optional: sensor data processing to derive higher level context data (Context Layer 1)

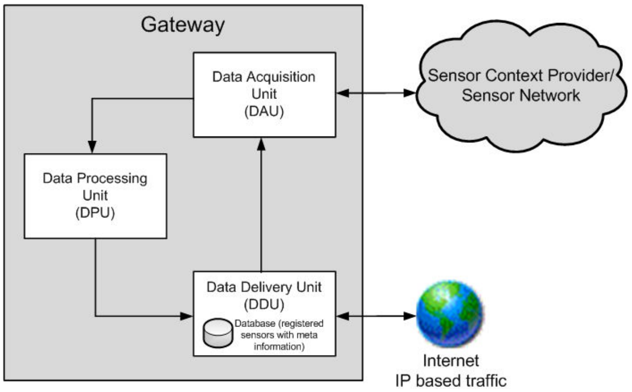
Gateway
Data Acquisition Unit (DAU)
Data Processing Unit (DPU)
Data Delivery Unit (DDU)
Database (registered sensors with meta information)
Sensor Context Provider/ Sensor Network
Internet
IP based traffic

Fig. 2. Generic gateway model.

A generic gateway model is shown in 2.
The gateway can be split into three main functional blocks:

- Data Acquisition Unit (DAU): This unit implements the hardware interface between sensors and gateway, and is responsible for acquiring raw sensor data from SCP(s).
- Data Processing Unit (DPU): The complexity of this unit depends on the Context Layer. To provide layer 0 context, sensor data will only be formatted in an applicable XML structure and delivered. In contrast, if layer 1 context can be requested, sensor context data will be generated via combining and reasoning of current and previous sensor values. For this procedure, only sensor data from the respective sensor network or SCP can be used. Gateways cannot request sensor data from other gateways. Moreover, gateways, that provide layer 1 context, can also provide their raw sensor data, used for derivation of layer 1 context, as layer 0 context.
- Data Delivery Unit (DDU): The DDU module provides the interface between gateway and Internet, and is responsible for registration with SAS. It implements a “request/provide” as well as an asynchronous “publish/subscribe” mode. Both control and sensor data have to pass this module. Moreover, it disposes of a database with all currently connected sensors and their meta-information. Additionally, the DDU is responsible for mapping and transferring a sensor data request to the according SCP and managing all sensor data subscriptions, taking access rights and privacy issues into account.

3.3 Sensor Address Server (SAS)

An important entity is the so-called Sensor Address Server (SAS). It is responsible for administrating a list of registered gateways, and for coordinating and mediating sensor data requests. The SAS is implemented as a hierarchical, zone-oriented structure comparable to the Domain Name Service

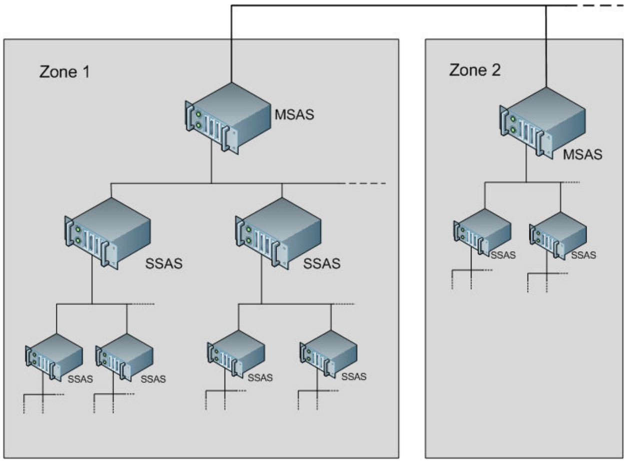

Fig. 3. Structure of Sensor Address Server (SAS).

(DNS) (Mockapetris, 1987). It groups sensors and SCPs depending on their location and allocates suitable sensor gateways to requests (e.g. by services). Figure 3 shows the system structure of the SAS. A more detailed description is given in (Schneider, 2010).

At system start-up, each component in the network allocates a co-located SAS. Therefore, each client or gateway sends a "getSAS" request to a known broadcast address. This address includes the current location of the client and a parameter "mobile", indicating whether the client is mobile or not. In case of a mobile client, an appropriate SAS is chosen with respect to the current position of the client. To ensure an up-to-date allocation of the SAS, each mobile client has to renew its announcement after a certain time interval. With the allocation of a SAS to a client, sensor data can be announced to as well as requested from the SAS. This includes sensor data and meta-information. Each announcement is valid for a certain time interval and has to be renewed. To ensure privacy issues, each SCP or gateway, respectively, can choose between two different privacy modes. In the "private-mode", advertised sensor information is restricted to the zone-internal SAS. Hence, sensor data can only be requested inside the local sub-network (LAN). If a gateway chooses the "public-mode", advertised sensor data is forwarded to an appropriate SAS in the Internet. Hence, sensor data and context, respectively, can be requested inside the sub-network as well as from the Internet (WAN). For managing the dynamic IP addresses of the clients, a mapper is used to assign the IP address to a fixed URL. Appropriate management services are e.g. DynDNS (dyndns, 2010), 2myDNS (2mydns, 2010), no-IP (no-ip, 2010). A generic model of the system architecture is shown in Fig. 4.

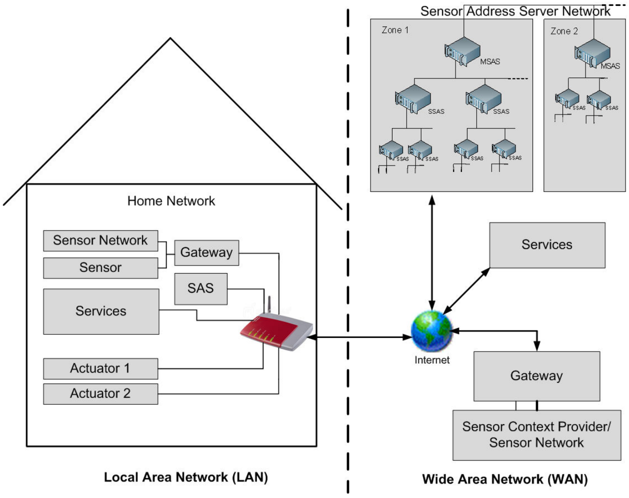

Fig. 4. Generic model of the system architecture.

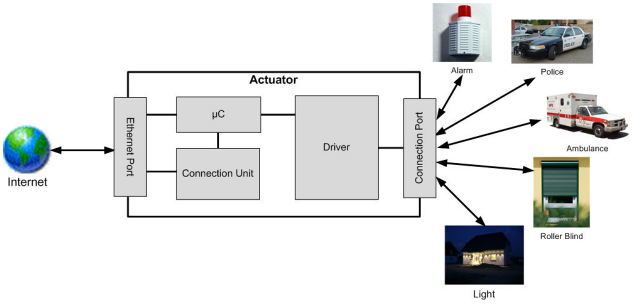

Fig. 5. Actuator block diagram.

3.4 Actuators (AC)

Actuators (ACs) execute commands of smart control systems, like alarm systems, for security or ambient assisted living, home automation, health care, etc. Each AC establishes an interface for connecting systems without bus/network capabilities. Figure 5 shows a block diagram of an actuator including examples for connected entities.

Actuators can only execute commands on request, where two different classes of commands are possible:

- Action Command: Actuator triggers e.g. an alarm.
- Control Command: Actuator receives a control command from a service and replies with its current actuator status or an error message.

An action command triggers an event on the connection port of the actuator. Usually, these ports are connected to other entities like alarm generator, roller blinds, etc.

3.5 Services

Services are the controlling entities in our architecture. With the capability to request and subscribe for sensor data, as well as to control actuators, they act as middleware. Logical functions such as data combining and reasoning will be applied in these entities. As mentioned in Sect. 2, Mannweiler (Mannweiler, 2010) introduced a design for a distributed service registry. To extend this system to a distributed service architecture, we include a privacy concept. Therefore, an additional registry node is added to the Local Area Network (LAN), for registering all services inside the LAN. If a service is marked as public, its registration gets forwarded to the public distributed registry node, located either in the same SAS zone as the LAN's SAS or a different SAS zone. This concept allows the user to restrict services to the local subnetwork. If a user decides to make it public, the service registration is forwarded to the Wide Area Network's (WAN) SAS.

4 Conclusions

This paper presented the design of a highly scalable and extensible solution for the integration of sensors and actuators into the future Internet, that overcomes the weakness of locally restricted sensor platforms or centralized sensor and actuator management architectures by means of open and standardized interfaces, and Sensor Address Servers. Further, privacy modes allow users to decide if their sensor data and/or services are available only for private or public use. In combination with security concepts, the proposed system architecture ensures reliable exchange of sensor data and meta-information for many important application areas, such as service adaptations in fixed and mobile environments, ambient assisted living, home automation, traffic management, as well as management of smart grids. Additionally, the partitioning into SCPs, gateways, services, and actuators allows for eased integration of new and rather low-cost components, since complex data processing and reasoning algorithms are shifted to service components. Thus, many components of the architecture, e.g. actuators and SCPs, can be constituted by mass market products. Furthermore, the use of the Internet as comprehensive transport medium ensures low installation costs, too. Therefore, these features will increase the attractiveness of the presented approach and hence the market acceptance.

Future work will include three areas: First, implementation of a Sensor Address Server network and the introduced privacy concept. Second, combining of the SAS network and the Distributed Registry. Finally, large scale tests with a high number of sensors, actuators, and services with different privacy levels will be performed.

Acknowledgements. This work has been supported by the Federal Ministry of Education and Research of the Federal Republic of Germany (Foerderkennzeichen 01 BK 0808, GLab; german-lab, 2010).The authors alone are responsible for the content of the paper.

References

Wang, Q., Hempstead, M., and Yang, W.: A Realistic Power Consumption Model for Wireless Sensor Network Devices, SECON '06 3th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2006.

Emmert, B. and Staehle, D.: Impact of Energy Models on Energy Efficient Sensor Network Routing, University of W—rzburg, Institute of Computer Science, Technical Report No. 415, April 2007.

Wang, Q. and Yang, W.: Energy Consumption Model for Power Management in Wireless Sensor Networks, SECON '07 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007.

C-Cast Context Casting, http://www.ict-ccast.eu/, last access: March 2010.

KNX Association, http://www.knx.org/, last access: Mai 2010.

Zúniga, M. and Krishnamachari, B.: Integrating Future Large-scale Wireless Sensor Networks with the Internet, USC Computer Science Technical Report CS 03-792, 2003.

Mannweiler, C., Amann, B., Schneider, J., Klein, A., and Schotten, H. D.: A Distributed Context Service Architecture for Heterogeneous Radio Environments, ITG-Fachbericht der 15, ITG Fachtagung Mobilkommunikation, Osnabrück, 2010.

Schneider, J., Mannweiler, C., Klein, A., and Schotten, H. D.: Einbindung von Sensoren und Sensornetzwerken in das Future Internet, ITG-Fachbericht der 15. ITG Fachtagung Mobilkommunikation, Osnabrück, 2010.

Mockapetris, P.: Domain Names – Concepts and Facilities, IETF, RFC 1034, http://tools.ietf.org/rfc/rfc1034.txt, November 1987.

Mockapetris, P.: Domain Names – Implementation and Specification, IETF, RFC 1035, http://tools.ietf.org/rfc/rfc1035.txt, November 1987.

http://www.dyndns.com, last access: January 2010.

http://www.2mydns.com, last access: January 2010.

http://www.no-ip.com, last access: January 2010.

http://www.german-lab.de, last access: January 2010.