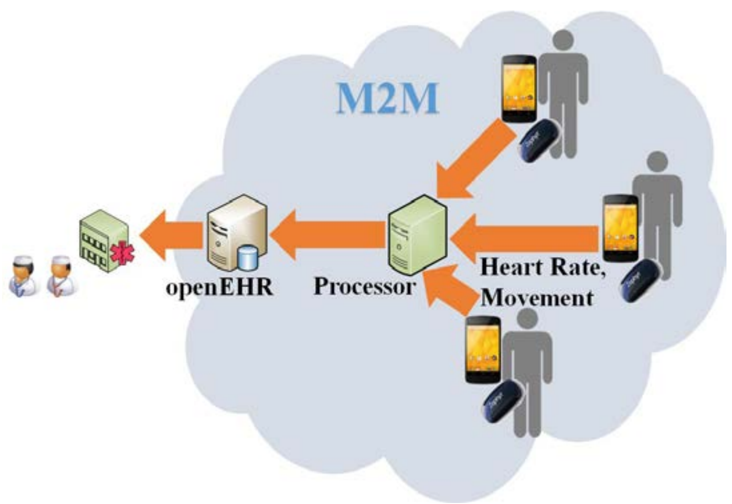
E-M2M 2-Energy-Eficient M2M Gateway 2

We evolved an ETSI M2M Gateway on a smartphone that resulted from a previous project, focusing on resource efficiency on the smartphone. We created libraries to ease the deployment of IoT applications using the ETSI M2M ecosystem. Finally, we validated the implementation with a mobile e-health pilot with 10 participants during 3 weeks, and evaluated the latency performance of an IoT application.



Main Project Team	
Ana Cristina Costa Aguiar	**NS-Po**
António Filipe de Carvalho Pinto	NS-Po
Carlos Miguel Silva Couto Pereira	NS-Po
Funding Agencies	
PT Inovação e Sistemas, S.A.	**20,000€**
Start Date	01/09/2014
Ending Date	31/05/2015
Indicators	
Journal Papers	1 (submitted)
Conference Papers	2
Concluded MSc	1

Two Main Publications

C. Pereira, A. Pinto, P. N. Rocha, F. Santiago, J. M. Sousa, A. Aguiar, **IoT Interoperability for Actuating Applications through Standardised M2M Communications**, 5th workshop on IoT-SoS: Internet of Things Smart Objects and Services, Coimbra, Portugal, Vol. -, pp. 1 - 6, June, 2016

C. Pereira, S. Frade Frade, P. Brandão, R. Correia, A. Aguiar, **Integrating Data and Network Standards into an Interoperable E-Health Solution**, SSH: IEEE Workshop on Service Science for eHealth - SSH, Natal, Brazil, Vol., pp. 1 - 6, October, 2014

PROJECT WEBPAGE URL
https://www.it.pt/Projects/Index/2064

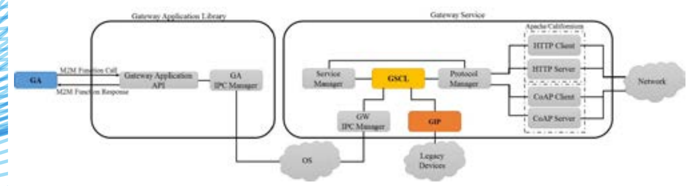


Fig. 1 Architecture of the M2M gateway to be deployed on smartphones.

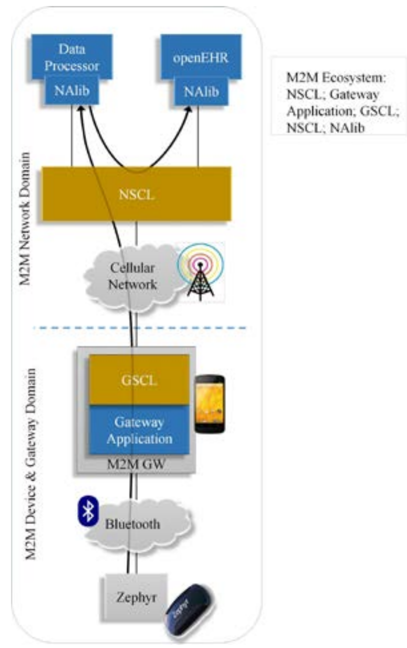


Fig. 2 Overview of deployed mobile IoT application with service composition using ETSI M2M for interoperability.

GENERAL MOTIVATION AND OBJECTIVES

In mobile M2M communications, smartphones are envisioned to play the important role of M2M Gateways (GWs), acting as proxy for nearby devices with constrained resources and limited connectivity, while likely be used as sensors themselves. However, users utilize smartphones for purposes such as SMS sending, Web browsing, phone calls.

Using a smartphone to host an M2M GW can impact the usability by introducing undesirable battery depletion due to network accesses, or CPU and memory usage.

Thus, we evaluated the impact of using a smartphone as an M2M GW, and improved the previous GW accordingly. To help application developers create their own M2M applications, we also developed a set of libraries for wrapping the complexity of the ETSI M2M standard. We experimentally validate the performance of all components using an e-health pilot with 10 people for 3 weeks, collecting nearly 480 hours of data. We measured latency between system components to assess the capabilities and limitations of current architectures.

CHALLENGE

Interpretation and implementation of standards is a cumbersome task. Specifically, here we address the implementation of actuation from M2M Network Applications (NAs) in M2M GW and GW accessibility necessary for actuation. On the other hand, latency is a critical metric of service composition performance, e.g. critical/ emergency applications respond to event alarms and emergency situations. Thus, M2M middleware should guarantee predictable and measurable End-to-End Quality of Service. However, this aspect has received little attention so far, due to lack of access to such platforms and applications on top of them.

WORK DESCRIPTION AND ACHIEVEMENTS

We implemented a Gateway Application (GA) library that wraps the standard interface, and makes the GW service accessible to all smartphone's applications through a simpler and cleaner programmatic interface. We also enable communication between GAs and SCLs in the same device without setting up network clients and servers by exploiting Android IPC capabilities.

We expose the GW's capabilities by publishing supported actuation commands under a container DESCRIPTOR. M2M NA's can retrieve the latest Content Instance to learn about available actuation commands. The GW is globally addressable, even behind NAT or firewall, via one of three options: UPnP, NAT-PMP, and Long Polling.

The GW sends data as collected (raw bytes), thus delegating the data processing to the NA, and precluding the need to integrate each device, since data must not be interpreted. This reduces the overhead introduced by JSON, and reduces the CPU and Memory usage.

We measured the performance of a smartphone as an M2M GWin terms of CPU, Memory and battery life. We showed that our M2M GW has a limited impact (<5%) on the CPU and Memory usage. The use of sensors can lead to undesired battery depletions and it is advisable to use data buffers, to decrease the frequency of network accesses, and to reduce battery consumption.

We carried out an experimental evaluation of latency in a IoT application with mobile gateways and service composition, using an e-health framework that combines M2M communications with openEHR, and that monitored 10 users for 3 weeks. We showed that latencies between a smartphone M2M GW and NSCL have the largest contribution to the end-to-end latency, which exceeds 1 second most of the time for infrequent transmissions, and depends on mobility. Moreover, our results show that the latency observed between services can extend the ping latency by 8000%, which can pose a problem to interactive applications. We verify that wireless access makes up most of that latency due to promotion delay, and conclude that IoT middleware should provide means to support low latency for applications with real-time requirements. An option is to provide a more expressive API that either enables conveying delivery guarantees or provides information that allows programmers to make that decision.