2013 IEEE Seventh International Symposium on Service-Oriented System Engineering

Enabling Cloud-connectivity for Mobile Internet of Things Applications

Pablo Puñal Pereira, Jens Eliasson, Rumen Kyusakov,
Jerker Delsing and Asma Raayatinezhad
Department of Computer Science, Electrical and Space Engineering
Luleå University of Technology
Luleå, Sweden.
Email: pablo.punal@ltu.se

Mia Johansson
Säktek Tillväxt Skellefteå
Skellefteå, Sweden
Email: mia.johansson@saktek.eu

***Abstract*—The number of small embedded devices connected to the Internet is increasing. This growth is mostly due to the large number of Internet of Things (IoT) deployments, with applications such as: industrial monitoring, home automation, and others. One common aspect with the majority of application areas is the lack of mobility. Most IoT devices are stationary and often use IEEE 802.15.4/6LoWPAN solutions. When a high level of mobility is required, the use of IEEE 802.15.4 is not possible without adding additional hardware for the user to carry.**

In this article, a holistic network architecture consisting of heterogeneous devices is presented. The architecture is composed of Embedded Internet Systems (EIS) and uses standard communication protocols. One important feature is the use of the Service-oriented architecture (SOA) paradigm. The use of SOA, by utilization of the CoAP protocol and standard services, enables the proposed architecture to exchange sensor- and actuator data with an Internet-based cloud as well as a user's local cloud consisting of sensor IoT devices, smart phones and laptops. Another component of the architecture is a web-based human-machine interface for configuration, monitoring and visualization of sensor and actuator data using emerging web technologies for structured data processing.

Results from experiments and real-world tests show that the proposed architecture can support sample rates of up to several kHz while enabling sensor data to be transmitted to SOA services in real time. This proves that the use of SOA, and RESTful web services in particular, is feasible on resource-constrained platforms while supporting true mobility.

***Keywords*-SOA, Mobile, CoAP, Internet of Things, Low power, Platform, PAN, Bluetooth, Cloud.**

I. Introduction

In the last few years, mobile devices have become more and more pervasive, and by the growth of the technology, their processing speed has increased resulting in the ability to develop more advanced mobile applications. Billions of smart devices will soon be connected to the Internet to form the Internet of Things (IoT) [1]. In the last years, the number of devices using technologies with support for IP (Internet Protocol) has increased.

With the increase of the number of connected embedded devices it becomes particularly important to enable intuitive and simplistic human-machine interface that allows the users to interact with these devices. This interface must allow the deployment of various IoT services for enhancing the quality of life, safety and productivity of mobile users in a number of application areas. Moreover, the users should not be required to learn new interaction models for every device or vendor, hence the interface should be based on standards that are not proprietary and are widely adopted.

One application area requiring mobile and robust sensor solution is the security field, specifically the personal security field. Combining and analyzing data from sensors in wearable electronics, cameras or sound collectors with secure communication is essential to technical solutions where the individuals security is in focus. An example of such a solution is a prototype developed by the Säktek cluster [2] of security and technology, where a combination of mobile sensor data is analyzed to create alarms that are transferred to operators using a secure mobile platform for further actions. A sophisticated network of devices ensures a highly mobile and robust solution which is essential for maintaining the individuals freedom while being monitored for specific purposes.

For mobile sensor applications, the following properties and requirements are important: devices must be of small size to not interfere with the user's daily life, communicate with standard protocols to allow maximum interpperability so that standard consumer devices can be used, and be low power so that the battery lifetime is extended. Example applications include: health care and elderly care [3], security (policemen, firemen etc.), industrial application, as shown by Karnouskos et al. [4], vehicle testing [5] and winter road conditioning [6]. Typical requirements for these types of applications ranges from a few data transmissions per hour (posture, position, temperature etc) to continuues data transmission with sample rates of several kHz (vibration, audio, CAN-bus, etc). Some of these applications could transmit sensitive information, therefore, the security is a really important requirement. In this paper, we discuss the security, but it is only part of the future work.

However, an often seen aspect is that proprietary communication protocols are used in many mobile applications. The use of non-standard communication protocols severely limits interoperability and makes it difficult to create new

978-0-7695-4944-6/12 $26.00 © 2012 IEEE
DOI 10.1109/SOSE.2013.33

systems and services using existing hardware and software. A better solution would be to enhance the Internet of Things paradigm by having Internet-connected devices communicating using the Service-oriented architecture (SOA) approach. This would enable creation of Systems of systems and formation of a personal mobile sensor cloud where sensor- and actuator nodes can cooperate directly with human users and standard consumer devices, such as smart phones and tablets, as well as the global Internet.

The paper is outlined as follows: Section II presents related work followed by enabling technologies - Section III, used for the proposed architecture. Sections IV and V present the architecture for mobile sensor clouds and the experimental setup, respectively. The results are presented in Section VI. Section VII outlines future work and the paper is concluded in Section VIII.

II. Related Work

This section focuses on analyzing the state of the art of different technologies and research areas used in this paper.

1) Internet of Things: The Internet of Things (IoT) is a concept of communication between people and smart objects, e.g. mobile phones and sensors, and communication among Internet connected devices. A number of applications rely on enabling the objects in everyday living environment to communicate with each other in order to exchange the information they have collected from their surroundings. One of the valuable applications is in transportation to use the traffic information for traffic control purposes. Healthcare is another example where IoT technology can be applied as well as disaster alerting or recovery in work environments such as mines [7].

2) Mobile Cloud: Mobile phones have become a crucial part of people lives because of their computation and communication capabilities. While they suffer from deficiencies in performance (e.g. battery life, storage, and bandwidth) and in security (e.g. reliability and privacy) [8], they are programmable and come with a variety of embedded sensors, such as microphone, camera, GPS, digital compass, gyroscope, and accelerometer. This collection of sensors enables applications in wide range of domains, such as healthcare, social networking, transportation, environmental monitoring, business, safety [9]. The data obtained from the sensors can be sent over the Internet to *clouds* where the data storage and data processing are performed. This concept can also be used to provide more computational capabilities for resource constrained IoT devices. An example of that is the *thin server architecture* proposed by Kovatsch et al. [10] where even low-level firmware functionality is executing in the cloud. The *mobile cloud computing* helps overcome the obstacles and deficiencies of the mobile phones, and brings a broad range of new services and facilities [8].

3) Bluetooth PANs: Bluetooth is a specification for the use of low-power radio communications to connect wirelessly wide range of mobile electronic devices such as mobile phones, PCs, laptops, media players, tablets, ect. The distance range is not high, around 10 meters, but nevertheless this is sufficient to create Personal Area Networks (PANs). The relatively short range helps to increase the security as it is very difficult to eavesdrop the network.

Bluetooth is a mature and reliable technology with large number of development tools and extensive studies and information in the literature about Bluetooth and the PAN Profile [11].

4) Wearable Electronics: Novel techniques on miniaturization in the semiconductor's world and the recent investigations on new materials with electromagnetic properties have opened up new ways to understand the electronic design.

The wearable electronics with a low power design could be a powerful tool to measure data on a human body with a low level of body-intrusion. Already now, the current technology allows to create a pseudo-complex wired(less) sensor network.

One of the most important things in wearable electronics is to avoid extensive use of wires. This is the main motivation of studies in wearable antennas [12] that aim to increase the performance with some extra features like more flexibility, waterproofness etc.

When it comes to sending data to the users there are different approaches that can be used - if the application requires visualization of some data it is possible to use a display directly on the clothes [13] or use special glasses (like Google Glass). The screens on the clothes can get more technical failures, but are more comfortable and less intrusive than the glasses. Another recent studies are focused on haptic displays using dielectric elastomers to recreate surfaces directly on fingers [14].

The wearable sensors are an important tool to get data form the human body with the least level of intrusion. As an example, it is possible to take measurements of the level of oxygen in the blood, electrocardiograms, breath rate, temperature, blood pressure, skin humidity, level of stress etc.

III. Enabling Technologies

In this section, a number of enabling technologies are presented together with their characteristics.

A. Wireless communication

A mobile (sensor) network, or Personal Area Network (PAN), is usually based on general-purpose technologies and protocols, and has higher processing capabilities than an average IoT or WSN device. PAN devices are more often used on, or in the vicinity of, human users.

Today, two of the most widely used radio technologies for personal networks found in mobile devices are WiFi and Bluetooth. WiFi is becoming more and more low power, but still cannot compare with Bluetooth. Especially the relatively new standard of Bluetooth 4.0 with the extension of Bluetooth Low Energy (BLE). Bluetooth Low Energy uses only a fraction of the power compared to Bluetooth 2.0. Even though IEEE 802.15.4 with 6LoWPAN is widely used for IoT applications, the technology is better suited for stationary monitoring as todays mobile phones are not supporting it. Therefore, the only way of using IEEE 802.15.4 with IPv6 in PAN is by using an additional gateway. This approach is feasible in some applications, but should be avoided when minimum size and low cost is important. In some simple cases, each sensor node can feature its own GPRS modem for easy Internet-connectivity, but this is a very costly solution. By using Bluetooth, no separate gateway is needed, and the user is not forced to have several SIM cards.

Bluetooth-equipped networked sensor nodes can achieve good interoperability with consumer devices, have lower power consumption than WiFi, and have a lower cost. Bluetooth is also by far the most widespread technology supported by existing consumer devices, which further makes it an interesting technology to use for Personal Area (Sensor) Networks. A sensor network composed of Bluetooth-equipped EIS devices used in the context of sensor networks is called a Bluetooth Sensor Network (BSN). Regarding the communication security, Lindell et al. [15] showed that Bluetooth 2.1 can provide a good level of security. By using Bluetooth and the standard PAN profile, virtually any Bluetooth-enabled mobile phone can act as a gateway to the Internet.

B. CoAP (Constrained Application Protocol)

The IETF Constrained Application Protocol is an application-layer protocol designed to provide web services working with constrained nodes - devices using microcontrollers with small amounts of ROM and RAM, running 6LoWPAN network stacks with high packet error rates etc. The protocol is designed for low-power networking allowing the nodes to switch to sleep mode to extend their battery life.

CoAP provides a request/response interaction model between application end-points, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs, RESTful interaction, extensible header options, ect. CoAP easily interfaces with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead and simplicity for constrained environments. CoAP runs over UDP unlike HTTP.

Some features of CoAP are:

- Two types of request messages: Confirmable Message (CON) - the message is retransmitted (maximum four times) with an exponential timeout waiting for an Acknowledged Message (ACK) or the correct response from the server. The second type is the Non-Confirmable Message (NON) - the message is sent without any kind of response.
- The URI format allows the use of standard and specialized service endpoints. One such example is the resource discovery defined in RFC 5785 that uses the */.well-known/core* path and the CoRE Link Format.
- CoAP also allows to send very big messages with a stop-and-wait mechanism called ”blockwise transfers” (splitting messages).

C. Embedded Web Technologies

The user interactions in our platform are based on the web architecture. Key technologies to enable embedded RESTful web services, web linking and data representation are CoAP and Efficient XML Interchange (EXI) protocol for binary representation of XML structured information. The EXI format has W3C recommendation status [16] and is designed to increase the compactness and processing efficiency of XML data while keeping the compatibility with the XML Infoset.

By using EXI, XHTML visualization of the sensor data can be efficiently transmitted and processed and hence allowing the use of standard web technologies to interact with the sensor nodes through the user's mobile phone. The envisioned support in the mobile browsers of the IETF CoRE technologies such as CoAP, Observe, Blockwise transfers, CoRE Link Format; will enable the use of asynchronous RESTful client-server applications hosted on the sensor nodes.

IV. Mobile Cloud IoT Platform

The architecture of the proposed IoT platform consists of the following components:

A. Network Connectivity for Mobile Devices

The devices part of the network architecture are Bluetooth-based sensor nodes and a standard mobile phone acting as an access point to the Internet. In our experiments we used the Mulle sensor platform shown in Figure 2 developed by Eistec AB [17]. One Mulle is used as an IP-router that is started by initiating a Bluetooth connection towards the mobile phone's access point service. When the router Mulle has established a Bluetooth PAN connection and acquired an IP-address using a DHCP client, it starts its own access point service. Other devices (Mulles, PDAs etc) can now connect to the router Mulle's PAN-NAP profile, using the PAN-U on the clients. When a client Mulle has an established connection, DHCP is used on both router and sensor Mulles in order to distribute IP addresses. The router Mulle also features a NAT (Network Address Translation) service, allowing up to seven sensors (clients) to share one

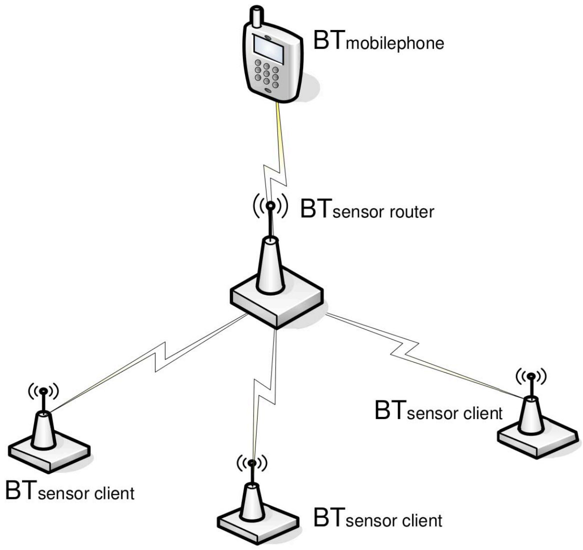

Figure 1. Bluetooth sensor network

Internet-connection at the same time. As a result, a sensor network consisting of one mobile phone, one router Mulle and up to seven client Mulles can be formed. Mulles either use the NTP protocol or CoAP-based time synchronization for security reasons and to correctly timestamp sensor data.

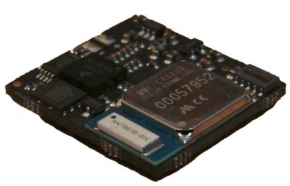

Figure 2. Mulle v3.2

The Mulle platform has low power consumption and its large number of I/Os make it a suitable component in mobile personal sensor networks. The Bluetooth version is capable of communicating with virtually any Bluetooth-enabled devices, e.g. computers, PDAs and mobile phones, using only standard Bluetooth protocols and profiles. The inherent support for TCP/IP, by the lwIP stack [18], enables the Mulle to transmit sensor data directly to the Internet without proprietary gateways or middleware services.

B. Application Services

The REST engine of our platform is based on a modified version of *libcoap* implementation first presented in [19]. The CoAP implementation covers Draft IEFT Core CoAP 11 [20] therefore the REST Engine provides functions to initialize and configure a RESTful Web service resources according to the layered model presented in Figure 3.

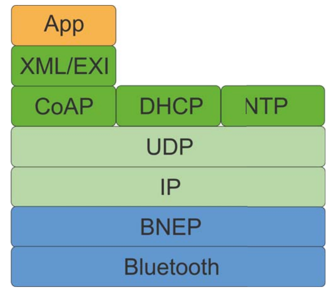

Figure 3. Mulle communication stack

The REST engine offers three different types of RESTful resources:

- Resource: A basic REST resource should have an URI-path, allowed methods, and a string for the Web Linking information. For every resource, the application must provide a resource handler function, which receives the request and generates the corresponding response. Both messages are accessed through the REST Engine API. This resources could be masked/hidden depending on the client or even have different options.
- Event Resource: This type requires a second handler function to be implemented by the application developer. A client-defined event triggers this handler, and it generates a PUT message with the content that would like to change.
- Timed Resource: Additional to the signature of the basic resource, it is possible to define a time interval or an specific time/date when the REST Engine periodically calls a second handler function similar to the one for event.

C. Cloud Integration

Integration of server applications, mobile phones and sensors is based on the SOA approach. A server has the ability to communicate with mobile devices or furthermore connect to each sensor directly. Each sensor node, a Mulle v3.1, is connected to a mobile device and communicates with the server via the Internet provided by the mobile device (as shown in Figure 4. The server is composed of several services which can be integrated in mobile phones or can be distributed on cloud infrastructure. Additional services, such as data filtering, alarm management and others, can be implemented on the sensor nodes as well. Some services that are important for the proposed mobile platform are listed below:

1) Configuration service - it is used by all the devices to get a configuration parameters such as the IP and port addresses of all other services.
2) Time service - it is used to synchronize the devices.
3) Proxy service - facilitates the communication with the sensors and other mobile devices.

4) Filter service - the data is analyzed and filtered.
5) Alarm service - based on certain rules it warns the user for possible critical conditions through a SMS, email or other mechanisms.
6) Historian service - collects the data and stores it.

The server has the ability to communicate with the devices in different types of formats such as text, binary, XML, and EXI. The advantage of the text format is its convenience when reading, but it is hard to parse. Binary is efficient to communicate, but requires external tools to make it readable by users. XML is understandable and very well structured, but the size of its messages is big and it is much worse to parse compared to binary formats. EXI has smaller size and it is very efficient to process and easy to transfer.

The proposed SOA architecture can be deployed on three different network levels, or a combination of the three levels. Each level provides its own set of requirements, restrictions and performance.

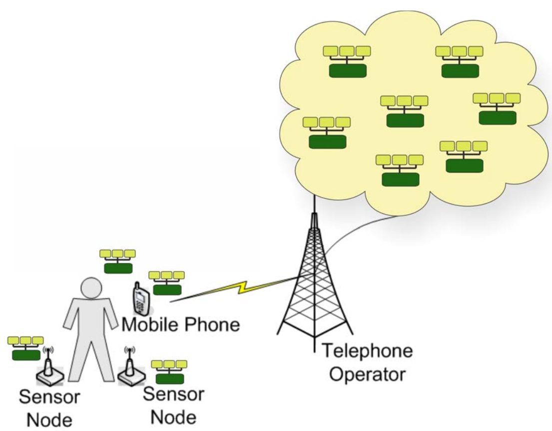

Figure 4. Mobile SOA-enabled IoT architecture

The levels are:

1) Node: The first level is the local mobile network of sensor- and actuator platforms. Deployed services on this level allows for maximum performance since no data need to be transmitted to the Internet and back just for two nodes to communicate. Instead services can be invoked with a minimum of overhead. Deploying services on node/network level will also allow the mobile cloud to function where no Internet connectivity exists, such as in very rural areas or in mines for example.

2) Network: The second level is to rely on a SOA-enabled gateway. This approach has the advantages that a mobile phone or a typical sensor network gateway [21] has drastically more processing power, memory and bandwidth. Disadvantages are that not any mobile phone can be used since the phone must support TCP/IP and SOA.

3) Internet: The third level is the global Internet. By deploying services on web servers or even data centers, all limitations of processing power are mitigated. However, more data must be sent and received by the sensor- and actuator nodes since much of the control logic has been moved out of the local mobile cloud. Integration of Internet services and sensor node is currently an important research field, see for example [22].

Since a user might want to be able to configure his or her network in an optimal way, a combination of the three levels might be desirable. By dynamically utilize services at different levels, an optimum on performance and power consumption can be achieved. For example, several sensors can cooperate using level 1 in order to detect an anomaly. When a deviation from the normal is detected, one or several services can be utilized on the gateway level, thereby allow services with relatively high memory and processing power to be utilized. Finally, when alarms are to be generated, or other systems and services could be invoked, the third level could be used to offload node and gateway devices.

D. Human Machine Interaction

Another component of the proposed IoT platform is a standard-based human machine interface that supports visual input and output to small embedded nodes.

The advancements of micro electronics in the last decades have led to lowering the cost and size of computing hardware to levels that allow the integration of embedded devices into everyday objects and activities a.k.a. ubiquitous computing. The first developments in this area can be traced back to the work of Mark Weiser in the late 80s and early 90s [23] who stressed the importance of intuitive interface between the humans and the computing devices. The web technologies have since enabled the fusion of digital and real worlds by providing a simple yet intuitive platform to interact with cloud services and virtual reality platforms. While the development of more advanced techniques to interact with IoT devices are available and will play important role in the future [24], our focus is on enabling the web technologies on IoT devices. By using the touch displays available in the current mobile phones, the mobile users can interact with the sensor and actuator devices in a intuitive and standard-based manner.

E. Communication Security

Our focus in this work is on analyzing the different security levels that could be implemented in our system while taking into account the energy consumption and performance of different solutions. Currently, we identified a set of possible technologies on different layers of the proposed design (see Table I). As usual, a higher level of security requires more CPU cycles, and hence increased cost and latency, and a bigger packet size, which directly involves a greater expenditure of energy, and could be critical on mobile devices as this shortens the battery life of the devices.

Layer	Possible Security
Application	CoAP encryption
UDP	None
IP	IPsec
Bluetooth 2.0	SAFER+

Table I
SECURITY LAYER MODEL

The level of security depends on two distinct types of solutions because there are two kinds of connections. In a Point-to-Point (P2P) connection the communication is among nodes of the same PAN and here it is possible to use the CoAP ncryption and SAFER+ for Bluetooth. This connection is used also to connect all the nodes of a PAN to the router. In a Point-to-Internet (P2I) connection the communication is between the router of the PAN and the server and in this case it is possible to use the same security modes but there is another possibility i.e. IPsec. The security on the CoAP layer has 3 modes (Security section of IEFT CoAP 11 [20]):

- PreSharedKey: DTLS is enabled and there is a list of pre-shared keys and each key includes a list of which nodes it can be used to communicate. At the extreme there may be one key for each node that a particular CoAP node needs to communicate with (1:1 node/key ratio).
- DTLS is enabled and the device has a raw public key certificate that is validated using an out-of-band mechanism. The device also has an identity calculated from the public key and a list of identities of the nodes it can communicate with.
- Certificate: DTLS is enabled and the device has an asymmetric key pair with an X.509 certificate that binds it to its Authority Name and is signed by some common trust root. The device also has a list of root trust anchors that can be used for validating a certificate.

V. EXPERIMENTAL SETUP

The constrained REST engine implementation used in this work is based on the IETF CoAP [20] draft. The UDP layer on Mulle sensor nodes is callback-based (RAW API [25]). This does not allow the use of sockets and the specific socket functions. During the programming design step, we focused on maximum performance of the code to get the maximum rate of packets per second as well as low-power features.

The Mulle software has a full set of control functions to manage the parameters of the UDP connections (with optimized power-consumption). In order to manage all the incoming and outgoing messages, the libcoap queuing mechanism for CoAP packets was extended to store the data of the connection including IP addresses, ports, number of retransmissions, etc.

The incentive for this design of the Mulle nodes software is to enable the sensor to communicate its data to a server. In this scenario the software has to be able to manage all the connections (low number if possible) and create new packets (take data from the sensors, analyze the data and compress several samples in a new packet). To do that, the sensor nodes have to use events (interrupts), timers and callbacks. Specifically for our use cases, any sensor is able to dispatch 6 CoAP packets (Non Confirmable Messages) every 250 ms and check the incoming queue for new messages.

As with all constrained embedded devices, the memory consumed is one of the biggest problem especially on devices with network connectivity where the connection state needs to be stored. In CoAP the maximum packet size is defined as 1.4 kB and all the incoming messages need to be saved in memory together with the conformable messages. This can be a huge problem in some applications requiring the traffic to be routed through a busy proxy node. However, this is not a problem on the final sensor nodes because they must send data primarily with Non Confirmable Messages (NON). To prevent overflows during the packet saving step, the system should have a limit on the number of packets that can be saved at the same time on the Mulle's packet queues, defining two types of limits: an outgoing limit for Confirmable Messages (COM) and an incoming limit for the received messages. One of the powerful features of CoAP is the packet-retransmission of Confirmable Packets (see section III-B), but this could create problems on the receiver because it might store the same packet more than one time. To prevent multiple allocations in memory of the same data the system has to check (only) the previous packet ID and save it only if this ID is not already in memory (it is very important do this step exactly when an incoming message is received, not when the content of the packet is analyzed).

In this system, one node can receive requests from a server or from another node. As the response generation could take a long time, it is required that separate responses are used to prevent unnecessary request retransmissions or even a request timeout. This method is based on sending an empty ACK when a packet is received and then, when the answer is ready, sending a CON message.

The communication between the server and client applications is facilitated by the use of services, as time services, log, security, etc. According to the idea of dynamic services, the clients are able to send a request to the server about a list of available services and use any of them at run-time.

The advantages of this design are the possibility to do updates (the system allows to add new services to the server without modifications on the clients) and the intuitive mapping of the service endpoints to the server resources by URLs.

VI. RESULTS

To evaluate our design, we created a test to measure the memory footprint and the quality of our design between a server and a Mulle sensor node. For this test we used a PC with Ubuntu 12.04 as a server running our version of CoAP server and another tools like echo programs and network analyzers. We used an Android 2.3.7 mobile phone to create the Bluetooth PAN which was connected to the Internet by a wireless LAN. It is important to note that in this process we did not use any additional software on the phone. The software of the Mulle nodes has been compiled with the m32c-elf-gcc (GCC) version 4.4.3. For the performance tests on the transmission and reception we measured the latency for 10.000 messages (samples) for different payloads between 0 and 256 bytes.

A. Memory Footprint

	Total Memory Usage(kB)
System Base	17.3
Reserved Memory for Packets	6.1
CoAP core	4.9
CoAP dispatcher	0.9
CoAP packet emission	0.1
CoAP packet reception	0.2

Table II
MEMORY USAGE BY FUNCTIONALITIES

The memory footprint is mostly of interest on the Mulle devices in our mobile cloud design, because the other devices has much more resources that are needed. Table II shows a detailed memory footprint of our implementation of CoAP for Mulle sensor nodes.

The system requires 29.5 kB of RAM memory, the reserved memory for new incoming/outgoing packets is 6.1 kB, but it could be bigger if the applications requires it.

B. Transmission Performance

The nodes (Mulles) have been designed mainly to send data to the server or other systems, thereby the sending rate is a very important feature. Figure 5 shows the result of sending multiple messages with different size of the payload (10.000 packets per payload size), as expected the latency is increasing with the payload size.

C. Reception Performance

Another interesting thing is the incoming packet rate. In this case the type of the incoming CoAP packet has to be ignored to disregard the time of package management. Figure 6 represents the time for CoAP packet reception (including the time for critical package testing) versus the payload of these packets. If the latency of the incoming and outgoing packets is compared, one can observe that the reception is much faster than transmission. This is a good feature that prevents packet loss during the communication between two identical systems. This asserts that the communication should be stable and should not create blockages.

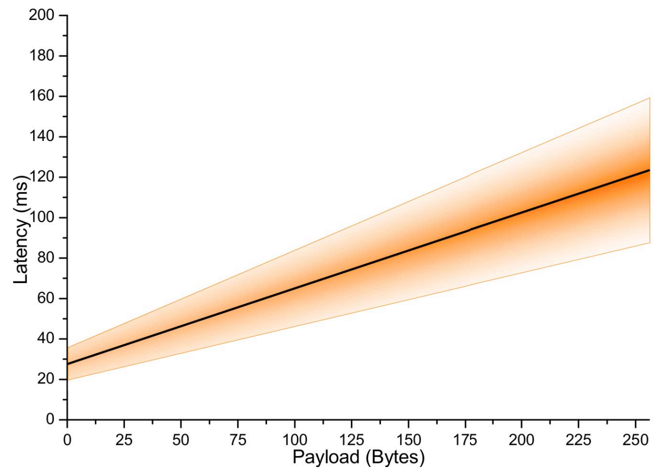

Figure 5. Test of Transmission

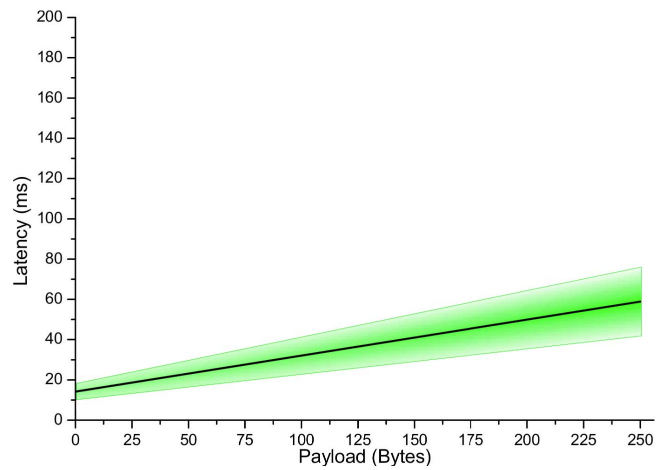

Figure 6. Test of Reception

D. Configuration and Data Visualization

In order to validate our approach of visualizing and interacting with sensor and actuator devices by using standard web technologies we performed some preliminary tests. As discussed earlier our main enabling technology in this area is EXI that allows us to compress and process more efficiently XHTML input and output on embedded nodes with limited memory and computing capabilities. By using standard web technologies the human interactions follow a familiar and intuitive pattern already used in the world wide web and thus the learning curve for the users is not steep. Moreover, the use of embedded XHTML/EXI servers on the nodes eliminates the need of special software installed on the users' mobile phones - it is sufficient to have an Internet browser.

A typical use of our approach is to visualize sensor data and control the embedded nodes behavior. A sample XHTML that supports this functionality is given in Figure

7 rendered by smart phone browser.

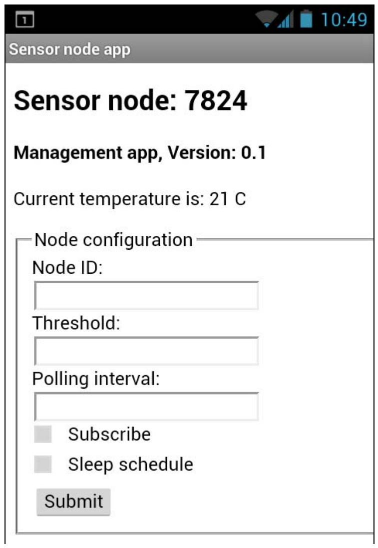

Figure 7. Example XHTML that can be used to visualize and control sensor nodes

The size of this XHTML when represented in text form is 1137 bytes. By using EXI representation the same document takes only 390 bytes - that is 34.3 % of the original size. Moreover, the processing efficiency is also improved and hence the requirements on RAM and CPU are less severe. While the suggested XHTML/EXI embedded server component has not been implemented in the present work, the achieved compactness suggests that in many scenarios the use of EXI-based web technologies will be beneficial. Future implementations can also use an optimized XML schema, instead of the generic XHTML schema used in this example. An optimized schema can better describe the concrete XHTML document to achieve even higher efficiency in terms of compactness and processing as compared to generic one.

VII. Future Work

For some applications, such as process monitoring, health care, and vehicle testing, a reliable stream of sensor data is important. This is especially true for high-frequency data such as vibration data, audio signals, CAN-bus data, etc. In order to fully support these types of applications, the system should be enhanced with support for Quality of Service (QoS).

Another important feature is dynamic configuration and re-configuration of sensors. Using event-based communication with the support of an advanced rule-based scripting-language would enable users to create filters and rules for events that the sensors should detect.

Another important issue is security. It is a very important feature for data streams sent over public networks such as the Internet. In the proposed architecture security has two different levels; the first level is the local wireless network (using Bluetooth) and the second one is the Internet. Currently, the Mulle platform does not support security on Bluetooth, IP, or UDP layers except for simple PIN-code pairing, but as shown in Table I, the possible security features of these layers are enough to get a good level of security on the complete system. Implementing security mechanism is however outside of the scope of this paper.

VIII. Conclusion

In this paper, a new architecture for mobile cloud sensor applications based on the Internet of Things approach is presented. The architecture consists of Bluetooth-based low-power sensor nodes communicating using standardized protocols and profiles. The use of a user's mobile phone and the mobile telephone access network allows true mobility. Furthermore, the use of CoAP and the Service-oriented architecture (SOA) paradigm allows a multitude of services and features to be accessible from the user's local sensor cloud as well as on the global Internet. In this architecture, services executing on sensor nodes, the user's mobile phone or pad, or even on the Internet, can collaborate in order to exchange data and perform distributed processing.

Performed experiments and tests show that the proposed solution is a viable solution for standards-based high performance sensor and actuator platforms. The proposed architecture can be used in applications that require a high level of mobility with requirements on small size, low power consumption yet the need to be able to communicate with standard consumer products and services. Two application areas; personal safety and security and vehicle testing, which both require a very high level of mobility is presented as use-cases, and the results are mapped toward these cases.

Acknowledgment

The authors would like to thank for their support the European Commission, and the partners of the EU FP7 projects IMC-AESOP (www.imc-aesop.eu) and I2Mine (http://www.i2mine.eu) for the fruitful discussions. We would also like to express our gratitude towards the CASTT project at Luleå University of Technology, and the Säktek organization.

References

[1] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.

[2] "Säktek," http://www.saktek.eu/, Oct 2012.

[3] D. Hoang and L. Chen, "Mobile cloud for assistive healthcare (mocash)," in *Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific*, dec. 2010, pp. 325 –332.

[4] S. Karnouskos, A. W. Colombo, T. Bangemann, K. Manninen, R. Camp, M. Tilly, P. Stluka, F. Jammes, J. Delsing, and J. Eliasson, "A SOA-based architecture for empowering future collaborative cloud-basedindustrial automation," in *38th Annual Conference of the IEEE Industrial Electronics Society(IECON 2012), Montréal, Canada.*, 2012. [Online]. Available: http://diktio.dyndns.org/files/2012_IECON_architecture.pdf

[5] T. ElBatt, C. Saraydar, M. Ames, and T. Talty, "Potential for intra-vehicle wireless automotive sensor networks," in *Sarnoff Symposium, 2006 IEEE*, march 2006, pp. 1 –4.

[6] S. M. Casselgren, J. and J. Leblanc, "Model-based winter road classification," *International Journal of Vehicle Systems Modelling and Testing*, vol. 7, pp. 268–284, 2012.

[7] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.05.010

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*. [Online]. Available: http://dx.doi.org/10.1002/wcm.1203

[9] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Comm. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2010.5560598

[10] M. Kovatsch, S. Mayer, and B. Ostermaier, "Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things," in *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012), Palermo, Italy*, 2012.

[11] R. A. Rashid and R. Yusoff, "Bluetooth performance analysis in personal area network (pan)," 14 September 2006.

[12] S. Kim, M. M. Tentzeris, and S. Nikolaou, "Wearable biomonitoring monopole antennas using inkjet printed electromagnetic band gap structures," 2011.

[13] C. Zysset, N. Mnzenrieder, T. Kinkeldei, and G. Trster, "Woven active-matrix display," *IEEE TRANSACTIONS ON ELECTRON DEVICES*, vol. 59, no. 3, pp. 721 – 728, March 2012.

[14] I. Koo, K. Jung, J. Koo, and H. R. Choi, "Wearable fingertip tactile display," 18 October 2006.

[15] A. Y. Lindell, "Comparison-Based Key Exchange and the Security of the Numeric Comparison Mode in Bluetooth v2.1," in *Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*, ser. CT-RSA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 66–83.

[16] *Efficient XML Interchange (EXI) Format 1.0*, W3C Std., March 2011. [Online]. Available: http://www.w3.org/TR/2011/REC-exi-20110310/

[17] "Eistec AB," http://www.eistec.se/, Oct 2012.

[18] A. Dunkels, T. Voigt, and J. Alonso, "Making TCP/IP Viable for Wireless Sensor Networks," in *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN 2004), work-in-progress session*, Berlin, Germany, Jan. 2004. [Online]. Available: http://www.sics.se/ adam/ewsn2004.pdf

[19] K. Kuladinithi, O. Bergmann, T. Pötsch, M. Becker, and C. Görg, "Implementation of coap and its application in transport logistics," *Proc. IP+ SN, Chicago, IL, USA*, 2011.

[20] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, *draft-ietf-core-coap-11*, CoRE Working Group Internet-Draft, 16 July 2012.

[21] W. Wang, Y.-X. Zou, G. Shi, and Y. Zhu, "A web service based gateway architecture for wireless sensor networks," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 02, feb. 2009, pp. 1160 –1163.

[22] F. Delicato, P. Pires, L. Pinnez, L. Fernando, and L. da Costa, "A flexible web service based architecture for wireless sensor networks," in *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, may 2003, pp. 730 – 735.

[23] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[24] M. Kranz, P. Holleis, and A. Schmidt, "Embedded interaction: Interacting with the internet of things," *Internet Computing, IEEE*, vol. 14, no. 2, pp. 46 –53, march-april 2010.

[25] A. Dunkels, "Full TCP/IP for 8 Bit Architectures," in *Proceedings of the First ACM/Usenix International Conference on Mobile Systems, Applications and Services (MobiSys 2003)*, San Francisco, May 2003. [Online]. Available: http://dunkels.com/adam/mobisys2003.pdf