ISSN 2277-3061

Communication Architecture design for an Interoperable Machine-to-Machine System

Guillermo Talavera[1], Antti Iivari[2], Xavier González[3], Jordi Carrabina[1]
[1] Universitat Autònoma de Barcelona
Escola d'Enginyeria; Campus UAB
08193 Bellaterra; Barcelona
[2] VTT Technical Research Centre of Finland
Kaitoväylä 1, Oulu
FI-90571 Oulu, Finland
[3] ITC S.L.
c/Mar Adriàtic nº 1; Pol. Ind. Torre del Rector,
08130, Sta. Perpetua de Mogoda,
Barcelona, Spain

ABSTRACT

In recent years, we have witnessed a continuous increase in the number of embedded devices with communication capabilities that are changing the way we live, work and play. Smart grids, remote monitoring and control of all kinds of consumer devices and industrial equipment, vehicular telematics and e-health devices, are some examples of this revolution. The communication between those devices (Machine-to-Machine communication) is leading to a complexity explosion and a strongly fragmented market. The goal of our work is to design an architecture for a generic communication system enabling many kinds of services and devices to function together in a distributed M2M ecosystem regardless of the application domain. This paper presents an initial communication architecture design for an interoperable Machine-to-Machine (M2M) system. The architecture of the system itself is divided into three main components: gateways, distributed servers and communication overlay. Gateways are designed to enable interoperability with various external systems that are, for some reason, unable to directly become parts of the M2M overlay. Servers are required to act as central points for relaying messages, providing authorization, enforcing security policies such as channel encryption and so on. Multiple intercommunicating servers, or server federation, is supported and considered an essential part of the interoperable M2M system. The overlay component refers to the logical M2M network, which is built on top of the existing ICT infrastructure.

Indexing terms/Keywords

Machine-to-Machine; M2M; Communications; Software; Networks; Applications

Academic Discipline And Sub-Disciplines

Communications, Tele-Communications infrastructure, computer science

SUBJECT CLASSIFICATION

Computer Science

TYPE (METHOD/APPROACH)

Survey

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol. 14, No. 11

www.ijctonline.com, editorijctonline@gmail.com

ISSN 2277-3061

1 INTRODUCTION

Machine-to-Machine (M2M) refers to the automated exchange of information between servers, sensors and various end devices such as mobile phones, vending machines, vehicles and personal computers. This brings many obvious advantages such as real time data exchange, remote monitoring and operation according to real needs, business information, consumption statistics and operational data are made readily available. Different application domains are brought together by M2M technology. Data communication takes place via established ICT (Information and Communication Technology) infrastructure and communication media is based on both fixed and mobile systems.

In essence, M2M (Machine-to-machine) is always a mixture of various kinds of electronic devices, communication technologies and software implementations. M2M can be defined simply as machines communicating without, or with very limited, human intervention. M2M constitute systems that enable all wireless and wired devices to communicate with other devices included in the M2M ecosystem. Devices such as sensors and meters are used to capture raw data which is then transferred through a network to the M2M applications. These pieces of data are then transformed into meaningful pieces of information by the background system. Any interoperable M2M system will obviously need to support a mixture of legacy and modern technologies and protocols, which is already a tough challenge in itself.

It is not hard to imagine a wide range of applications, where hundreds of various physical everyday objects around us are interconnected and communicating with one another. This vision is no longer far from reality as device costs and sizes are getting lower while wireless communication technologies are getting more and more efficient in terms of power requirements and bandwidth usage. M2M can be applied in several application domains, including: remote maintenance and control, security & public safety, smart grid, tracking and tracing, vehicular telematics, payment, healthcare & wellness, consumer devices & entertainment etc. However, the current lack of widely accepted standards and interworking mechanisms are dramatically slowing down industry progress.

The motivation for this work arises from the complexity explosion problem and a strongly fragmented vertical M2M market, where technological solutions are deployed specifically to a single specific application, with no concern for wider applicability or interoperability with other systems Our selected approach for solving this problem is application of the autonomic communication solutions for enhanced interoperability in the context of M2M networks. The approach for market fragmentation problem is the development of common technical and standardized horizontal M2M infrastructure applicable for several different M2M domains as we see in Figure 1. Using the same technologies is estimated to save development cost, enable interoperability and boost the arising M2M markets by contributing towards transfer from vertical towards more horizontal M2M markets.

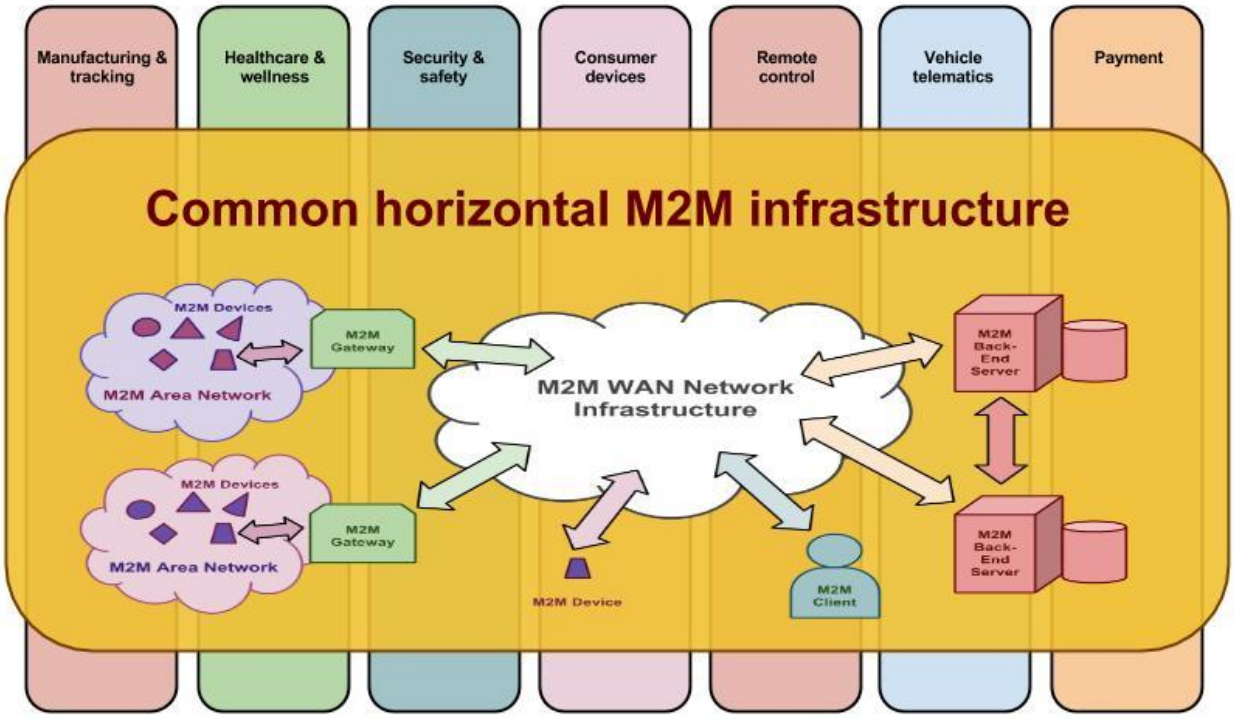

Figure 1. Common M2M infrastructure applicable to different domains.

The rest of this article is organized as follows. Section 2 introduces Machine-to-Machine background and relevant state-of-the-art, while section 3 presents some of the most important challenges and requirements for the interoperable M2M system design. In section 4 the main building blocks of the system are introduced and the general M2M communication architecture is overviewed. Section 5 concludes the article and proposes directions for future work in the domain of interoperable machine-to-machine communication.

ISSN 2277-3061

2 M2M Background and State-of-the-art

Machine-to-machine systems blur the line between the physical- and virtual worlds, making use of various communication technologies to enable remote monitoring, control, updating and interaction with all sorts of communication enabled machines and asset devices. Typically, these interactions occur without or with very limited human interaction. In this section certain communication technologies and standards relating to M2M are discussed.

2.1 Standards for M2M Communication

As the growth of the M2M communications industry has led to a clear need for standards and interoperability for M2M technologies, most of the major ICT standardization organizations have formed Machine-to-Machine related working groups. Some of these standardization activities are briefly discussed in this section of the article.

The European Telecommunications standardization Institute (ETSI) [1], 3rd Generation Partnership Project (3GPP) [2] and the Telecommunications Industry Association (TIA)[3] are some of the main standardization bodies that are putting significant effort in investigating and identifying M2M challenges, issues and architectures. Due to the fact that M2M has been declared as one of the main strategic topics of ETSI's standardization work, the ETSI has formed a M2M technical committee to facilitate the standardization of a M2M application layer. This committee in general has a clear focus on the service and application middleware rather than the actual M2M network and communication techniques. It has been tasked with the goal to develop an end-to-end overall high level M2M architecture. The committee includes experts and researchers from Europe, America and Asia representing research centres, telecommunication operators, device vendors, etc. All automated exchange of information between machines (even virtual ones) with limited human end-user influence is considered machine-to-machine communication. The focus is to develop a horizontal service platform that can be used by multiple M2M vertical applications. The group aims to provide an end-to-end view of Machine to Machine standardization that remains agnostic to the telecommunication technologies used at the lower layers.

The ITU-T (International Telecommunication Union)[4] is also working on the topic and has formed a focus group on M2M Service Layer, with the aim of developing a common M2M service layer. Also, in early 2012, Seven SDOs (Standards Development Organizations), including ARIB [5] (Japan), ATIS [6] (USA), CCSA [7] (China), ETSI [1] (Europe), TIA [3] (USA), TTA [8] (Korea) and TTC [9] (Japan), have taken initial steps to establish a worldwide M2M initiative, with an initial focus on standardization of a machine-to-machine service layer. The purpose of this M2M global initiative is to be open allowing also other organizations and parties to participate at various levels. This initiative, now referred to as oneM2M, was officially launched in July 2012.

From an M2M system engineer's viewpoint, considering the architecture design work especially, the work currently being carried out by these various standardization bodies is very relevant and interesting as it might eventually provide true industry standards for M2M communication and other aspects of the Machine-to-Machine framework. However, these M2M standardization initiatives have all been quite newly formed and hence the actual low-level specifications, communication interfaces and technical implementations are still mostly under development. The documents that have already been released are mostly dealing with requirements and high-level or abstract service-layer architecture analysis without going very deep into the actual functions and communication interfaces of the proposed solutions.

Therefore, the on-going work of the various standardization organizations should be closely monitored in order to keep the M2M system approach designed herein aligned with M2M standards being developed. If the M2M system components do not directly conform to certain specifications, conformity with the current M2M standards could be eventually achieved by designing suitable interfaces or by building novel interworking mechanisms.

2.2 Communication Technologies for M2M

The main technologies involved in the M2M communication process can be categorized in four different groups as follows: wide area communication, asset networks communication, software platforms and embedded platforms. Some of these technologies will be briefly discussed in this section.

Wide area communication technologies

Wide area communication technologies connect multiple geographically diverse area networks facilitating communications between them. There are many available technologies for M2M communication, such as LTE, GPRS, UMTS, GSM, WiMAX, Satellite, etc. [10]. All of these technologies are inherently different in many ways (in bandwidth usage for example) that must be carefully evaluated in order to design the optimum communication solution for the problem targeted.

The broad spectrum of wireless technologies available for wide area communications enables us to reach virtually the entire globe with M2M technology. However, not all technologies are available everywhere, so that interoperability should be a factor to consider in global networks. Another significant aspect is the variation in latency and bandwidth that present different alternatives and influencing the performance or QoS (Quality of Service), affecting application availability.

Providing a systematic classification of wireless wide area networks in the M2M context is a difficult task due to the complexity and heterogeneity of the all the possible cases. Nevertheless we can propose some groups attending to some common characteristics:

- Metering applications do not require wide bandwidths as they are typically not transferring large volumes of information and hence, almost all technologies can be used.

- Remote control: data volumes are small except for program modification operations, but require shorter latencies
- Tracking applications use to have small data volumes, but require mobile communication, due to their intrinsic mobility.
- Alarms: communication ranges are small and the intervals of communication are sparse but require large bandwidths from time to time (e.g. video, voice).
- Data loggers: The latencies and intervals of communication can be large, the bandwidth can be small, but may require large volumes of information.
- Billing: require small volumes of information and low bandwidth but the latency must be short.

The following table (Table 1) shows the possible technologies that can be used for each of the previously explained group of M2M applications.

Table 1. Wide area communication technologies vs. Use Cases

	ADSL/MPLS	2G MOBILE	3G/4G MOBILE	SAT.(WB)/WIMAX	SAT.(LOS)
Metering	✓	✓	✓	✓	✓
Remote control	✓	✓	✓	✓	
Tracking		✓	✓		✓
Alarms	✓	✓	✓	✓	
Dataloggers	✓	✓	✓	✓	
Billing	✓	✓	✓	✓	

Communication technologies of M2M asset networks

Asset networks or Wireless Sensor Networks (WSN) [11], another important aspect of M2M communication, are essentially wireless networks consisting of wireless sensor nodes (that may interact which each other) aimed at monitoring real world physical parameters (in many cases, covering a certain geographical area) and offering the sensed data to one or more data collection elements.

Communication among devices in WSNs is enabled by the following types of protocols: physical layer protocols; Medium Access Control (MAC) protocols; Routing protocols; Transport protocols; Data encoding and aggregation protocols, and additionally, there are important cross-layer services in WSNs such as security and topology control.

There are many available technologies [12] for wireless sensors networks as Bluetooth low energy, ANT, ZigBee, 6lowpan, etc. all present differences (in terms of bandwidth, protocol, QoS etc.) that must be carefully evaluated to get the best solution for the problem targeted. It is difficult to establish a single criterion for wireless sensor networks for all possible applications in the context of M2M but a priori we can establish large groups depending on your use case as in the case of wide are M2M technologies.

- General purpose solutions (ZigBee, 6lowPan, Bluetooth Low Energy, etc.) [13][14][15][16][17][18].
- Home/Building automation and metering solutions (z-wave, EnOcean, Insteon, X10, ONE-NET, etc.) [19][20][21][22][23]
- Other solutions (SensiNet, Dash7, XMesh, ANT, SIMPLICITI, DigiMesh, etc.) [24][25]

A software platform is computing related term which includes some sort of hardware architecture and a software framework (including application frameworks). The combination allows software to run. Typical platforms include a computer's architecture, operating system, programming languages and related user interface (run-time system libraries or graphical user interface), network services and systems of communication.

The software platform should focus allowing the incorporation of recognized standards and reliable technologies. These technologies involved in the underlying architecture and software platform should provide a clear trend towards integration and interoperability. These technologies should be considered as references already consolidated where the exchange of information with external systems should be based on certified standards for interoperability, at level of communications, transport protocols, data exchange protocols and service engineering.

The software platform should include various software frameworks, subsystems, components and services to envisage different scenarios, technologies and all the software components that are part of the interoperable M2M architecture, such as:

- Metering devices (example: software for Bluetooth device, NFC – Near Field Communication, etc.)
- Web Applications Java platform, .NET platform, etc.

- Software Architecture Communication (IMS for example)
- REST, SOA, SCA, Web Services, etc. [26][27][28] .
- Network Protocols (TCP/IP, UDP, HTTP, etc.)
- Mobile devices (software platform for Android, iPhone OS, Symbian OS, etc.)
- Software for embedded platform: Android SDK, Symbian, Linux Embedded, OSGi services, RTOS, etc.
- Technology standards: XML, WSDL, Semantic WS, etc.
- Frameworks technologies: Web frameworks, frameworks for mobile phones, etc.
- Database technologies: Database Architecture, components of DBMS (database management system) and motes and sensors.

Embedded platforms are computer systems designed for specific control functions, often with real-time or energy consumption constraints [29] . The design of the operating system has been proved as one of the most difficult aspects to verify robustness in the long-run. That has lead to many of the available nodes in the market to be supported by at least one low-footprint operating system, and in this way many of the difficult parts, e.g. networking, are relieved from the final user or practitioner. With exceptions noted, most of the nodes in the following list are supported by the two most prominent operating systems in the WSN field, namely TinyOS and Contiki.

Some of the most widely spread nodes are those nodes manufactured in the past by Moteiv and Crossbow. This generated a lot of agreed know-how in the wireless sensor nodes field, mostly because many people shared the same hardware platform, and also because of the success of the two operating systems mentioned before. Hardware platforms (motes or sensor nodes) in the market can be classified in several groups, depending on several aspects lke size, employed RF band, MCU employed, supported OS, power consumption, targeted audience (general, medical, body-area-networks, etc.), and a few more. Some of the available ones are: MicaZ, Mica2, Mica2DOT, Tmote sky, TinyNode, Shimmer, EPIC, Zolertia, WaspMote, etc. [30][31][32][33][34][35][36][37][38].

3 Challenges and Requirements for interoperable M2m

3.1 The M2M Challenge

An M2M system always includes software services in addition to the actual machines and hardware devices, in one form or another. One of the key challenges in designing an interoperable M2M architecture is indeed the software that enables horizontal interoperability between various services and software agents.

Existing M2M solutions are typically dedicated to a single specific purpose and serve only a very limited set of usage scenarios or application domains. In these cases, the software and hardware, sometimes even the communication protocols, are tailor-made for a certain proprietary products. This can be seen as “vertical M2M”, where technologies are customized and fitted specifically for a single solution, with no concern for interoperability or the bigger picture. Also, the multitude of technologies and competing standards, none of which are truly interoperable or able to understand each other, are causing major issues for engineers working within the M2M field. Consequently, there is a dire need for an interoperable M2M architecture that ensures horizontal interoperability between M2M clients and applications while still providing all the necessary functionalities to cover all application domains [39].

It is clear that steps must be taken to ensure interoperability with existing commercial and resource constrained systems and devices to the extent reasonably possible. Another critical issue is the efficient utilization of all the existing communication infrastructure for secure M2M message exchange. There are way too many competing protocols, tools, technologies and standards already in the field and, therefore, the focus should be on finding entirely novel approaches by combining and enhancing selected existing widely accepted technologies and harnessing them to serve M2M in new and interesting ways. The ultimate goal is to provide an architectural backbone, so to speak, consisting of certain central universal building blocks in order to enable various mechanisms for applications, services and users to interact, operate, exchange messages, utilize each other’s capabilities, share resources and discover each other in ways that are fitting and are useful in any application domain. With this design philosophy the aim is to provide common and basic connectivity, gateway, server, security and other building blocks for the whole M2M ecosystem. Also important security policies, such as authentication and channel encryption, must be built into the system and enforced.

3.2 Communication requirements for M2M

The communication requirements of any M2M complete system must satisfy different utilities, services and metrics that can vary depending on the usage scenarios. The different metrics should be evaluated and carefully traded-off to reach a satisfying end result.

Interoperability

Even if different definitions exist, interoperability can be defined as the capability to communicate, execute programs, or transfer data among various functional units in a manner that require the user to have little or no knowledge of the unique characteristics of those units. This property is crucial in horizontal M2M markets.

Transversability

In the same way that the systems must be interoperable, they must also be traversal. We defined this feature by the capacity of transfer data through the different agents involved in the communication process from the last mille device/service to the servers or to any other service provider that requires the data. As an example, in a cellular system, the phone network is interoperable between service providers, and the language provides spoken transverse communication between different people on the communications network.

Many data models are intrinsically linked with the medium or communication protocol which defines both the physical channel as the structure of such information but the change in the data transport layer, has become evident the need to isolate the transport of the content. Due to this fact, for many M2M systems, the communication needs to be transparent to the data structure, and that means that M2M devices must work at layer 2 of the OSI model where data structures are transparent.

Availability

Availability means the ratio of time than a unit is capable of being used. Availability is crucial in many M2M scenarios where the user or device can find very diverse and changing conditions. For example, a user inside a car can be moving at high speed, inside a car park, a tunnel, etc., or it can be in “unexpected” locations as a mountain with little or no connection. IP techniques through the different available networks (3G, GPRS, WiFi or even satellite) can cover most of the territory and provide as much as possible reliability.

Reliability

When considering the various M2M application scenarios, reliability usually refers to the ability to deliver packets to their destination with a very low risk of delivery failure [41]. These packets can have crucial information, might need high levels of confidence in the reliability of the system, at least of some of the services and data accessible.

Reliability can be improved by redundancy in the communication protocols and with different mechanisms of automatic reconnections, watch dogs, low-energy battery detection, etc.

Metrics: Bandwidth and latency

The applications must be able to operate in different contexts and link technologies that can make big changes in terms of latency, bandwidth and other communication characteristics [42] [43]. For example, different types provided for binding are: WIFI for parking and service stations and GPRS 3G/UMTS, while roaming.

Many utilities involved in different M2M scenarios can require still different relationships between latency/bandwidth. In particular the events related to the operation between the user and the system as well as alarm events, require low latency and the bandwidth usually is not critical, while the transfer of files associated with tracking and management system are not critical in terms of latency, but require more bandwidth, it is also necessary to consider the course of events and establish a proper policy priority and / or QoS, so that the critical mechanisms, are not affected in their functions.

Generally the following criteria are considered:

- Information that is not generated by events will be sent only by request of the interested system (pull).
- Remote systems avoid the method of sending periodic information to upper layers if it is not specifically requested.
- The intermediate systems of the data tree, act as "proxies" so that the information is transmitted only once to the upper layers and only towards the subsystems involved, maintaining the same at intermediate nodes for future reference.
- All information is transmitted via encryption and compression algorithms.

Services for historical monitoring of events such as vehicle routes, and time for each service, can send the information to close the transaction in batch. Such transactions will be conducted taking into consideration the availability of service at all times taking into account the available bandwidth and the cost thereof. Services involving the periodic transmission of historical data associated can be managed in pull, to avoid denial of service issues, operating in push only in case of event. It should be noted, that in systems with a large number of remote units, push models have problems of scalability when the number of participants increases. In these cases, it is necessary to control the synchronization of all clocks and careful programming of the communication window for each remote device. The incorporation of new remote devices presents problems of restructuring the pulling sequence which must be adapted to integrate new participants, so it should incorporate mechanisms to prevent this mode of operation.

Metrics: Memory

The memory capacity of the system should be sized to accommodate the operating system and related services according to usage statistics and traceability. Other services require a small percentage in relation to these. These services, as batch discharge their contents, must have sufficient memory resources to store the data generated during estimates loss intervals broadband connectivity. As an example, in our work we analyzed the memory requirements for a complex M2M case scenario: a rental car with full access of data coming from different sensors. The memory requirements of our analysis can be found in the following table:

Table 2: Example of memory Requirements vs. utilities

Utility	Memory Capacity	Period	Range
System	2GB	Static	Static
Car route logs	3MB	10 Days	1 min.
OTA services	2GB	Static	Static
Car services maintenance	50MB	30 Days	1 min.
Communications log	10MB	30 Days	1 hour

The personal data associated with the user is sent in real time, generating and maintaining in the remote systems a unique identifier to relate them in the database of the system. This precaution avoids distributed files with personal information that may be affected by national regulations on data protection, the rules of application being restricted to the central management systems.

As part of the communication is done via the mobile network, one must provide the historical record of traffic and subsequent loss of service for administrative control.

Metrics: Energy consumption and power

Energy consumption and power are two main issues in many M2M communication process [29][44]. For plugged in devices energy and power efficiency might not be so crucial. Nevertheless these two are important parameters to take care into account and carefully evaluated for the success of the project.

Metrics: Price

Price can also very strongly depend on the application domain. Nevertheless, in order to reduce the cost of an adequate service performance and cost ratio, the following criteria were applied in the design of the various utilities and services:

- The data transmitted is encrypted using algorithms that incorporate the highest possible degree of compression, using two-dimensional techniques of encryption / compression.
- The data captured and stored, be studied in such a way that is not stored information that can be derived indirectly.
- The mobile gateways provide Hardware SIM technology and enable roaming between operators, so that the system can adapt dynamically to the best possible relationship QoS / Operator / price at the time.
- When a Wi-Fi connection is available for M2M link, it will automatically switch to this technology.

While the initial cost of the equipment must be adjusted to the needs, you should minimize the impact of recurring costs

3.3 M2M System Requirements

In this section, the aim is to itemize and specify the general system-level requirements for the horizontally interoperable Machine-to-Machine system architecture, which will be further introduced in the following sections of this article. Requirements can be seen to arise mainly from the various already existing M2M applications and usage scenarios. The idea is to extract and compile these known requirements in order to compose a comprehensive set of requirements for the Interoperable M2M system discussed herein. This set of architectural requirements will then act as the basis and starting point later on in the design process of the interoperable Machine-to-Machine system architecture.

As all the application domains related to M2M will include a lot of very resource constrained devices (temperature monitors, small actuators, battery powered devices, etc.), it is clear that a Gateway-functionality of some sort is required. These extremely limited sensor devices cannot be made to directly link themselves to the M2M background system due to their constrained nature. Therefore, a device is needed that acts as a translator between a certain application specific set of devices and the rest of the M2M system. This device will make the services and information provided by these limited devices available and accessible to the rest of the system. The requirements for the M2M gateway architecture are presented in the following:

- Gateway for proprietary protocols - The M2M Gateway should have the capability to communicate with other devices that are limited by a proprietary protocol that cannot be supported by the whole system.
- Gateway for constrained devices - The M2M Gateway should be able to communicate with resource constrained and low-power devices that are not directly part of the M2M ecosystem.
- Gateway as a translator - The M2M must be able to act as a "translator" for proprietary communication protocols and a various sensor data-formats employed by the devices the M2M ecosystem.
- Gateway information relaying - The Gateway should be able to contact the M2M Back-end in order to make known what kind of resources are "behind" it (e.g. sensors, actuators), what they can provide and how they can be accessed by other devices in the M2M ecosystem.

- Gateway for two-way communication - The Gateway must be able to provide 2-way communication between the device "behind" the gateway and M2M services or applications.
- Data filtering and prioritization - In some environments, the gateway must be able to make smart decisions upon what incoming data is important enough to be relayed. Some things communicated to the gateway device may be more important, while others may just be "noise".
- High-level of self-configuration and automation - The Gateway should be able to autonomously make connections to various devices and the M2M back-end and start relaying necessary information with as little user intervention as possible.
- High-level of adaptation to the current environment - The gateway should be able to adapt its operation characteristics based on the current application environment.

The various M2M devices included in applications with strict privacy requirements, but also other types of M2M devices on a more general level, need to be able to efficiently and securely communicate with each other by utilizing the already existing ICT-infrastructure (such as the internet). In order to accommodate this, an overlay network of sorts will be formed between the devices included in the M2M ecosystem. The more specific requirements for the overlay aspect of the system architecture are presented in the following:

- Utilize existing ICT infrastructure - The M2M system should operate in a manner that utilizes the existing ICT infrastructure for maximum benefit.
- Serverless Messaging - A mechanism must be provided for serverless communication between M2M devices in challenging environments without reliable connections to a back-end.
- Internet connectivity - The M2M system should provide internet connectivity whenever possible.
- Overlay operation - The M2M System shall operate in an overlay fashion in order to hide the diversity of the various supported protocols, radio technologies and underlying devices from the end-user and M2M Service applications.
- Group Communication - The M2M overlay shall provide a mechanism for group communication and one-to-many type of messages.
- Publish/Subscribe - The M2M overlay shall provide a PubSub method to avoid frequently polling for information in certain use cases.
- Addresses & Naming Scheme - Every M2M entity shall have an address/name other than raw protocol (e.g. IP) address-numbers. In naming M2M entities, a widely accepted scheme should be adopted (e.g. something DNS based).
- Name uniqueness - Every M2M entity shall have a unique name or identifier.
- Status Monitoring - The M2M API should provide a method for accessing information about the device (if at all supported and possible when considering the individual device/platform in question), such as application specific data, logs, battery/memory usage, device capabilities or other information.
- Message Loss reporting - In case M2M messages are sent but cannot, for some reason, reach the specified destination or other communication failures occur, the system should have a way to report this to the sender.
- Open standard - The M2M Messaging solution should be based on an open, widely available and widely accepted standard or technology.
- Extensible - The capabilities of the M2M system should be easily extended by writing additional functionality to the open code.
- Indifference to the underlying communication technologies - The system must be indifferent to the underlying radio/network technology and protocols. Applications running over the M2M platform must be able to operate regardless of the communication technique.

It should also be considered that much of the information exchange occurring in typical M2M scenarios is of a client-server nature. The Gateways or individual M2M client devices connect to one or multiple servers. This is for the purpose of efficiently gaining information on other devices and services in the M2M system. By first contacting a server, a new entity may quickly and securely become a part of the system and start sharing and/or consuming information. In the following we concentrate on the system requirements especially related to the M2M server side:

- M2M client-server communication - A mechanism must be provided for an M2M device to attempt communication with the M2M server in a client-server fashion.
- M2M server queries - There shall be a query-mechanism in the API for M2M devices to query the back-end in order to gain information on other (nearby) devices, services in the M2M ecosystem or other relevant information.
- Presence Information - M2M server will provide M2M Devices with presence information on other known nodes (e.g. online/offline/sleeping). Entities should be able to affect the way their presence is made known to others by the system.

- M2M server data collection - There must be a way for the M2M server to store and collect relevant data to ensure smooth operation and resource discovery within the M2M ecosystem.
- M2M Federation - The M2M system should provide a mechanism for devices in different domains to communicate with each other via secure "server-to-server" connections.

By having the devices connect to a background system in this manner we will also ensure scalability and minimize the need for flooding the network with all kinds of searches or queries. However, as mentioned earlier in this document, in extreme situations serverless communication in some form should be supported by the system. Servers shall also have the ability to communicate with one another in order to provide a sensible method for communication between devices residing in different domains.

4 The Main building blocks and general M2M architecture

The purpose of this part of the article is to be an initial overview of the chosen approach in the interoperable M2M communication architecture design, essentially containing a high-level specification of the functional M2M Architecture and its general components and main building blocks. A high-level overview of the interoperable M2M system is given in Figure 2. In the previous chapter, we compiled a collection of common requirements for the communication architecture and the things discussed in this document are clearly built upon the work done herein.

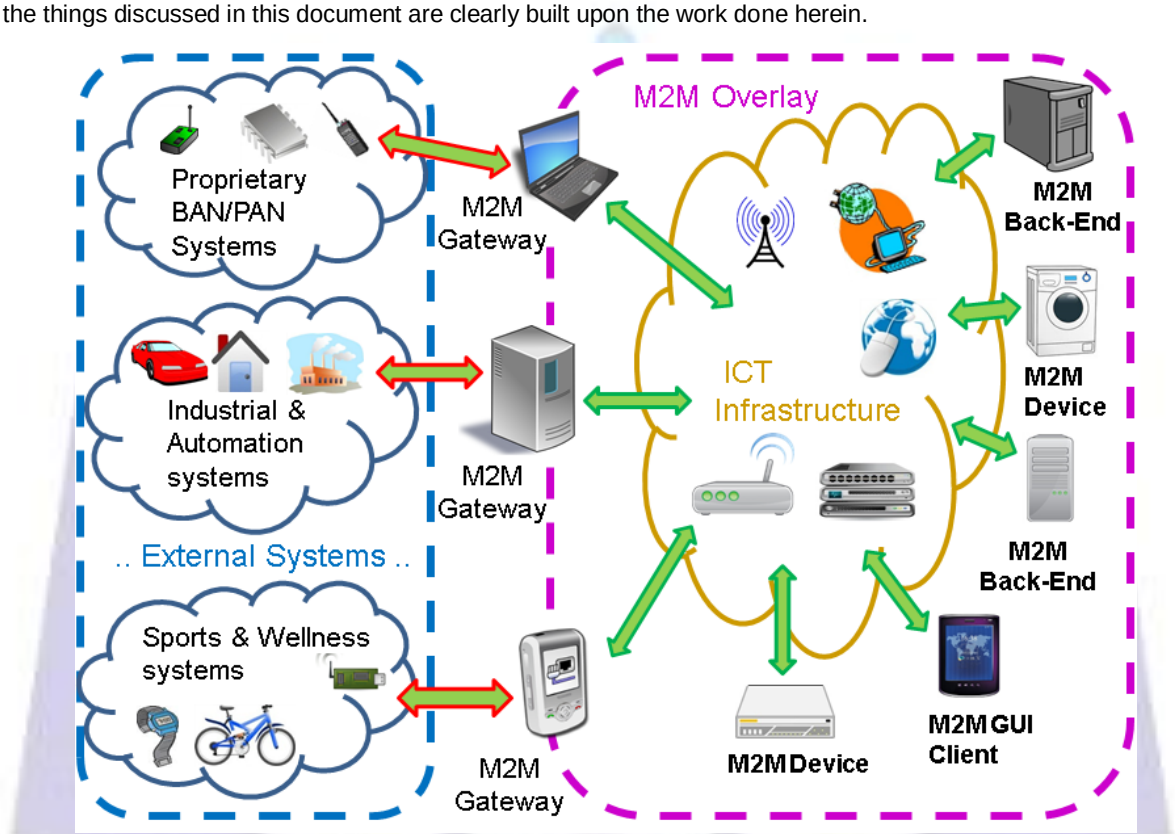

Figure 2. An overview of the interoperable M2M system.

4.1 The M2M Gateway

During the requirements gathering process, it became quickly evident that some form of Gateway-type functionality would be critical in order to meet the requirements and realize the features necessary for a truly interoperable Machine-to-Machine system. The main considerations and observations that lead to this conclusion are outlined below. In order to ensure true interoperability and prominent M2M functionalities, the system must be able to include and support:

- Various already existing legacy systems
- Future systems yet to be designed
- Constrained systems with small devices and sensors
- Delay-tolerant systems
- Challenging environments with e.g. SMS-based communication
- High-bandwidth applications, such as multimedia streaming
- Real-time applications, critical time-sensitive alerts etc.

- Commercial systems based heavily on closed & proprietary technology

The above points must be carefully taken into consideration all the while also remembering the following constraints and limitations:

- It is not possible to force every (or even most) device manufacturer in the world to suddenly start supporting this or that protocol/technology in current, future or legacy products.
- One cannot start re-engineering, disassembling or rewriting firmware in order to make certain devices or systems support this or that protocol/technology that would suit us.
- Certain systems or devices must not be ignored in favour of other more convenient systems.
- And one certainly cannot go back in time in order to make all existing devices or systems work in a way that suits us better from an M2M viewpoint.

These considerations inevitably lead us to the conclusion that a Gateway component is required and will become an integral part of the overall architecture and it is therefore further discussed in this section. The gateway is not simply a communication gateway that translates one communication protocol to another. The gateway must include application specific logic and data processing capabilities the extent of which will be dictated by the characteristics and requirements of the current application or vendor-specific system.

The gateway architecture must contain an interface between a M2M gateway and the existing system that cannot, for some reason, be directly made part of the M2M ecosystem. Reason for this might be some of the issues discussed previously (resource constrained, closed and proprietary, legacy, etc…). This interface and the case-specific “gateway-logic” must be implemented individually for each specific system or application. There is no ”one-size-fits-all” solution for this, as there are so many different systems with differing operating characteristics, requirements and limitations. Again, the gateway is more than simply a protocol-translator or a communication gateway. However, the amount of intelligence and processing load required from the gateway is highly dependent on the system to which is applied.

The other interface, in turn, depicts the interface towards and into the M2M overlay. The overlay consists of other gateways, servers and M2M clients. It is essentially a ”doorway” to enable a closed/proprietary/restricted existing system to become a part of the open M2M Ecosystem without posing any changes/restrictions toward the existing system, or letting the existing system pose limitations or requirements towards the M2M overlay design. The resources within the existing system will be made available and visible to other devices in the M2M network. The existing system and the overlay will be made independent of one another, while giving them a capability to share data, services and functionalities but also to operate together in a way. Thus, the existing systems that are operating unchanged ”behind” the Gateway need not know anything about the overlay or its protocols. Thus, we can design the Overlay and M2M functionalities therein without worrying about the various limitations and restrictions that apply only to a single specific existing system implementation or application environment.

4.2 The M2M Server

One of the most basic requirements for the system is the ability for simple, effective and secure client-server communication. This means, essentially, the communication from M2M clients or gateways towards the M2M servers. The servers will effectively provide querying devices information other available devices, services and resources in the M2M ecosystem. The servers act as “central points” or servers that collect and store data on nearby entities, their status, availability, services, resources, etc. The server will also be critical in the process for authentication and establishing trust between M2M Devices. Also, communication between servers, or server federation, must be supported. Relying on devices and services to function properly using only pure peer-to-peer and ad-hoc mechanisms, such as flooding routing/discovery queries of various kinds, in a large-scale M2M ecosystem is not realistic. It is clear that M2M server functionalities are required and form another key component of the interoperable communication architecture.

Furthermore, the interoperable M2M system must support:

- Multiple M2M servers working together and being part of the same overall system
- Server Federation, secure server-to-server communication
- Cross-domain connectivity between M2M devices

All the while it makes no sense to:

- Assume that one or two servers can handle the whole M2M Ecosystem.
- Build a system with multiple servers without having a sensible means for the servers to communicate with one another and share data.
- Design an interoperable M2M system without providing a possibility for devices in different domains to communicate with each other.

We therefore arrive to the conclusion that a distributed Client-Server Architecture provides us the means and scalability to meet these challenges. Ultimately, a vast and dynamic overlay network of clients and servers that inter-communicate will arise, forming a huge cloud-like M2M ecosystem. These M2M servers are designed to support server federation and facilitate inter-domain communication between end-devices. Most of the traffic in the overlay goes through a server. Therefore, servers are also essential in handling issues such as authentication, up-to-date presence information on devices and users, service discovery etc.

4.3 The M2M Overlay

As already discussed earlier, The M2M system must be able to:

- Operate efficiently over existing ICT infrastructure & Utilize the Internet for maximum benefits
- Function correctly regardless of the existing technologies or protocols underneath – Adaptability and flexibility
- Allow any device to communicate securely over whatever internet connectivity is available 3G, Ethernet, Wi-Fi, GPRS, ADSL, Etc.

And, again, there are certain limitations we must keep in mind. It cannot be assumed that:

- Certain convenient communication technology is available at all times for M2M messaging.
- Existing communication infrastructure could somehow be changed or altered (internet, routers, masts, telecom-operators, etc.) or the TCP/IP Stack could somehow be rewritten to enable some nifty new M2M-specific features.
- All underlying communication technologies would suddenly somehow become inherently secure for M2M applications.

The implications of the points discussed above are now discussed further. Secure, efficient and adaptive M2M Communication over existing ICT-infrastructure in an ”overlay” [45] type of manner will become a critical part of the interoperable machine-to-machine solution. On top of the TCP/IP stack we will need to have some kind of an open and extensible (e.g. XML-based) messaging solution, that will effectively create an overlay or ,“a logical network on top of a network”, if you will. Machine-to-Machine clients, asset devices, back-ends and gateways will exchange M2M messages with one another without the underlying TCP/IP world actually understanding anything about them. The standard IP-nodes are just relaying and routing them forward according to all the rules like any other packets over the internet. As not every modem, router or device along the way will directly be part, or even aware, of the M2M ecosystem, an overlay network will inevitably be formed.

5 Conclusions

When designing the communication architecture for a machine-to-machine system, interoperability will inevitably become one of the key issues. There are already dozens of existing solutions in the field of M2M, but they are always designed to work only in a specific application environment with devices that conform to certain protocols and technologies. Scalability and end-to-end security can also be added among the most important issues early in the design phase. The various different application domains relevant to M2M communication must be taken into account by designing the main components and essential features to be general, widely applicable and flexible as much as possible. Most of the existing M2M systems today are tightly focused on narrow usage scenarios that are only relevant within certain highly specific application domains. Designing an entirely new system which works only with a limited set of devices that are compliant with certain technology standard or within a tightly focused application domain is simply not enough anymore.

Therefore, the main architectural building blocks for the interoperable M2M system presented in this article have been designed with interoperability and flexibility in mind, which allows for actors in any application domain to apply and benefit from the design. We designed the interoperable M2M communication architecture based on these general components; gateways to ensure interoperability, servers to manage security, scalability and cross-domain communication and the overlay communication component to enable the efficient use of existing ICT infrastructure and legacy time-tested internet protocols.

A system such as this could also include some sort of web-interface effectively allowing users to have some manner of access to the information within the M2M system while using nothing more than a standard web-browser and a password, for example. This web-interface is one issue that will be examined more closely in future work. Keeping up with the various on-going standardization activities is also an issue that cannot be abandoned.

ACKNOWLEDGEMENTS

This work has been partially conducted within the European joint research project A2Nets under ITEA2 cluster project of EUREKA network, and financially supported by TEKES (Technology Development Centre of Finland), Catalan Government Grant Agency Ref. 2009SGR700 and Spanish project TSI-020400-2010-51, to which organisations the authors wish to express their gratitude.

REFERENCES

[1] European Telecommunications standardization Institute (ETSI), http://www.etsi.org

[2] 3rd Generation Partnership Project (3GPP), http://www.3gpp.org

[3] Telecommunications Industry Association (TIA), http://www.tiaonline.org/

[4] International Telecommunication Union (ITU-T), http://www.itu.int/ITU-T

[5] Assotiation of Radio Industries and Businesses (ARIB), http://www.arib.or.jp/

[6] Alliance for Telecommunications Industry Solutions (ATIS), http://www.atis.org/

[7] China Communications Standards Assotiation (CCSA), http://www.ccsa.org.cn

[8] Telecomunications Technology Assotiation (TTA), http://www.tta.or.kr

[9] Telecomunication Technology Committee (TTC), http://www.ttc.or.jp

[10] Martin Suater; Beyond 3G – "Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0". Wiley Ed. 2009. ISBN-10: 0470751886

[11] Ian F. Akyildiz, Mehmet Can Vuran; "Wireless Sensor Networks"; Wiley Ed. 2010. ISBN-10: 047003601X.

[12] Holger Karl, Andreas Willig; "Protocols and Architectures for Wireless Sensor Networks". Wiley-Interscience. 2007. ISBN-10: 0470519231

[13] Kevin Roebuck. "6LoWpan: High-impact Technology – What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors". ISBN-13: 978-1743044131

[14] Jean-Philippe Vasseur, Adam Dunkels. "Interconnecting Smart Objects with IP: The Next Internet." ISBN-13: 978-0123751652.

[15] Robin Heydon . "Bluetooth Low Energy: The Developer's Handbook"; Prentice Hall. 2012. ISBN-13: 978-0132888363

[16] ZigBee Alliance, http://zigbee.org/

[17] Zach Shelby and Carsten Bormann: "6LoWPAN: The Wireless Embedded Internet". Wiley Series on Communications Networking & Distributed Systems. ISBN-10: 0470747994.

[18] Bluetooth Core Specification Version 4.0, https://www.bluetooth.org/Technical/Specifications/adopted.htm

[19] Z-Wave, http://www.z-wave.com/modules/ZwaveStart/

[20] EnOcean Alliance, http://www.enocean-alliance.org

[21] Insteon, http://www.insteon.net/

[22] X10, http://www.x10.com/

[23] ONE-NET, http://www.one-net.info

[24] DASH7 Alliance, http://www.dash7.org/

[25] SimpliciTI, http://www.ti.com/corp/docs/landing/simpliciTI/index.htm

[26] IBM, "RESTful Web services: The basics", http://www.ibm.com/developerworks/webservices/library/ws-restful/

[27] Fielding, Roy Thomas. "Architectural Styles and the Design of Network-based Software"

[28] Cheng Bo, Hu Xiaoxiao, Zhang Shicheng, Chen Junliang, "Session and media signalling for communication components-based open multimedia conferencing Web service over IP network"; European Transactions on Telecommunications Currently known as: Transactions on Emerging Telecommunications Technologies. Article first published online: 16 AUG 2011 DOI: 10.1002/ett.1494. Vol 22. Issue 8. Pages 435-450.

[29] Farruh Ishmanov, Aamir Saeed Malik and Sung Won Kim, "Energy consumption balancing (ECB) issues and mechanisms in wireless sensor networks (WSNs): a comprehensive overview", Transactions on Emerging Telecommunications Technologies.Vol. 22, Issue 4, pages 151-167, June 2011. Online ISSN: 2161-3915

[30] TinyNode, http://www.tinynode.com

[31] Shimmer, http://www.shimmer-research.com

[32] EPIC, http://www.cs.berkeley.edu/~prabal/projects/epic

[33] Zolertia, http://www.zolertia.com

[34] WaspMote, http://www.libelium.com/products/waspmote

[35] Arduino, http://www.arduino.cc

[36] Digi RF Modules, http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules

[37] RadioCrafts, http://www.radiocrafts.com/

[38] Jennic, http://www.jennic.com/?gclid=CJeQzLidz6sCFUO9zAodPTTyWA.

[39] Boswarthic D., Hersent O., Elloumi O. “M2M Communications: A Systems Approach”. John Wiley & Sons, 2012.

[40] Scott Guthery, Mary Cronin, “Mobile Application Development with SMS and the SIM Toolkit”. McGraw-Hill Professional; 1 edition (November 15, 2001). ISBN-13: 978-0071375405

[41] J. C. Reyes-Guerrero, J. J. Murillo-Fuentes, P. M. Olmos, “Remote detection of interfered downlinks in wireless cellular systems”, Transactions on Emerging Telecommunications Technologies. Published online: 28 Feb 2012 DOI: 10.1002/ett.2501.

[42] Theodoros A. Tsiftsis, Kyeong Jin Kim, Kyung Sup Kwak; “Cooperative wireless personal area network systems with partial best relay selection”. Transactions on Emerging Telecommunications Technologies Volume 23, Issue 2, pages 133–136, March 2012.

[43] Erik Bergfeldt, Svante Ekelin, Johan M. Karlsson. “Real-time bandwidth measurements over mobile connections”. European Transactions on Telecommunications Volume 22, Issue 6, pages 255–267, October 2011. DOI: 10.1002/ett.1474

[44] Haojun Huang, Guangmin Hu, Fucai Yu; “Energy-aware multipath geographic routing for detouring mode in wireless sensor networks”; European Transactions on Telecommunications Volume 22, Issue 7, pages 375–387, November 2011. DOI: 10.1002/ett.1490.

[45] Keong, L. E., Crowcroft, J., Marcelo, P., Sharma, R., and Lim, S. “A Survey and Comparison of Peer-to-Peer Overlay Network Schemes”. IEEE Communications Surveys and Tutorials, 2005, vol. 7, no. 2, pp. 72-93.

Author’ biography:

Guillermo Talavera received a BS degree in Physics in 1999 and a BS in Electronic Eng. 2001 in the Universitat de Barcelona, Spain. In 2001 he joined the Universitat Autònoma de Barcelona (Spain), as an assitant teacher and received a MS degree in Microelectronics in 2003 and a PhD degree. in Computer Science in 2009. In 2010 he joined the Center of Ambient Intelligence and Accessibility of Catalonia (CAIAC) a research group of the Universitat Autònoma de Barcelona - Spain- as head of the Wireless Ultra Low Power Area. His research interests are mainly in wireless embedded systems for e-health applications, source code transformations and energy optimization for embedded systems. He can be contacted at guillermo.talavera@gmail.com.