D4.3.1 Smart Object Integration (Prototype I)

Authors: TUDA, ASCORA

Delivery Date:	Due Date:	Dissemination Level:
2013-03-06	2013-02-28	PP

This deliverable provides a description of the first prototype implementation of task T4.3. As stated in the Description of Work (DOW), this deliverable is a prototype deliverable. As such, this document is reduced in length and its only purpose is to briefly describe the prototype functionality and give usage instructions and clarifications. This document is an accompanying document to the developed hard- and software solution.

Document History	
Draft Version	V0.1, TUDA, December 04th, 2012 V0.2, TUDA, January 15th, 2013 V0.3, TUDA, January 29th, 2013 V0.7, TUDA, February 13th, 2013 V0.8, TUDA, February 14th, 2013 V0.9, TUDA, February 19th, 2013 V0.99, TUDA, February 21th, 2013 V1.0, TUDA, March 06th, 2013
Contributions	TUDA - Sebastian Zöller ASC - Sven Abels - Tim Dellas - Rafael Karbowski
Internal Review 1	UVI, February 20th, 2013
Internal Review 2	TIE, February 22nd, 2013
Final Version	March 06th, 2013

Table of Contents

List of Figures

Executive Summary

The prototype corresponding to this deliverable D4.3.1 represents the intermediate outcome of task T4.3 after 18 months of the project duration. Its main purpose is to provide a solution to integrate Smart Object technology in the ADVENTURE platform and thus allow the interconnection between the ADVENTURE system and Smart Objects.

In the considered context, wireless sensor network technology constitutes the basic building block for Smart Objects. Hence, the focus of the work is on providing means to transmit monitoring data from wireless sensors to the ADVENTURE platform, both by using event-based push communication and webservice-based pull communication.
Wireless sensors usually rely on short-range communication based for example on the IEEE 802.15.4 standard or the ZigBee standard. Thus, wireless sensors cannot be connected without an intermediary component to the ADVENTURE system. Android-based end devices are employed functioning as such intermediaries between a wireless sensor node and the ADVENTURE platform. In consequence, a distinct focus of the work has been put on interconnecting wireless sensor nodes and Android-based smartphones to provide for the required bi-directional communication channel between a wireless sensor node and the ADVENTURE platform.
The developed solutions comprise a data exchange solution employing the event-based push communication paradigm and a data exchange solution employing the pull communication paradigm relying on webservices. Within the push approach, the sensor node transmits current monitoring data to a smartphone-based Gateway from which it is currently written to the ADVENTURE Cloud Storage. Within the pull approach, a webserver is realized on the smartphone-based Gateway, which provides an interface to retrieve current monitoring data from the wireless sensor node.

As stated in the Description of Work (DOW), this deliverable is a prototype deliverable. As such, this document is reduced in length and its only purpose is to briefly describe the prototype functionality as well as to provide instructions on how to use it. This document is an accompanying document to the corresponding hard- and software.

1 Introduction

ADVENTURE – ADaptive Virtual ENterprise manufacTURing Environment – is a project funded in the Seventh Framework Programme by the European Commission. ADVENTURE creates a framework that enhances the collaboration between suppliers, manufacturers and customers for industrial products and services.

1.1 ADVENTURE Project Aims

The framework proposed by ADVENTURE provides mechanisms and tools that facilitate the creation and operation of manufacturing processes in a modular way. ADVENTURE combines the power of individual factories to achieve complex manufacturing processes. It provides tools for partner-finding, process creation, process optimization, information exchange as well as real-time monitoring combined with the tracking of goods and linking them to Cloud services.

There have already been several research projects that address the combination of different independent manufacturers to so-called virtual factories. Most of these research projects focus primarily on the business-side in general and on aspects like partner-finding and factory-building processes in special. However no proven tools or technologies exist in the market that provide the creation of virtual factories applying end-to-end integrated Information and Communication Technology (ICT). ADVENTURE is aiming to provide such tools and processes that will help to facilitate information exchange between factories and move beyond the boundaries of the individual enterprises involved. The collaborative manufacturing process will be optimised by enabling the integration of factory selection, forecasting, monitoring, and collaboration during runtime.

ADVENTURE builds on concepts and methods of Service-oriented Computing and benefits from the advancements in this field. The monitoring and governance of the collaborative processes will be supported by technologies from the Internet of Things such as wireless sensors. Existing tools and services that can be integrated will be considered during the development of the platform for ADVENTURE.

The increased degree of flexibility provided through ADVENTURE will benefit SMEs especially as it helps them to react quickly to changes and to participate in larger, cross-organizational manufacturing processes. Furthermore, ADVENTURE will help manufacturers in assessing the environmental friendliness of actual manufacturing processes and resulting products and services. Other objectives of ADVENTURE include research in areas such as service-based manufacturing processes, adaptive process management, process compliance, and end-to-end-integration of ICT solutions.

1.2 Deliverable Purpose, Scope and Context

The purpose of this deliverable is to accompany the prototype of task T4.3. As such, its main purpose is to briefly clarify the scope of the prototype and to explain its functioning and usage. The document is limited in length as the main focus of the prototype is the hard- and software solution rather than its accompanying document.

1.3 Document Status

This document is listed in the DOW as 'PP', which means 'Restricted to other programme participants (including the Commission Services)' primarily since the audience of the document is largely internal and it is not the final prototype version. It presents the first prototype of the Smart Object Integration and is mainly targeted to development teams and user partners, rather than external audience. The follow-up deliverable of this deliverable, which is due at project month M30, will comprise the extended second prototype and will be made publicly available for external dissemination.

1.4 Target Audience

This document constitutes the accompanying document to the first prototype of the Smart Object Integration. The document's and the corresponding prototype's main purpose is to provide primarily the project partners with a first version of the solution to integrate Smart Objects with the ADVENTURE platform as a basis to create a more detailed understanding of what the Smart Object Integration will provide in the project's context and how it works. Furthermore, it shall be used as input for discussion and coordination of the work for integrating it into the overall ADVENTURE platform to achieve a fully integrated system at the end of the project. In consequence, the target audience are mainly the project partners and more specifically the involved development teams and relevant use case partners.

1.5 Abbreviations and General Terms

A definition of general, common terms and roles related to the realization of ADVENTURE as well as a list of abbreviations is available in the supplementary document "Supplement: Abbreviations and General Terms" which is provided in addition to this deliverable.
Further information can be found at: http://www.fp7-adventure.eu

1.6 Document Structure

This deliverable is broken down into the following sections:

Section 1 provides a short overview of the ADVENTURE project and this deliverable's context.

Section 2 outlines the scope of the developed Smart Object Integration prototype and its position within the overall ADVENTURE architecture.

Section 3 specifies the different components of the Smart Object Integration prototype, their functioning, and initial setup.

Section 4 describes the different steps to be taken to deploy the developed Smart Object Integration solution.

Section 5 presents the functional principles of the developed prototype for Smart Object Integration and the different ways, users can make use of it.

Section 6 highlights the limitations of the current prototype solution and presents corresponding future work.

Section 7 provides a summary of this deliverable.

2 Scope and Relationship

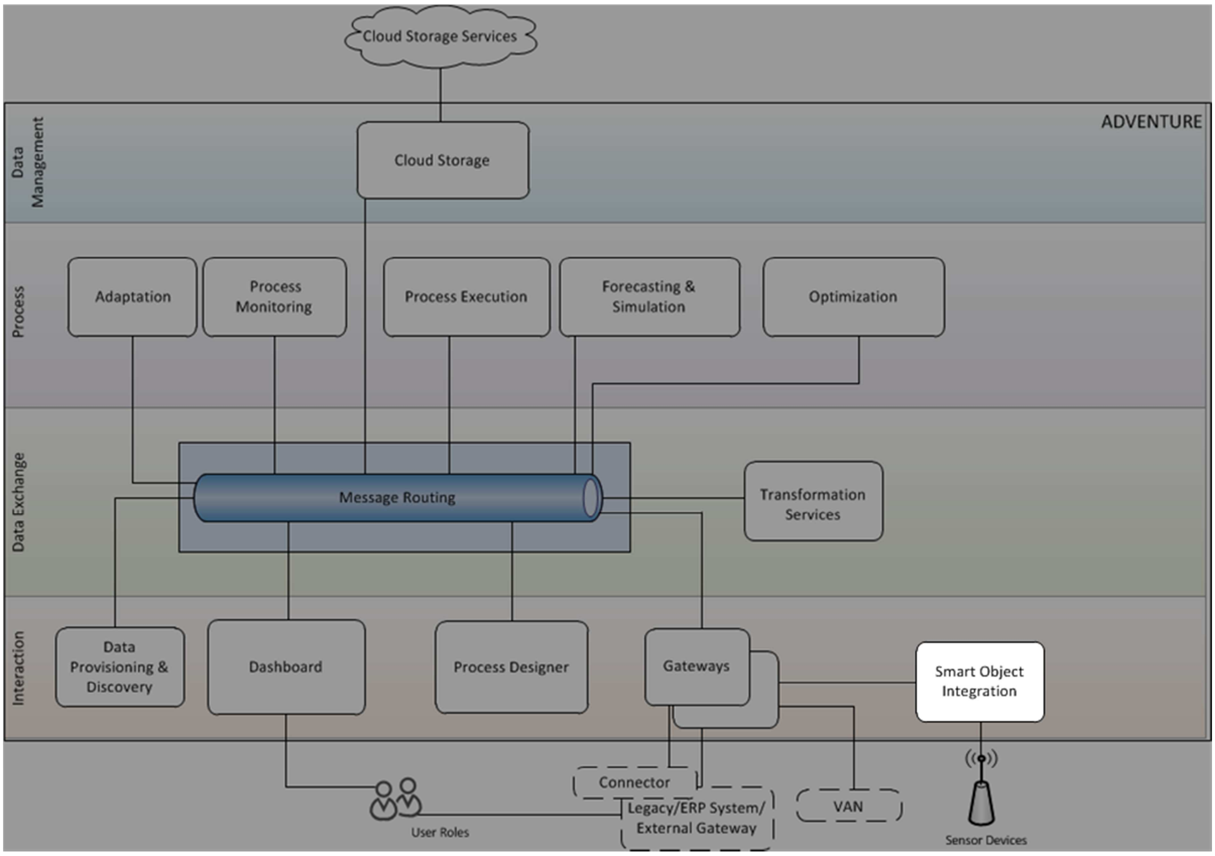

Figure 1: The Smart Object Integration within the context of the overall ADVENTURE architecture

The developed Smart Object Integration prototype is responsible for the integration of Smart Objects into the ADVENTURE platform. This means, it provides communication channels between Smart Objects and the ADVENTURE system and corresponding mechanisms to retrieve monitoring data from Smart Objects.

Wireless sensor network technology has been identified as basic building blocks for Smart Objects. Therefore, the developed Smart Object Integration solution provides in particular data transmission possibilities between wireless sensor nodes and the ADVENTURE platform. Because wireless sensor nodes usually employ only short-range communication, commonly based on the IEEE 802.15.4 standard[1] or the ZigBee wireless communication standard[2], an intermediary is required to provide the connection between such a Smart Object and the ADVENTURE system. For this

[1] http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf
[2] http://www.zigbee.org/Specifications/ZigBee/download.aspx

purpose, the developed prototype makes use of an Android-based end device, specifically an Android smartphone.

In consequence, the Smart Object Integration provides the bridge between a Smart Object deployed for monitoring purposes and the ADVENTURE platform employing a corresponding Gateway component. For demonstration purposes, the current prototypical implementation of the Smart Object Integration provides a connection to the ADVENTURE Cloud Storage Service to store the gathered monitoring data there. In the final, fully integrated version, a connection with the Smart Process Execution component will be realized, which will then despatch the data to its point of destination.

3 Requirements & Preparations

The requirements and preparation procedures depend on the one hand on the employed hardware and on the other hand on the communication paradigm to be realized. The developed solutions provide prototypical implementations for the TelosB wireless sensor node platform[3], the SunSPOT wireless sensor node platform[4], and Pikkerton's multisensory platform ZBS-121[5] for event-based push communication and webservice-based pull communication. Different smartphones have been employed as Android-based end devices, for example a Galaxy Nexus[6] smartphone and a Galaxy S[7] smartphone, to be able to test the compatibility and identify peculiarities of the developed solutions on different smartphone hardware.

Wireless sensor nodes typically make use of wireless short-range communication based on the IEEE 802.15.4 standard or the ZigBee standard. However, none of these standards is supported by current smartphones. Current smartphones usually employ Bluetooth as one predominant means for wireless local communication with other devices. Consequently, possibilities to bridge this communication gap are needed. Basically, these can be divided in extending wireless sensor platforms with Bluetooth capabilities on the one hand or on the other hand realizing an extension for smartphones, which provides them with IEEE 802.15.4, respectively, ZigBee communication capabilities.

The developed solution realizes these two different possibilities and consequently provides means to enable Bluetooth-based communication on wireless sensor nodes and IEEE 802.15.4-based, respectively ZigBee-based, communication on smartphones. Consequently, differing preparations for the hardware setup are required, which are described in the following.

[3] http://memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=152:telosb
[4] http://www.sunspotworld.com/docs/Yellow/eSPOT8ds.pdf, http://www.sunspotworld.com/docs/Yellow/edemo8ds.pdf
[5] http://www.pikkerton.com/ZigBee/ZigBee_MultiSensor.html
[6] https://play.google.com/store/devices/details/Galaxy_Nexus_HSPA?id=galaxy_nexus_hspa
[7] http://galaxs.samsungmobile.com

3.1 Enabling IEEE 802.15.4-based Communication on a Smartphone

For realizing IEEE 802.15.4-based communication on a smartphone, a corresponding hardware extension is required. Such extension modules can be plugged to the USB port of the device employing a so-called USB On-The-Go cable and exploiting the USB host mode of the smartphone. Consequently, the data exchange is then realized via the phone's USB port, which means that data that is received by the extension module from a wireless sensor node and is then transmitted to the smartphone via its USB port. Vice versa, data which has to be transmitted from the smartphone to a wireless sensor node is sent via the USB port to the extension module and then from the extension module to the wireless sensor node. Figure 2 depicts the developed solution in this context, which makes use of an X-Tick extension module[8] connected to the smartphone's USB port.

Figure 2: Prototype for IEEE 802.15.4-enabled smartphone based on an X-Tick hardware extension

3.2 Enabling ZigBee-based Communication on a Smartphone

To provide smartphones with ZigBee-based communication capabilities, corresponding hardware extensions can be plugged to the phone's USB port (analogous to the description in Section 3.1). Again, a USB On-The-Go cable in combination with the phone's USB host mode is used to connect the external extension module to the smartphone. Consequently, the following communication takes place by shipping received data from the ZigBee-capable extension module via the USB port to the phone and vice versa shipping data which has to be sent by means of ZigBee communication to the extension module via the phone's USB port. Figure 3 illustrates the prototypical

[8] http://www.cooking-hacks.com/index.php/shop/arduino/x-tick-xbee-programmable-usb-stick.html

solution devised in this context, which employs an XStick ZB module[9] connected to the smartphone's USB port.

Figure 3: Prototype for ZigBee-enabled smartphone based on an XStick ZB hardware extension

3.3 Enabling Bluetooth Communication on a Wireless Sensor Node

Several extension modules equipped with Bluetooth transceivers (transmitter and receiver in one unit) exist in the market. Most often, they rely on the basic idea that they are attached to a wireless sensor node via its UART/USART[10] interface and just take the incoming data streams via this interface, generate Bluetooth packets with the content received via UART/USART and transmit these Bluetooth packets to a corresponding receiver. Our prototype employs the BlueSMiRF Gold-module[11], which provides a quite satisfying cost-performance ratio. As mentioned, this extension module

[9] https://www.digi.com/products/wireless-modems-peripherals/wireless-range-extenders-peripherals/xstick#overview

[10] UART (Universal Asynchronous Receiver Transmitter), respectively USART (Universal Synchronous/Asynchronous Receiver Transmitter), realizes a digital serial interface for data exchange, which allows establishing serial data communication between two components. For example, in the considered context, a wireless sensor module provides UART/USART pins to which extension modules, like a Bluetooth module or additional sensors, can be wired. The communication between the wireless sensor module and the extension module is then realized by a standardized serial communication with according data streams.

[11] http://www.cooking-hacks.com/index.php/bluetooth-modem-bluesmirf-gold.html

has to be attached to the wireless sensor node by wiring it to the node's UART/USART port to provide the required Bluetooth capabilities. Figure 4 shows the resulting hardware setup of wiring a BlueSMiRF Gold-module to a SunSPOT as an example.

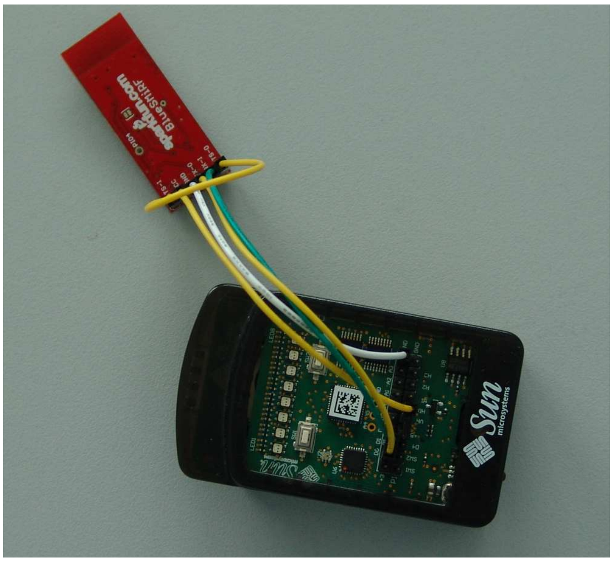

Figure 4: BlueSMiRF Gold-module wired to a SunSPOT wireless sensor node

4 Installation (Deployment)

4.1 Event-based Push Communication

In the prototype setup for the event-based push communication, a Galaxy S smartphone has been employed. The Helly Bean ROM[12] has been flashed to the smartphone in order to provide the required USB host mode and successfully attach the required extension module for ZigBee-based push communication to it. Furthermore, since the Galaxy S does not natively support USB host mode, it is incapable of providing power supply to devices attached to its USB port, even with the employed Helly Bean ROM. Thus, for the current prototype setup, the power supply for the XStick extension module needs to be provided externally. For this reason, the XStick currently has to be attached to the smartphone via a custom built Y-adapter, which comprises a female USB and a male USB plug as well as a USB On-The-Go plug leading to a hardware setup as illustrated in Figure 5.

[12] http://hellybean.com

Figure 5: Illustration of an XStick connected to a smartphone with an external power source

Having flashed the smartphone accordingly and provided the hardware wiring as described, the developed app has to be installed and started on the smartphone and an Internet connection provided. In this first prototype, the app takes care of receiving the data correctly from a wireless sensor node and provides the Gateway functionality of pushing the received monitoring data to the ADVENTURE Cloud Storage component using the ADVENTURE Message Routing component. Within the ADVENTURE Cloud Storage, the delivered monitoring data it can be viewed with the corresponding database monitoring tool. With this, the prototype is ready to be used (cf. Section 5.1).

4.2 Webservice-based Pull Communication

In the current prototype for the webservice-based pull communication, i-jetty[13] has been employed as a webserver in order to manage incoming data requests. The i-jetty web server is installed on a rooted Galaxy Nexus smartphone, which runs Android version 4.1. In consequence, the phone needs to be rooted to acquire required superuser rights. Afterwards, i-jetty has to be installed on the smartphone and equipped with the developed servlet. This is done by deploying the servlet via i-jetty's download button and specifying the corresponding installation path and download URL. Then, the developed "XtickApp" has to be installed on the phone. The app takes care of accessing the monitoring data from the X-Tick extension module.

As described in Section 3.1, an X-Tick extension module is used to equip the smartphone with IEEE 802.15.4-based communication capabilities. Thus, the smartphone's USB host mode has to be enabled and the X-Tick module has to be connected to the smartphone's USB port via its USB cable and an according USB-On-the-Go cable. Afterwards, the X-Tick module can be switched on and then the i-jetty webserver app and the XtickApp can be started. Having performed these steps and established an Internet connection, the prototype is ready to be used (cf. Sections 5.2.1 and 5.2.2).

[13] http://code.google.com/p/i-jetty/

5 Execution & Usage

5.1 Retrieving Monitoring Data with Event-based Push Communication

The prototype for retrieving monitoring data employing event-based push communication has been realized using the multisensory platform ZBS-121 from Pikkerton. Consequently, it constitutes a proof-of-concept realization for integrating a ZigBee-based wireless sensor node with the ADVENTURE system.

The current prototype setup comprises a rooted Galaxy S smartphone with an XStick ZB attached to it. Due to some restrictions of the Galaxy S smartphone which lead to the situation that the employed XStick cannot be supplied with power via the smartphone's USB port, an external power supply has to be provided in the current version of the prototype as already indicated. The resulting setup is depicted in Figure 6. However, several options to overcome the problem with the required external power supply for the XStick exist, like the usage of another smartphone platform, and will be realized in the next iterations of the prototype development.

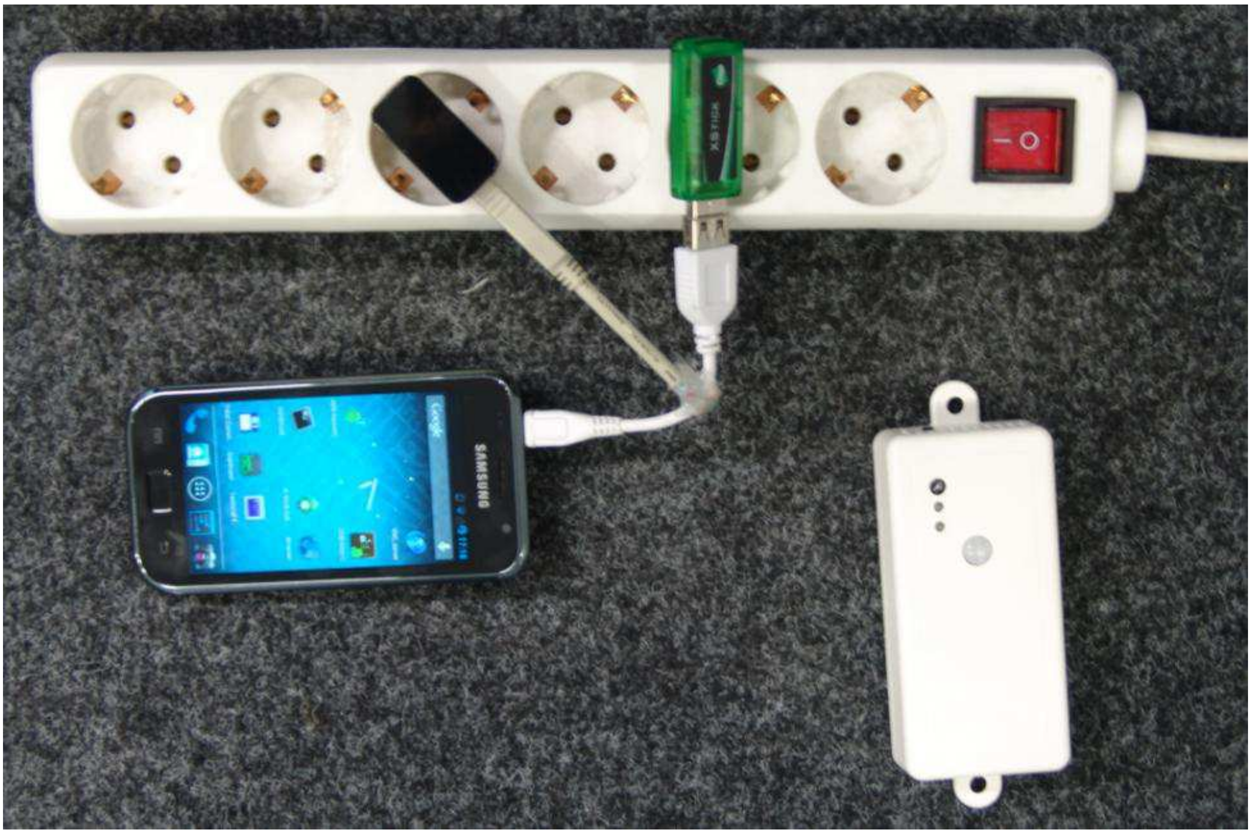

Figure 6: Prototype setup for event-based push communication employing an XStick ZB and a Pikkerton ZBS-121 wireless sensor node

With the currently devised prototype setup, the turned on Pikkerton sensor is transmitting environmental data, namely temperature and light values as exemplary measurement values, via ZigBee communication to the XStick ZB. The developed app

on the Galaxy S retrieves the incoming data from the XStick via the USB port and transmits it via wireless Internet connection, e.g., WLAN, to the ADVENTURE Cloud Storage component. There it is stored in a corresponding bucket. The user gets the received monitored data visualized by the developed app directly on the smartphone. Furthermore, the data transmitted and stored in the Cloud Storage Component can be viewed via the Cloud Storage's management interface (cf. Figure 7).

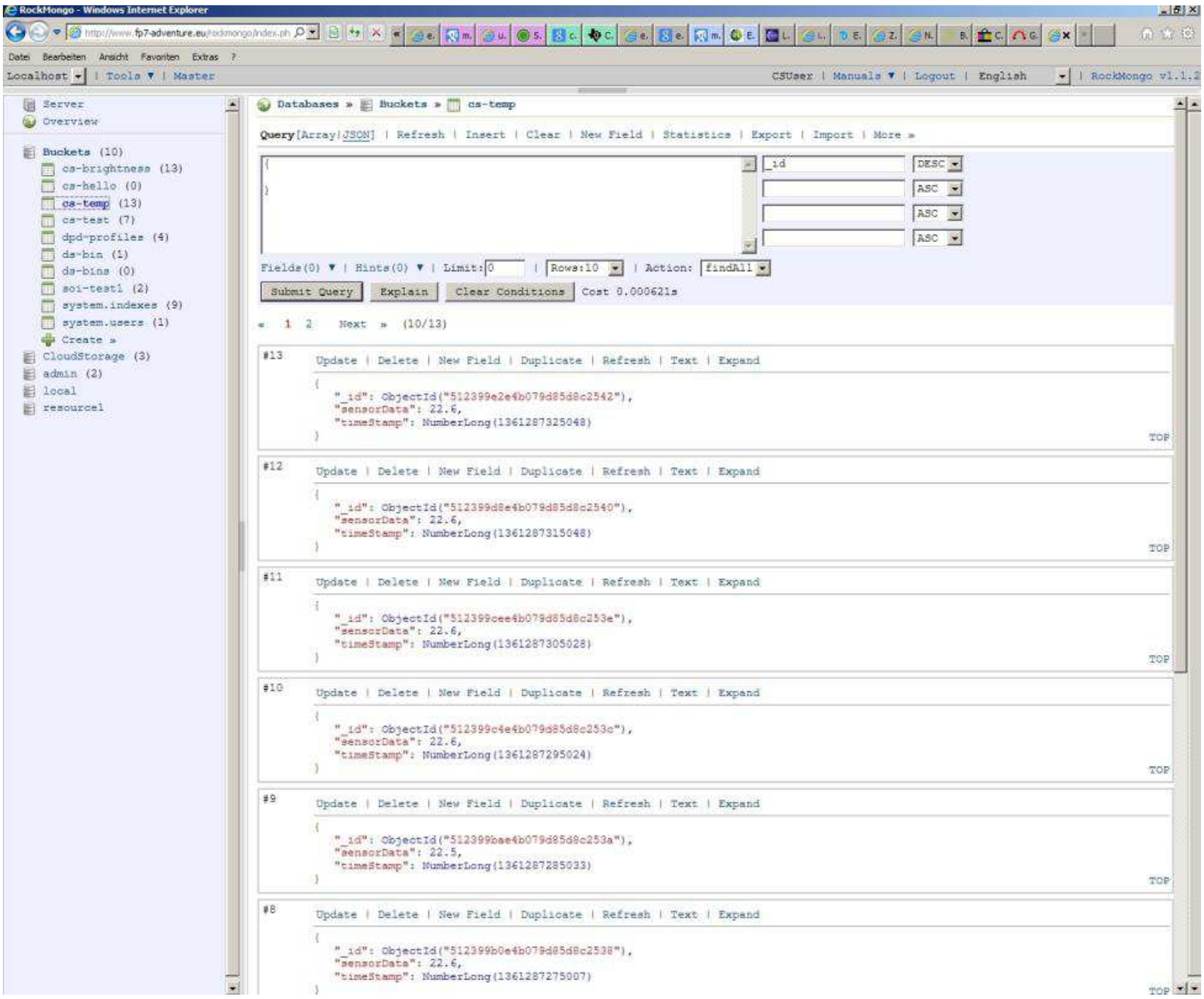

Figure 7: Screenshot of the ADVENTURE Cloud Storage management system displaying a data bucket filled with received temperature monitoring data

5.2 Retrieving Monitoring Data with Webservice-based Pull Communication

The current prototype solution for pulling a wireless sensor node's monitoring data based on a webservice approach relies on the usage of the i-jetty webserver[14] on an Android-based smartphone. This webserver satisfies data requests by reading received monitoring data from the smartphone's memory and providing according reply messages to the received query containing the inquired monitoring data. This functional

[14] http://code.google.com/p/i-jetty/

principle remains the same, independent of the employed wireless sensor node platform.

As data sources for the monitoring data, we employed in the current prototype two different wireless sensor node platforms, namely the TelosB platform and the SunSPOT platform. Thus, we provide two proof-of-concept implementations for integrating different sensor nodes, which make use of IEEE 802.15.4-based communication. Correspondingly, in the following we divide the description in two parts. On the one hand, we cover the prototype using the TelosB platform and on the other hand the prototype using the SunSPOT platform, which both allow capturing temperature and light data as basic sensor data, but can easily be extended to capture other parameters, like humidity, acceleration etc. The employed Android-based smartphone remains the same in the two prototype settings. In both settings, we employed a rooted Galaxy Nexus, as already described.

5.2.1 Prototype Execution using the TelosB platform

As described, the prototype is based on a Galaxy Nexus smartphone running the i-jetty Webserver and an X-Tick extension module connected to the phone's USB port to realize the IEEE 802.15.4-based wireless communication. The resulting prototype setup with a TelosB wireless sensor node is shown in Figure 8.

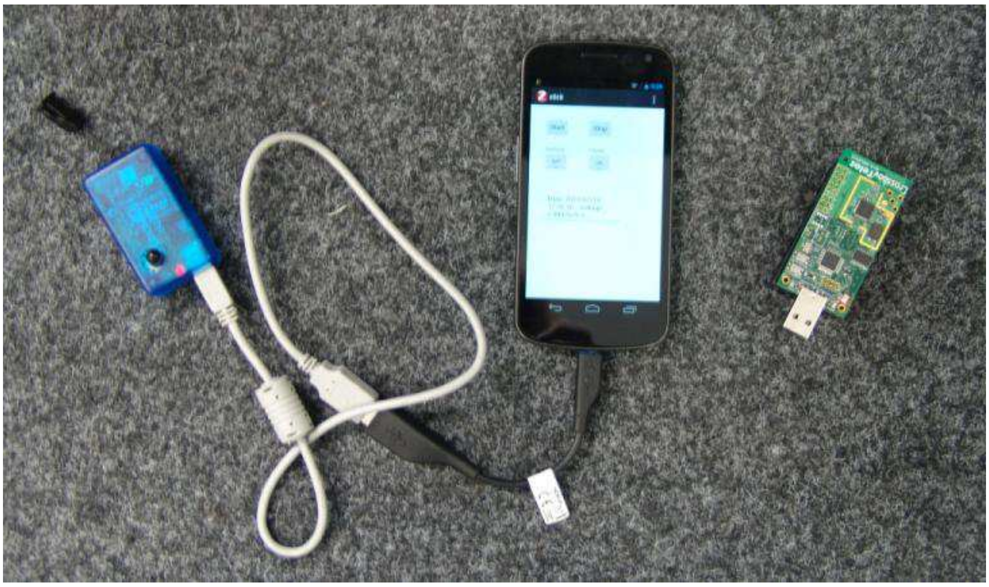

Figure 8: Prototype setup for webservice-based pull communication employing an X-Tick and a TelosB wireless sensor node

After having turned on the X-Tick module and the TelosB wireless sensor node, the webservice within the developed app needs to be started up. Afterwards, the correct wireless sensor node platform, in this case the TelosB platform, has to be chosen within the app (cf. Figure 9). Finally, the received monitoring data from the TelosB node, for example temperature and light values, is displayed within the app and accordingly stored on the phone's SD-card.

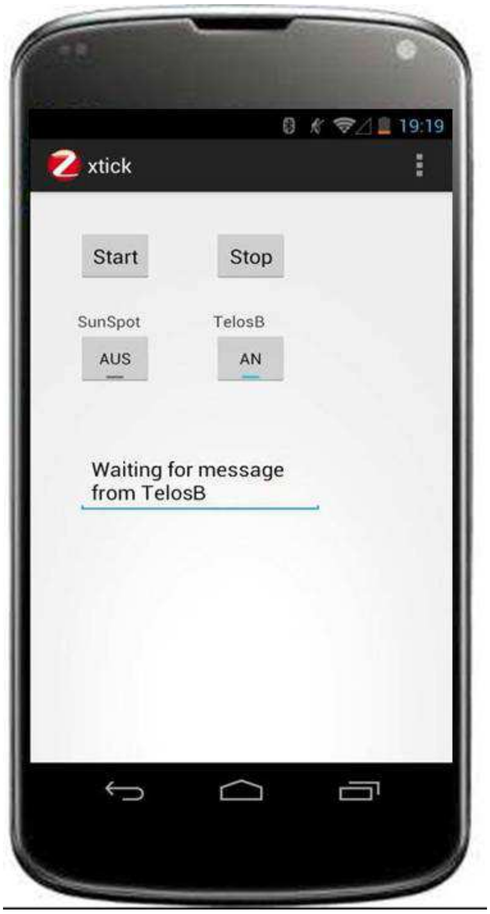

Figure 9: Screenshot of developed XtickApp for webservice-based push communication with wireless sensor nodes

Users can now query the monitoring data via a web browser and the smartphone's IP address and get the current data displayed on a corresponding website (cf. Figure 10).

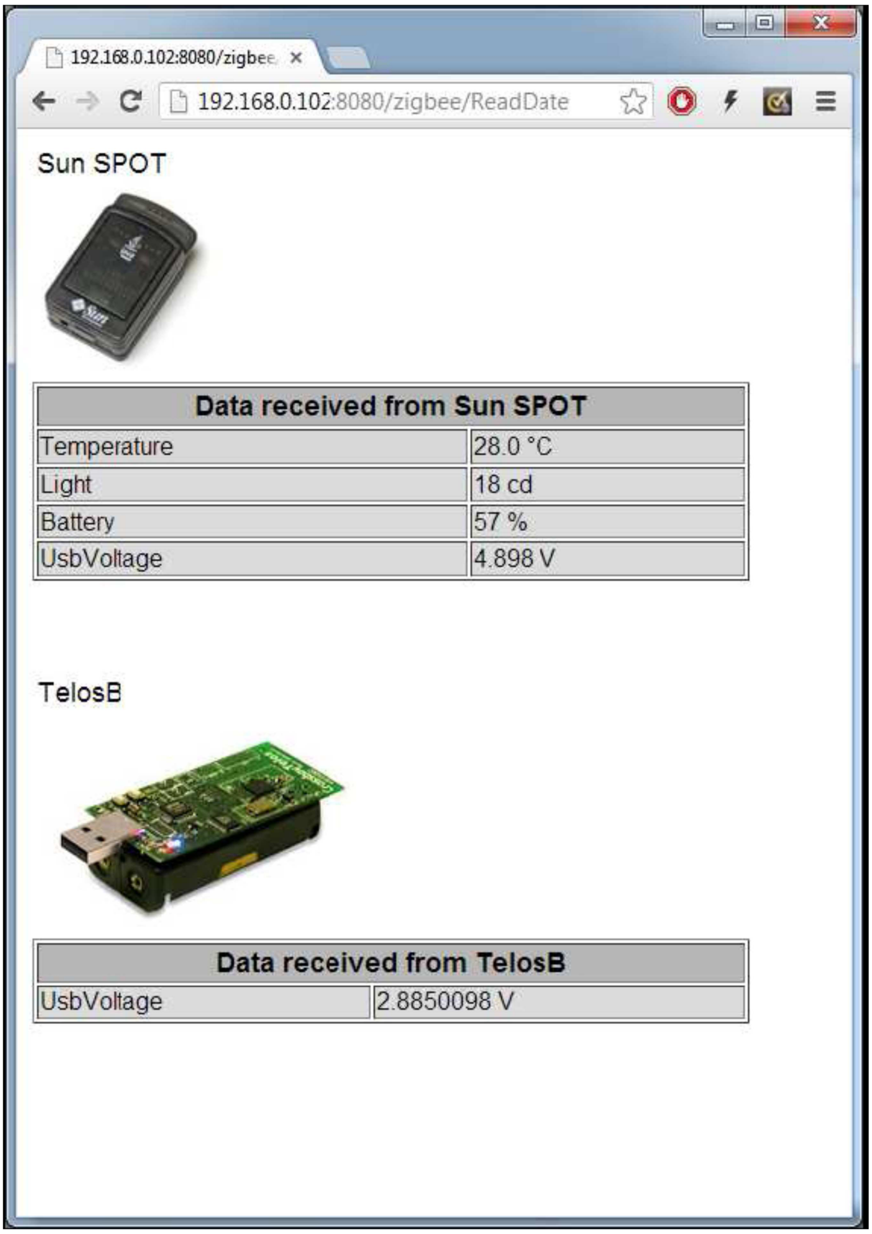

Figure 10: Webpage displaying monitoring data retrieved by a webservice-based pull communication

5.2.2 Prototype Execution using the SunSPOT platform

The prototype developed for the SunSPOT platform works similar as the prototype described in Section 5.2.1 for the TelosB platform. However, in this prototype setup a SunSPOT node is wirelessly connected to the smartphone via the X-Tick extension module, which results in the setup depicted in Figure 11.

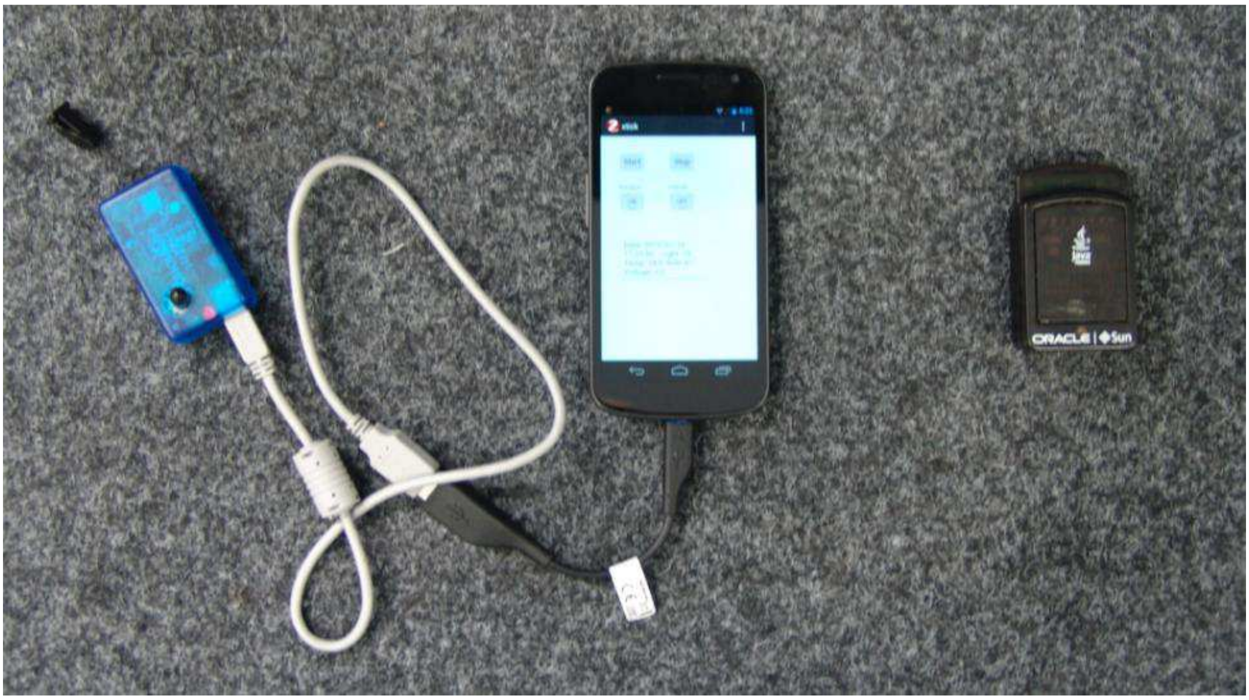

Figure 11: Prototype setup for webservice-based pull communication employing an X-Tick and a SunSPOT wireless sensor node

In consequence, within the Android app, the SunSPOT has now to be chosen as wireless sensor platform. Afterwards, data is received by the smartphone, displayed in the app, and provided to users in the same way as described for the prototype using the TelosB platform (cf. Section 5.2.1 and Figure 10).

6 Limitations & Further developments

To provide a first proof-of-concept implementation of the developed solution for integrating Smart Objects with the ADVENTURE platform, a possibility for storing monitoring data originating from a wireless sensor node in the ADVENTURE Cloud Storage has been realized. However, from an architectural perspective and internal discussions on this topic, the interaction of the Smart Objects with the ADVENTURE system shall be realized via the Smart Process Execution component. Thus, an urgent next step will be to integrate the Smart Process Execution component as the destination and origin point for the communication between the ADVENTURE platform and connected wireless sensor nodes for process monitoring.

The focus of the current prototype was on realizing a basic bi-directional communication channel between the ADVENTURE system and a wireless sensor node. Having realized this goal, one distinct focus of further developments will be the integration of a network of wireless sensor nodes, i.e., providing communication means with multiple (different) wireless sensor nodes via one, respectively a few, intermediate nodes like the introduced Android-based smartphone.

A central goal for providing an integrating bi-directional communication channel between Smart Objects realized with wireless sensor technology and a corresponding backend system, like the ADVENTURE platform, is an energy-efficient operation. Thus, initial energy measurements for the current solution have been conducted. In future work, more energy measurements will be conducted to evaluate and potentially enhance the energy efficiency of the developed solutions. Furthermore, additional hardware will be integrated, which can lead as well to a more energy-efficient and even a more unobtrusive way for the integration of wireless sensor nodes with the ADVENTURE system. Additionally, the implementation of rule-based monitoring means on wireless sensor nodes is aimed at. Such approaches allow interpreting data locally on a wireless sensor node and sending data only in case it is relevant, opposed to periodic monitoring approaches, which rely on data transmissions according to specified time intervals independent of the gathered monitoring data. Consequently, rule-based monitoring allows reducing data transmissions, which enables lower energy consumption on a wireless sensor node and thus contributes to a more energy-efficient operation.

The developed solutions rely primarily on IEEE 802.15.4-based, respectively ZigBee-based communication, as such communication has already been shown to be more energy-efficient within the considered context. However, the initial prototypical setups for employing Bluetooth-based communication have been already provided and thus the developed solutions for pushing data to the ADVENTURE system and pulling data from the wireless sensor nodes will be ported to these Bluetooth-based setups. This allows a more in-depth analysis of these two technologies and their advantages and disadvantages.

Finally, when the functionality of data gathering by Smart Objects and corresponding data exchange with the ADVENTURE platform has been satisfactory realized, an investigation of security risks can be conducted and means to address them can be identified.

7 Summary

This deliverable document constitutes an accompanying document to the first prototype for the Smart Object Integration in the ADVENTURE platform. With this first prototype, a solution has been developed and realized, which provides bi-directional data exchange between wireless sensor nodes and the ADVENTURE system employing different communication paradigms and hardware setups.