Demonstration of OpenMTC – M2M Solutions for Smart Cities and the Internet of Things

Sebastian Wahle
Fraunhofer FOKUS
Berlin, Germany
sebastian.wahle@fokus.fraunhofer.de

Thomas Magedanz
Technische Universität Berlin
Berlin, Germany
thomas.magedanz@tu-berlin.de

Frank Schulze
Fraunhofer FOKUS
Berlin, Germany
frank.schulze@fokus.fraunhofer.de

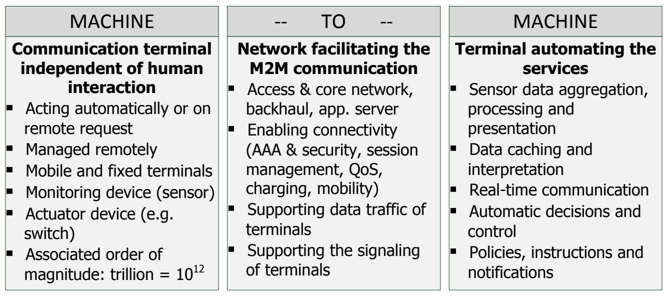
***Abstract*—This extended abstract covers the setup of our demonstration, giving a practical view on our research and development activities in the field of Machine-to-Machine (M2M) communication or machine type communication (MTC). M2M communication is a paradigm in which end-to-end communication is executed without human intervention connecting non-IT objects to an IT infrastructure. Our demo aims at enabling the audience to use different client devices (e.g. smart phone, tablet PC, notebook) to access our M2M applications and control sensors, actuators, and devices (e.g. lamp, fan). Also, the visitor can feed the M2M system with policies to trigger automated sequences of actions and thereby steer and control the M2M communication that is performed without human intervention.**

I. Introduction

This demo has been described and published earlier in [1]. Todays communication networks such as the mobile operator networks have been designed and build to fit the needs of human communication. At the same time, M2M communication solutions have emerged that are available on the market today. However, such solutions are mostly monolithic infrastructures that do not interoperate. This has been identified as a major show stopper for open, customizable, and ubiquitous M2M system deployments and operation.

Our experiments with state-of-the-art operator network architectures (like IMS [2]) have shown that such do not suffice the needs and specific characteristics of large scale machine type communication scenarios. With maturing M2M standardization expected in 2012 and 2013, the aim of our OpenMTC platform is to provide a standard compliant middleware platform for M2M oriented applications and services. While supporting application domain driven scenarios such as eHealth and Smart City services, OpenMTC inherits advanced networking capabilities provided by our highly successful 3GPP Evolved Packet Core (EPC) [3] implementation.

Although the demonstration shows scenarios for a specific application domain (Smart Home), we have to emphasize that our architecture aims to be generic and abstracts from any vertical application domain specifics. Our R&D activities are focused on the communication mechanisms to support different M2M verticals in terms of a common platform. Figure 1 and 2 show this scope. The remainder of this document will detail the demonstration setup from a technical perspective and will outline three different communication scenarios that are demonstrated to the audience.



MACHINE	-- TO --	MACHINE
Communication terminal independent of human interaction	**Network facilitating the M2M communication**	**Terminal automating the services**
▪ Acting automatically or on remote request ▪ Managed remotely ▪ Mobile and fixed terminals ▪ Monitoring device (sensor) ▪ Actuator device (e.g. switch) ▪ Associated order of magnitude: trillion = 10^{12}	▪ Access & core network, backhaul, app. server ▪ Enabling connectivity (AAA & security, session management, QoS, charging, mobility) ▪ Supporting data traffic of terminals ▪ Supporting the signaling of terminals	▪ Sensor data aggregation, processing and presentation ▪ Data caching and interpretation ▪ Real-time communication ▪ Automatic decisions and control ▪ Policies, instructions and notifications

Fig. 1. M2M communication

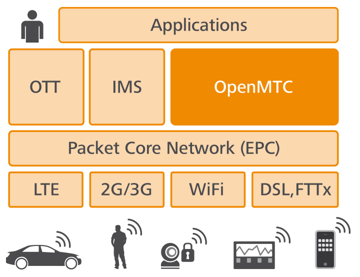


Fig. 2. The OpenMTC framework

II. The Demo Setup

Figure 3 shows the abstract demo setup. We control 2 devices: a lamp and a fan. Also, we use a multi-sensor that can measure temperature, humidity, and brightness.

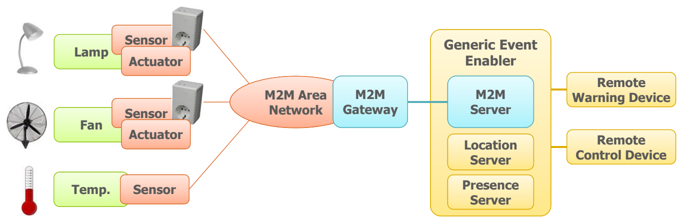


Fig. 3. Abstract demo setup

The devices are controlled using power plugs that measure different electrical power consumption values and also allow for remote switching. The plugs connect to the rest of the infrastructure by means of an M2M area network. All elements of the setup are briefly summarized below:

Sensors	Devices which receive information from the physical environment and transmit it to the M2M gateway.
Actuators	Devices which receive remote commands from the M2M gateway to modify the physical environment.
Area Network	Local sensor network which terminates with an M2M gateway towards the operator network.
M2M Gateway	Device terminating the sensor gateway towards the telco gateway. Executes protocol translation from sensor messages to IP-based messages and receives commands from warning devices.
M2M Server	Exchanges information with M2M gateway. Contains the generic event enabler mashing location and presence with sensor information. Transmits notifications towards warning devices.
Warning Device	Client receiving warnings from the sensor environment and making commands for ameliorating specific situations.

Figure 4 shows the demo setup from a more technical point of view allowing some insight on how the different elements are interconnected.

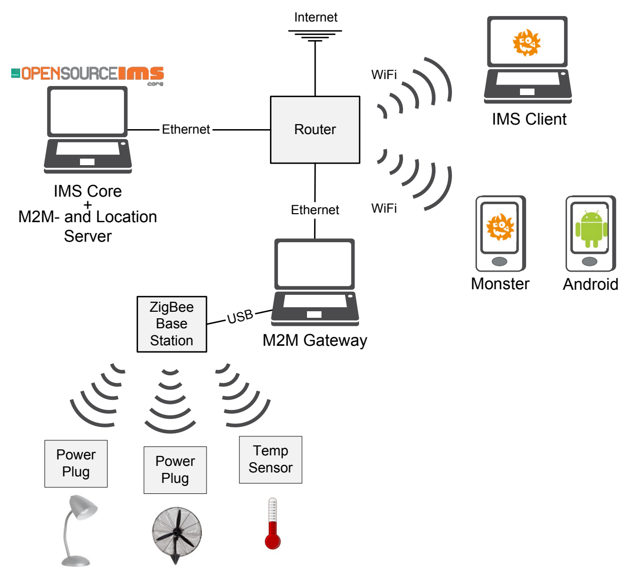


Fig. 4. Demo elements overview and interconnection

The power plugs connect to a ZigBee base station which plugs into our M2M gateway via USB. The gateway could run on standard desktop class PCs or smaller setups such as embedded systems depending on the deployment scenario. For our purposes we use a standard off-the-shelf notebook where the USB ZigBee bases station can be plugged and which connects to the M2M server via its Ethernet interface. The M2M server hosts our open source IMS Core [2] and the M2M server capabilities. Via the router, an Internet connection is available to all connected elements. A variety of client devices wirelessly connect to the setup running our IMS client implementation [4] and our Android smart home application.

III. Demonstration Scenarios

A. First Scenario: Actuating the Environment

In a first step, the user browses available devices at home using his client device and application (our IMS client). The client enables the user to check upon available parameters and actions supported by any available device. For the first demo scenario, the user then decides to power on/off one of the devices (e.g. the lamp). Here, the location of the user is irrelevant as long as he can connect to the M2M server. Therefore, his devices at home can be controlled while he is at home or away.

B. Second Scenario: Sensor Remote Notification

For the second scenario the lamp will be powered and it it assumed that the user leaves his home. As soon as he exits a certain user-defined radius he will be notified that his light is still switched on. Together with the notification he is presented a list of available actions such as to turn the light off. This combines location/presence information with sensor information aggregated at the M2M server in order to support the user in making a decision.

C. Third Scenario: Context-Aware Automation

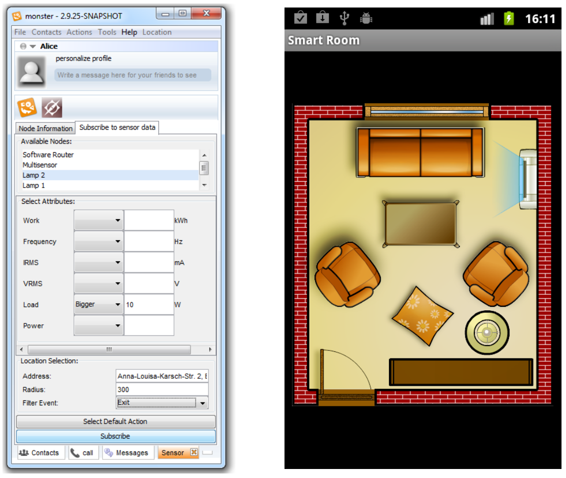
The third scenario will combine information from several distinct sources with a user-defined policy in order to automate the behavior of the system in case a certain situation is detected. For this, we will assume that the user comes home (enters a certain user-defined radius around his home location) and the temperature is above a user-defined threshold. In this case, the fan is automatically started. The policy consisting of the trigger conditions and associated actions is defined using the IMS client application. This scenario gives a nice “demo effect” because the audience will see the temperature raising above the threshold and than suddenly the fan starts showing the system communication and automatic behavior in a tangible fashion.

IV. Scientific and Technical Discussion

This section discusses the core technologies and scientific challenges that are associated with the demo. Most M2M systems today have been designed to function in a specific context and solve a single problem in a vertical application domain. Our system is able to aggregate information from arbitrary sources and automate actions based on user- or network-defined policies using operator technologies.

For example, our IMS system implementation includes IMS Call Session Control Functions and a lightweight Home Subscriber Server, which together form the core elements of all IMS/NGN architectures as specified today within 3GPP, 3GPP2, ETSI TISPAN, and the PacketCable initiative. The IMS provides services which can be facilitated by attaching application servers (service enablers) like presence server, XMDS, etc. to the IMS network. IMS clients can then connect to the IMS and use these services. The communication between the IMS, clients, and service enabler is done via the Session Initiation Protocol (SIP) [5].

For this demo, the main service enabler is a generic context server (M2M server) capable of handling context information and providing event-based services based on different kind of context information like presence, location, and sensor data. Also, there are two different kinds of IMS clients: the M2M gateway and the user client (IMS client / Android App, see also figure 5). With the integrated IMS client, the M2M gateway publishes the sensor information via SIP PUBLISH [6] to the M2M server. The M2M gateway provides sensor information such as availability information, describing the currently available sensors and capabilities of these sensors, and sensor data which contain measurement values of the sensors. The data format used for the description of capabilities and data of sensors is based on a proprietary XML format. We are in the process of replacing this with the data format defined by the latest ETSI TC M2M specifications [7], [8].



Available Nodes:
Software Router
Multisensor
Lamp 2
Lamp 1

Select Attributes:

Work			kWh
Frequency			Hz
IRMS			mA
VRMS			V
Load	Bigger	10	W
Power			

Location Selection:
Address: Anna-Louisa-Karsch-Str. 2, E
Radius: 300
Filter Event: Exit

Fig. 5. Client screenshots: IMS client (left) and Android application (right)

Actuation of sensor devices is realized via a SIP MESSAGE which is sent to the M2M gateway which then triggers actions on the actuating sensor devices. The SIP MESSAGE contains an XML document describing which action shall be triggered on which actuator.

On the other side, there is the user client which receives sensor data via a subscribe/notify mechanism using SIP SUBSCRIBE and NOTIFY [9] messages. The client subscribes to the M2M server for sensor information and gets notified when relevant new sensor information is received from the M2M gateway. These subscriptions may contain filter criteria to get only notified if certain conditions are met. Once the user client receives capability information of currently available sensors, the client can create subscriptions that contain policies based on the capabilities of sensor parameter/location information that describe in which situations the user client shall be notified by the M2M server.

The M2M server is the data aggregation point. It receives context information about sensor and location data and subscriptions for such data. Every time the M2M server receives data, it evaluates the filter conditions of the received subscriptions and notifies the subscribed IMS client if the conditions are met. This enables the convergence of classic telecommunication services, in this case location based services (LBS) [10], and M2M services. These services can also be used independently of each other, but the context server provides a generic mechanism to combine services in order to create new converged and more complex services. This is the major advantage over silo solutions, as a single platform can be used to support multiple verticals and combine data and functions across those in a meaningful manner.

The amount of transported sensor data may become quite large when many sensor devices are involved. Since the IMS is not meant for payload transmission but for session management, a content indirection mechanism based on HTTP is used. The M2M server notifies the client about an event by including an HTTP URL into the notification message and the client fetches the actual data via HTTP.

Such communication mechanism related questions will be subject to further research given that the new ETSI M2M specifications define a resource-based approach and propose HTTP as a transport protocol. We currently use the system that is demonstrated here for experiments comparing the resource-based vs. session-based communication paradigms for M2M traffic in telecommunication operator networks.

Demo visitors can use our client devices (smart phone, tablet, notebook) to access the IMS/Android/Smart Home applications and run through the scenarios defined in section III, control the lamp/fan, and feed the M2M system with their own policies to trigger automated sequences of actions.

References

[1] S. Wahle, T. Magedanz, and F. Schulze, "The OpenMTC framework – M2M solutions for smart cities and the internet of things," in *World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM 2012)*, June 2012, pp. 1–3. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2012.6263737

[2] (2012, June) Fraunhofer Open IMS Core website. [Online]. Available: www.openimscore.org

[3] (2012, June) Fraunhofer OpenEPC website. [Online]. Available: www.openepc.net

[4] (2012, June) Fraunhofer MyMonster client website. [Online]. Available: www.monster-the-client.org

[5] J. Rosenberg *et al.*, *SIP: Session Initiation Protocol*, IETF Std. RFC3261, 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[6] A. Niemi *et al.*, *Session Initiation Protocol (SIP) Extension for Event State Publication*, IETF Std. RFC3903, October 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3903.txt

[7] *Machine-to-Machine communications (M2M); Functional architecture*, ETSI Std. TS 102 690, October 2011. [Online]. Available: webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=30459

[8] *Machine-to-Machine communications (M2M); mIa, dIa and mId interfaces*, ETSI Std. TS 102 921, February 2012. [Online]. Available: webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=34000

[9] A. B. Roach, *Session initiation protocol (SIP) - Specific Event Notication*, IETF Std. RFC3265, 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3265.txt

[10] *Location in SIP/IP Core V1.0*, OMA Std., January 2012. [Online]. Available: www.openmobilealliance.org/Technical/release_program/locsip_v1_0.aspx