

# **The Digital “Connected” Earth: Open Technology for Providing Location-Based Services on Degraded Communication Environments.**

Piedrafita, Ramón\* ; Béjar, Rubén; Blasco, Rubén ; Marco, Alvaro;  
Zarazaga-Soria, F. Javier

*Aragón Institute of Engineering Research*

*Universidad de Zaragoza*

*C/ María de Luna 1, E-50018 Zaragoza, Spain*

\*corresponding author, e-mail: [piedrafi@unizar.es](mailto:piedrafi@unizar.es)

## **Abstract**

In the current world, it is easy to listen that everybody and everything is connected. Over this connected world, the concept of Location-Based Services has grown in order to provide digital services in everyplace and at every time. Nevertheless, this is not 100% true because the connection is not guaranteed for many people and in many places. These are the Degraded Communications Environments, environments where the availability of high-speed communications is not guaranteed in at least the 75% of the time. This paper works over the experience of a previous work in developing light protocols that do not need broadband for communication. This work provides an extension of these protocols for the inclusion of mobile devices as elements of the communication process, and a set of libraries to allow the development of applications in Degraded Communications Environments. The work done has involved the development of two frameworks: An Android framework that makes the incorporation of Android devices easier, and a server-based framework that provides the server side for the development of the referred applications. A use case that uses these both frameworks has been developed. Finally, all technology developed is available through a public Git repository.

## **KEYWORDS**

Location Based Services, Degraded Communications Environments, Open Technology

## **Introduction**

Nowadays we have the possibility of having a complete digital representation of the Earth. This offers the base for developing several kinds of services that will use this digital representation as a reference for fixing the context to operate them. One of these kinds of services are the Location Based Services (LBS). LBS seek to offer a personalized service to users based on their geographic location. For operating, they used to use geographic information systems technology, positioning technology either side customer (e.g. GPS, Wi-Fi, etc.) or server-side (e.g. service position provided by the network operator), and communication networks to transmit information to an application that can process and respond to the request.

The fundamental basis for the growth of the LBS has been the availability of high capacity wireless networks, which has facilitated their rapid development to enable their construction and deployment according to the same patterns followed in wired networks. Although the majority of the population of the so called “Developed Countries” has access to high capacity mobile networks, there is a lot of people around the world that has not (Comisión Nacional del Mercado de las Telecomunicaciones 2005). Even in those “Developed Countries”, the situation is not the same when it is taking into consideration the territory. For example, currently in

Spain mobile broadband has an annual growth of 50%, according to the Commission for the telecommunications market (Comisión Nacional del Mercado de las Telecomunicaciones 2005), because of this, mobile broadband is considered one of the inducing factors of a new step in the development of the information society in Spain. In terms of infrastructure, there has been significant progress in the deployment of mobile broadband with speeds comparable to ADSL technologies with a 3G coverage in Spain of 80% of the population, higher than the European average of 71.3%.

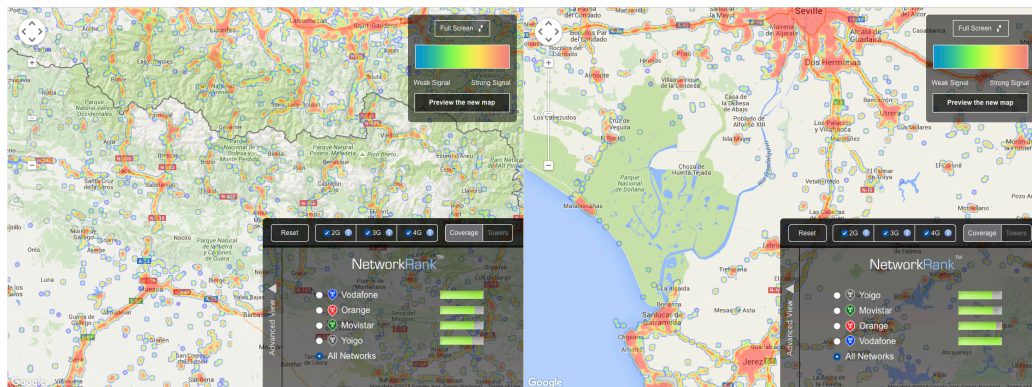


Figure 1: Mobile data coverage in Pyrenees and in Marismas of Guadalquivir (source Open Signal, <http://opensignal.com> August 2016)

Despite the foregoing, Spain has a serious problem of high-speed Internet coverage. The main operators cover more than 80% of the population, but with a clear limitation in territory coverage (see Figure 1) where only areas in orange and red have enough capacity for high-speed services). Unfortunately, it is very usual to find rural areas without 3G (or even 2G). In these areas it is very complicated to offer LBS that could be useful in several domains such as agriculture (most of the agriculture activities corresponds with rural areas), tourism (e.g. trekking activities, old villages ruins – Roman, Arabian, etc., ...), social and remote-health (population in rural areas used to be older than in urban areas), emergencies, etc. Additionally,

this situation is not expected to change because current data usage in those areas does not justify the infrastructural investment.

As Degraded Communication Environments (DCE), this paper considers such environments where the availability of high-speed communications is not guaranteed in at least the 75% of the time (see Figure 2). This includes areas with limited access to 3G/4G/WiFi communication coverage. Otherwise, they are considered Non Degraded Communication Environments (NDCE). There is not a clear definition of the reliability of an access service. This paper has worked over the specification of the Framework Directive 2002/22/EC of the European Parliament and of the Council of 7 March 2002 on universal service and users' rights relating to electronic communications networks and services (Universal Service Directive). The transposition of this Directive to national legislations has established some metrics in this sense. For instance, the Spanish legislation fix a limit of six hours without service during the billing period for the reliability of an Internet access service. After this limit, the provider has to pay penalties to the client. If we reduced the billing period to one day, six hours means 25% of the time. This implies that we do not have a quality service during at least 75% of the time. This is the reason because this paper has chosen this percentage.

One way to be able to have high-speed communications in DCE is the use of satellite communications. Nevertheless, this channel is very expensive in communications costs and in specific devices. Details of the costs and conditions can be obtained from specialized providers such as Iridium (<http://www.iridium.com>). In most cases, it is necessary to contact directly with them and request a budget for a specific use case.

An alternative is the development of technology optimized for the less level of communications that the nets can provide. This implies the creation of LBS with the corresponding optimization of the information to be exchanged, and the provision of a communication channel that could operate by using the limited resources of the nets. This paper presents an approach based on a set of libraries that implements a communications protocol suit abled for DCE developed by the authors (Asensio et al. 2014). This protocol has been designed for providing communications services over networks with IP (Wi-Fi, 3G/GPRS) or SMS messaging capabilities (GSM). The libraries that implement it have been used in the development of a LBS application scenario that includes server side (over an Internet server) and client side over an Android platform (the implementation was attempted in iOS, but it was impossible due to the limitations imposed by iOS for the sending and receiving of SMS within Apps). The use of SMS system for offering mobile services is not new (as it could be seen later). In this sense the main contributions of this paper are:

- (1) the implementation of a protocol that can manage transparently the use of different network protocols;
- (2) the availability of the source code of the libraries; and
- (3) the use of this implementation in a use case that shows its capabilities.

These contributions will reduce the problem in DCEs because they open new opportunities for developing LBSs in areas with only GSM access. This approach implies the possibility of having this kind of services in near 100% of the territory in advanced countries, and in most of the towns (<https://opensignal.com>).

Probably, one of the main problems in the development of services over different protocols is the necessity of building the technology that implements the whole protocol stack. As a consequence, there are a lot of ad-hoc solutions that cannot reuse past experiences and technologies. With the openness of the source code, this paper tries to help in this way.

This work starts with the aim of offering LBS in DCE. For this reason, the more natural fields of application are built over tracking systems. To develop tracking systems, it will be necessary to develop a suitable and optimized Protocol for DCE environments. It will also be necessary to carry out the implementation of the Protocol on the server of the tracking system and the implementation of the Protocol in the communicating nodes of the tracking system. These nodes may be specific hardware devices, provided with GSM and 3G communication systems, but also mobile devices with the operating system Android. As a result, one of the objectives of this work has been to develop a framework for Android devices which implements the Protocol of the tracking system, as well as the rules of communication with the server in degraded communications environments (SMS) and in Non degraded communications environments (HTTP), as well as the operations described in this Protocol.

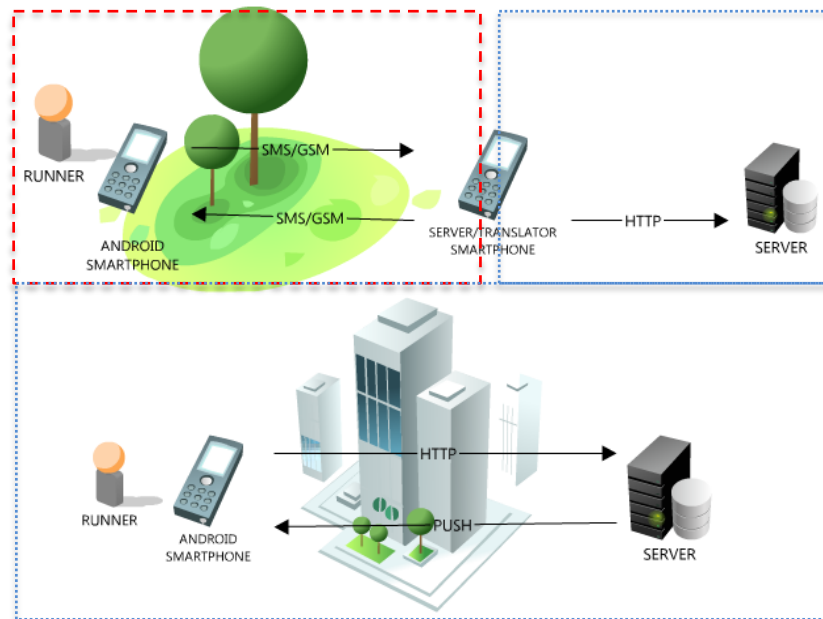


Figure 2: Degraded (red) and Non- Degraded (blue) Communication Environment.

The rest of the paper is structured as follows. Section 3 presents the State of the Art. Section 4 outlines the communications protocol technology. Section 5 shows the implementation of the Android App and in Section 6 of the server system. Section 7 shows an example of application: the tracking of sporting events. In section 8 the availability of different libraries implemented is specified.

### State of the Art

In a Degraded Communication Environment, availability of high-speed communication is not guaranteed all the time. Therefore, some alternative communication channel is required.

With the arrival of the Internet of Things, there is a great amount of technologies that may provide a fallback channel to high-speed communications. Standards such as LoRaWAN (Lora Alliance 2016) or Weightless (Weightless.org

2017) operating in the Sub-GHz band may provide communication ranges starting from 2-5 km in urban environments to 45 km in rural environments, and up to 100 kbps data rates. To use them, it's required not only a specific hardware on the device, but you need also to deploy access points (AP) acting as Internet gateways to provide coverage in the interest area.

There are also proprietary solutions such as SigFox (Ali, Shah, and Arshad 2016) or RPMA (Ingenu 2017), that don't require deploying your own APs, as there are licensed operators that provide them and offers a subscription operating scheme. SigFox technology offers communication ranges from 3-10 km in urban environments to 30-50 km in rural environments, but very low data rates (10-1000 bps), and with additional constrains in the network usage (limited number of messages allowed per day). RPMA (Random Phase Multiple Access) operates in the 2.4 GHz ISM, thus having a less penetration than technologies operating below GHz, and provides similar characteristics than SigFox, but potentially supporting unlimited devices in the same area by incorporating more APs.

The cellular industry has also introduced new standards such as LTE-M (Ratasuk et al. 2014), NB-IoT (Ratasuk et al. 2016) or EC-GSM-IoT (E-UTRA 2016) , which aim to provide similar functionalities than these technologies, but from the operators' perspective. LTE-M, or more precisely LTE-MTC (LTE for machine-type communications) may provide peak data rates up to 1 Mbps, but as it relays on the existing LTE infrastructure, LTE-M will have the same availability than high-speed communications. NB-IoT (Narrowband IoT) provides a lower data rate of 0.2 Mbps and will require operators' APs to be adapted. EC-GSM-IoT (Extended Coverage GSM IoT) is a new proposal to reuse existing GSM network that will



provide data rates of 0.5 Mbps and first commercial launches are planned to 2017. Nevertheless, these technologies are still not available for general usage, and a suitable option easily available is to use Global System Mobile Communications (GSM) messaging capabilities.

SMS (Short Message Service) was defined in 1985 as part of the GSM series of standards (ETSI 2005) as a means of sending messages of up to 160 characters to and from GSM mobile handsets. It has been supported by all mobile networks and all mobile terminals since it became available in a very early version of GSM (namely GSM Phase 2 from 1995, (ETSI 2005)). SMS are part of the core services that can be offered for everyone in every place. SMS are not a deprecated communication mechanism. They still being used by people in their day by day communication. SMS is one of the most popular channel for advertisement (Chou and Lien 2014), (Chutijirawong and Kanawattanachai 2014). Even more, (Hu et al. 2013) presents an analysis of the use of the mobile phones by university students and teachers, and it includes SMS as a relevant channel. SMS are especially relevant in offering services for people that live under the minimum economic level (four billion people at the base of the economic pyramid live on incomes below \$3000 a year) (Karippacheril et al. 2013)

This standard has given the opportunity for developing mobile-based complex services, from e-banking (Bhattacharya 2015) to IoT communication channels (Kala Fleming et al. 2016). It is very useful in the development of low cost systems with high level of communication necessities. For instance, (Alam 2014) presents a solution where RFID, SMS and Web (RSW) are embedded to develop an online monitoring system for municipal waste collection authority of under

developing countries. Similar scenario is presented by Kai Walter in (Walter 2010) where SMS are used for the communication of events from a sensor network. Event big Internet companies manage the necessity of being able to offer services over SMS. For instance, Google offer access to its most popular services over this protocol (<http://www.google.com/mobile/sms/>: SMS in Gmail, Calendar SMS, Blogger Mobile, and Google+ SMS).

More than its standard use as text message transferred between two human users, SMS has been used in several different ways. For instance, (Ficek, Pop, and Kencl 2013) presents a SMS-based active tracking in a GSM network. In this approach, the system forces the network to locate a mobile phone by sending it a SMS. Just in the opposite, (Bellini et al. 2010) proposes a solution where the mobile identifies the Cell-IDs where it is registered and send an SMS with this information to the tracking system in the server side. In this work (Liu et al. 2011) presents a review of mobile-health applications. It presents several examples of the use of SMS for providing access to information. The review is quite interesting because includes examples in developed countries such as Australia, Spain or United States of America.

The use of SMS as communication channel for transferring data (not only notifications) in mobile apps have also several good examples. In (Reinau, Harder, and Weber 2015) presents a new method for collecting data from GPS incorporated in the mobile device and transfer them to the server side via SMS to be used for trip management. Delgado and Martinez (Delgado and Martínez 2015) proposes the use of the mobile phone for collecting solar radiation and evaporation information (for being used in the agriculture domain) and transferring it to other users and systems

using several channels, including SMS. Mut-Puigserver et al (Mut-Puigserver et al. 2012) presents a review of electronic ticketing services applied to transport that includes approaches based on SMS. Langer et al (Langer et al. 2014) presents a system for providing search facilities about medicines over short message service. It has special focus on developing a natural language interface allowing the user to be flexible in phrasing their queries. Also in the use of SMS as a communication channel for apps, “Social.Water” is an open-source code for the deployment of a server that could allow the obtaining of environmental data via SMS. (Fienen and Lowry 2012) (‘Social.Water Source Code’ 2012).

The reliability of the SMS is considered one of the main key aspects for using them in several scenarios. For instance, most of the personal safety apps of the mobile markets use SMS as the main help request channel (<http://www.techhive.com/article/2057930/5-personal-safety-apps-that-watch-your-back.html>). In the work (Ferreira et al. 2013) presents an Android application designed for making easier the communication between an elder person and her/his caregiver. It uses SMS as one of the main communications channels. Even in a cloud data access control system oriented to the new generation of mobile networks, SMS are used for remote confirmation (Yan, Li, and Kantola 2015). This reliability could be one of the reasons because SMS had been also the base for the development of government services based on mobile devices (m-government) work presented by Al-Hujran in (Al-Hujran 2012). Even during the first wave of mobile apps, with a lot of solutions built over this applications and the smartphones, a SMS-based e-government had been considered a priority system for delivering e-government services in developing countries services in developing countries presented by

Susanto in (Susanto and Goodwin 2011). Several companies have in their business lines the development of SMS platforms for m-government (see <http://www.noveltech.gr> for instance). Nowadays, in developed countries such as United States there are numerous public administrations that offers SMS based services like weather information, (<http://www.weather.gov/subscribe>), hurricane-specific information (<http://www.weather.gov/subscribe-hurricaneinfo>), or tsunami-specific information (<http://www.tsunami.gov/sources.php>).

In other countries, like United Kingdom, there are services that allows the notification of emergencies by using only SMS (<http://www.emergencysms.org.uk>). In this way, and related with the interaction with the phone cells, in the work (Řezník, Horáková, and Szturc 2013) proposes an approach for the notification of emergence situations to citizens by sending a SMS directly from the phone cells that operate in the area. This approach does not need special software operating in the mobile phones because it is using the default SMS application installed in all devices. It exposes one of the main values of the SMS that is the existence of a default application for their management in all mobile phones.

SMS has been also used as base for the development of several mobile pay-systems as it is presented in a recent work by Hedman and Henningsson (Hedman and Henningsson 2015). In the other side, the use of SMS is identified in numerous works as a less relevant technology in mobile services. For instance, Zarpou et al (Zarpou, Drosopoulou, and Vlachopoulou 2013) presents a study of the best practice in tourism applications and technologies that take place in the mobile tourism sector over the analysis of fourteen tourism services. It asseverates that “the preferable way delivering the service is the mobile applications; mobile browsers,

SMS and MMS seem to be not widely used in this study's sample of examination". Even taken into account that this paper cited (Wan 2009) as "a tourism information system should provide personalized services, serve customers ANYTIME-ANYWHERE and improve the quality of service continuously", it does not take into account in the study proposed the possibility of not having high-speed data connection.

In a more ambitious use of SMS, they have been considered as an additional communication channels that can extend the Web to the mobile devices. In this way, Lombera et al (Lombera et al. 2013) presents the design of peers that form a mobile ad hoc network over SMS and Android phones. It describes a decentralized search and retrieval system that provides resistance against the vulnerabilities of centralized services. The SMS interface and its SMS-HTTP bridge enable SMS-capable mobile phones to communicate with and obtain information from HTTP nodes in the ad-hoc network. The system over SMS protocol enables mobile phones to communicate directly with each other and share information in a peer-to-peer fashion bypassing the Internet. In other approach, Wilde and Vaha-Sipila (Wilde and Vaha-Sipila 2010) presents a proposal for integrating SMS in the Web standardization processes by the specification of Uniform Resource Identifier (URI) scheme "sms" for specifying one or more recipients for an SMS message. SMS messages are two-way paging messages that can be sent from and received by a mobile phone or a suitably equipped networked device.

All these cases present ad-hoc solutions, not a general approach that could offer base technology for the development of new applications (the main objective of this paper). Of course, other alternatives for developing high-level communications

schemes over SMS have been developed. For instance, Constrained Application Protocol (CoAP) was introduced as a simpler alternative to the Hypertext Transfer Protocol (HTTP) for connecting constrained smart objects to the Web and offering an implementation over SMS. Nevertheless, the support over this channel can be considered as a draft expired according with the Internet Engineering Task Force (Poetsch et al. 2013).

In the same line than CoAP, several protocols oriented to M2M communications in the IoT domain exist, relying on TCP/IP or UDP/IP as transport layer. MQTT or Message Queue Telemetry Transport (Banks, A., Gupta 2014) was designed as a subscribe/publish messaging protocol for constrained devices, with an intermediate server acting as a broker. Devices publish data in a MQTT broker under a topic, and the broker forwards this data to other devices subscribed to that topic, but with no indication about the device publishing the data (other than the topic itself). MQTT was adopted as ISO standard in 2016 (ISO/IEC 2016).

XMPP or Extensible Messaging and Presence Protocol (Saint-Andre 2011) is another Device-to-Server protocol based on text/XML and designed initially for Instant Messaging applications, although it has been adopted also for M2M communications in scenarios where the data exchanged must be interpreted by humans and response times are not critical. In this case, the identification of the device sending the data is available as it is inherent to the IM applications.

AMQP or Advanced Messaging Queue Protocol (Vinoski 2006) is based on message queuing (despite the Q in MQTT protocol means “Queue”, it is not a queue-based protocol) and provide advanced functionalities such as queuing and transactions, which makes it ideal in scenarios with partial connectivity if time

constraints are not critical. Devices can include a client-id property in the messages when sending them, thus receivers may know the identity of the device publishing some data. AMQP was also adopted as ISO standard in 2014 (ISO/IEC\_2014 2014).

Finally, DDS or Data Distribution Service (OMG 2015) is a communication middleware which enables a common data space shared between devices (called Global Data Space), and every device has access to the information generated by other devices that it is interested in. It is a totally distributed system and with a low latency, which makes it appropriate for scenarios where time constraints are important. Device identification is not provided directly but the data published in any topic includes a key property that can be used to identify the device without introducing additional properties.

### **Development and improvement of the Communication's protocol**

The technology shown in this paper is based in a previous work (Asensio et al. 2014). This work proposes a protocol specifically designed to allow interoperability among *things* with different communications standards in an *Internet of the Things (IoT)* scenario. This implied the necessity of encoding operations by giving priority to minimize the bytes used for transmitting the information because of the heterogeneity of channel characteristics provided by the different communications standards that operate in these scenarios. This characteristic makes this approach very useful in the design of a solution for degraded and non-degraded communications environments (the objective of this paper). As the environment

considered in this scenario is not necessarily an IoT scenario, it has been required making some adjustments in the design and the implementation of the protocol in order to provide the requested capabilities.

The necessity of real time information in a LBS environment is directly related with the tracking of the devices involved. This implies the necessity of sending their location information as the base of the information transferred. Because the main effort of this work is to provide a standardized approach, the solution shouldn't have a restriction related with the identification of the type of device implied in the services. In this way, the server of the tracking system does not need to know what kind of devices (nodes of the system) are involved in the communication. The nodes of the system may be sensors, vehicles or even smartphones. In this context, "node" components refer to devices whose main function is to send and receive information to/from the server. Everyone must be able to communicate with the server, so the protocol must provide the abstraction level that give the system the capability for adapting the communications to different use cases. In addition, at low level it is necessary to provide specific implementations of the protocol according to the nature of the node.

Using a tracking system means adopting a centralized communication solution, where all messages are destined for or come from the server. We preferred to opt for this solution because one of the objectives of this work is to provide LBS in areas where Internet access is not guaranteed in at least 75% of the time, and to be able to make available on Internet the data returned by those services. Also this solution reduces the number of necessary messages (SMS) and minimizes the cost compared to decentralized solutions.



In the context of the present work, the communications standards supported for the different scenarios are SMS for DCE and HTTP for NDCE. As a first approach for encoding messages in SMS, the Open GeoSMS (Consortium 2012) standard was selected and extended to provide additional details. Nevertheless, this approach made the compatibility with the HTTP protocol more complex to support (especially in the evolution of the system).

Open GeoSMS is a text messaging format (SMS) used for the exchange of geospatial information in GPS and LBS applications and services. The Open GeoSMS protocol provides developers with communication between devices and/or applications to transmit localization data using text message encoding (SMS). Any application or service can send an SMS message using the Open GeoSMS format. In addition, any mobile phone has available SMS sending (no smartphone required).

Initially, it was evaluated the use of Open GeoSMS to transmit the data. Open GeoSMS could be used as the specification of the data format for the Protocol, for instance in the operation and in the event that send the location via SMS. But in many cases, the operations specified in our Protocol correspond with the transmission of sensor data. While it is a standard that might be helpful towards the project, it presents certain inconveniences such as the necessity of encoding URIs (Wilde and Vaha-Sipila 2010). This encoding involves the use of a large number of characters which, together with the limitation of 160 characters in text messages, makes that it will miss out on much of the available space that can be used to transmit as much information. Finally, the standard does not specify the format of the message response or confirmation from the server. For these reasons, it was decided to discard its use.

As a consequence, it was decided to replace the GeoSMS encoding with a JSON (<http://www.json.org>) based encoding with a specific restriction of using less than 160 characters (SMS restriction). JSON is the acronym for JavaScript Object notation. It is a lightweight format for data exchange. JSON is a subset of the literal notation of JavaScript objects that does not require the use of XML. One of the supposed advantages of JSON over XML as the data interchange format is that it is easier to write a parser and it is lighter. In the tracking system, the nodes communicate with the server using this format, and vice versa.

Another possible alternative is the use of GeoJSON (Butler et al. 2016), GeoJSON is a format for encoding a variety of geographic data structures. A GeoJSON object may represent a geometry, a feature, or a collection of features. Features in GeoJSON contain a geometry object and additional properties, and a feature collection represents a list of features.

Our protocol is very similar, but not equal to GeoJSON. Initially, the functionality that GeoJSON offers to define features could be enough. Nevertheless, it has been necessary to add the definition of a set of operations to ensure the correct functioning of the tracking system. These operations correspond with the services that configure, reset, or start the system nodes.

Other adjustment to the protocol that was necessary to make is related with the auto-identification of the nodes. In a general IoT approach, the devices involved used could use static IP that play the role of identifiers. Another option could be MAC address as identifiers of the devices. Nevertheless, the inclusion of smartphones as nodes of the system put the focus of the origin/end of the information on the person that is making the communication. For this reason, it has been

necessary to provide a solution that could support the change of the device (smartphone) that is taking part in the process. The approach adopted is oriented to the identification of the nodes in the local context of a specific service. In DCE, the phone number is used for the identification of the device: this information is provided by default by the SMS protocol. In NDCE a first approach would be the use of the IP of the device. Nevertheless, in most of the cases the communication channel (3G, WiFi, etc.) is providing a dynamic IP that could be modified in case of part-time loss of coverage. For this reason, the implementation of the operations under HTTP had to provide a local identification of the node in each operation. This solution was extended to the SMS solution in order to give the system the opportunity of automatic reconfiguration of the communication channel without loss of “conversations”.

Finally, when the communication has to be initiated by the node, the protocol could operate without problems because every node knows the identification of the server (both phone number for SMS, and IP for HTTP). Nevertheless, if the server has to initiate the communication process, it could know the identification for the SMS standard (phone number) but there is not the same situation for HTTP. In this case, because the possibility of having nodes without static IP (smartphones for instance), it was necessary to extend the protocol by giving support to the communication in this way using Push notifications (Bell, Bleau, and Davey 2011). The message in JSON format containing the operation to execute is sent inside Push notifications. This message may even include the server request to open a HTTP connection with the server and then use this channel without restrictions. The implementation presented in this paper use Google Cloud Messaging (GCM)

(Google 2014) as the infrastructure for Push notifications. The GCM service is responsible for all aspects of keeping the message queue and sending them to the application that runs on the node.

Figure 3 shows an overview of the operations that can be performed over the infrastructure taking into account the communication standard that could support them. The operations have been separated in two groups: right side includes the operations offered to a client application that will offer final functionality to users avoiding details of the communications channel used; left side level that presents the operations that use the sever for connecting to the nodes and solve the requests coming from the client.

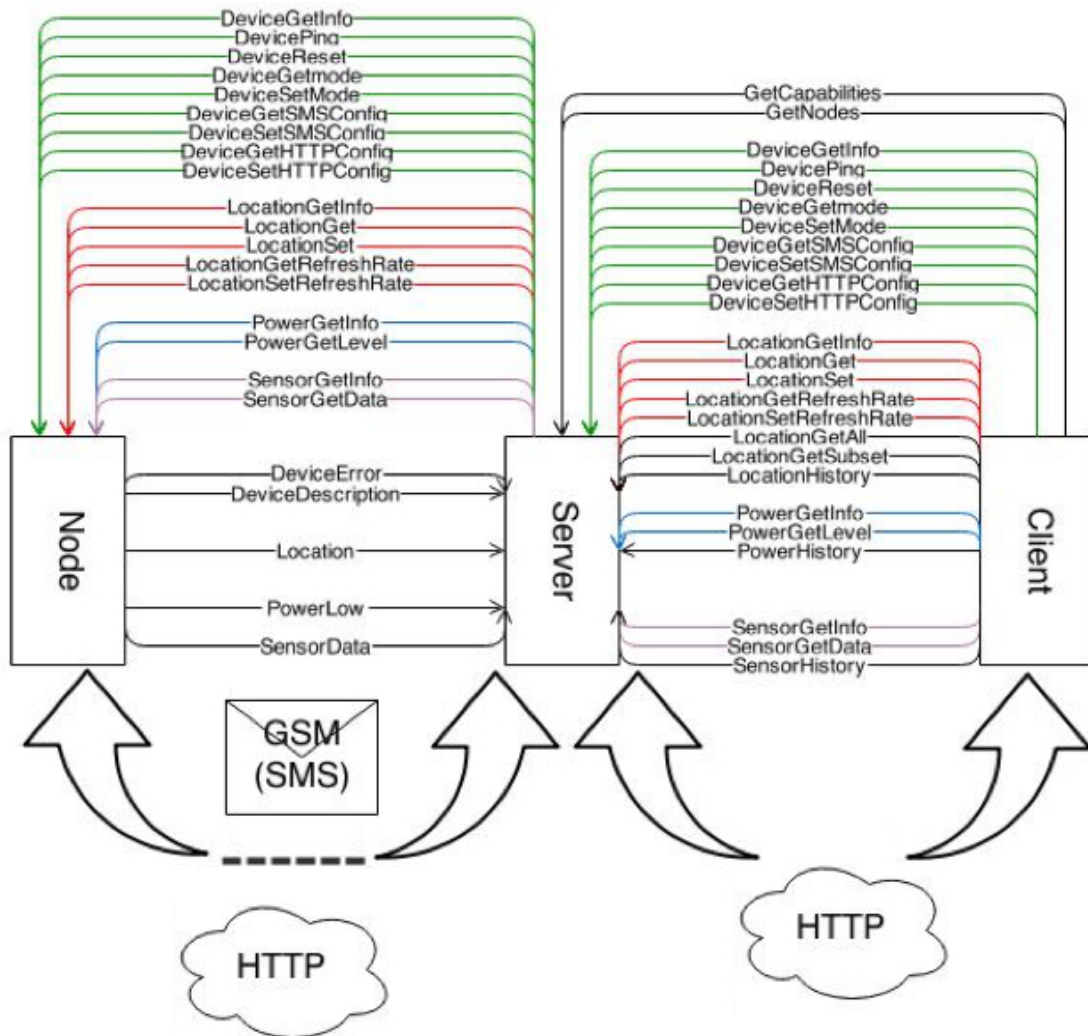


Figure 3: System Operations diagram

When considering Android devices as nodes in the protocol, it is possible to use the additional set of sensors that they incorporate for obtaining more information from the environment. For this reason, it was decided to extend the protocol in order to incorporate new operations for being able to transfer the data obtained by these sensors. This approach could be also used in case of integrating other devices that could offer some level of “intelligence”.

### ***Protocol structure***

Communication Things Protocol (CTP) (Asensio et al. 2014) has been taken as starting point in order to define this new protocol. CTP is a binary protocol optimized for low power sensors which aims to provide a specification that allows interoperability among communication standards and coexistence with other IoT protocols, but prioritizing simplicity, efficiency and functionality, to build final IoT systems. CTP considers a “thing” device as a set of independent entities called EndPoints, which aggregate specific functionalities. Currently, CTP has defined five EndPoint categories: Base, Communication, Sensor, Actuator and HMI. Each one of these EndPoints could supports several sets of commands and events called Clusters, which define capabilities and functionality of the device. For example, there are defined six clusters for the EndPoint Base: DEVICE, LOCATION, POWER, TIME, PROXY and BEHAVIOR, and cluster POWER defines four commands: POWER\_GET\_INFO\_REQ, POWER\_LEVEL\_GET\_REQ, POWER\_GET\_CONFIG\_REQ and POWER\_SET\_CONFIG\_REQ).

In the definition of the CPT implementation in JSON for SMS, that CTP structure was strongly simplified. Endpoint addressing disappears, and clusters and commands are joined in a new command field encoded in ASCII. Even these changes, the work made in the taxonomy and identification of a “thing” device has been maintained. Also, thanks to using the ASCII encoding, it has increases humans usability of the protocol. It should be noted that current JSON version has been focused only in some clusters (mainly referred to sensor and location capabilities) and it is not a full CTP implementation but it could be extended in future versions.

Focusing in the JSON implementation, we find three kinds of messages:

- *Commands* are usually action requests to a device (of the same or different thing) that should send back a response notifying about the action result.
- *Responses* send back the information requested by a command.
- *Events* are messages sent by the device without a previous command request (e.g. a low battery event)

All Commands, Responses and Events defined are JSON objects composed by four fields (some of them are optional) as it is shown in Table 1. Any additional field required by the message in order to command the device or send the requested information is included in the parameters vector field.

Field name	Mandatory/Optional	Description
CMD	M	Request ID
p	O	Parameters (p) is a vector with additional information about the request or the response
h	M	Message hash
s	O	Request sequence identifier

Table 1: JSON message structure

For example, a ping message to a device is encoded in JSON Format as {"CMD": "DevicePingReq", "h": "XX"} where XX is a number encoded in hexadecimal format. In the binary version of CTP, this message is included into the cluster Device and is called "DEVICE\_PING\_REQ". Similarly, the device generates the next response {"CMD": "DeviceACKRes", "p": [{"DevicePingReq"}], "h": "XX"}. These messages used in SMS communication have been easily translated to an HTTP request, where IP communication is available (see Table 2).

JSON	http request
------	--------------

{"CMD": "DevicePingReq", "h": "XX"}	http://<IPaddress>/?CMD=DevicePingReq&h=XX
{"CMD": " DeviceACKRes", "p":[" DevicePingReq"], "h": "XX"}	http://<domainName>/?CMD=DeviceACKRes&commandSource=DevicePingReq &h=XX

Table 2: JSON messages vs HTTP Requests

### **ANDROID technological implementation**

This section presents an overview of the architecture of the developed frameworks for the Android App. Figure 4 shows the main components and their relationships in the Protocol framework and the sensors framework. They are: the server of the tracking system, the protocol framework and the sensors framework. While the Tracking System Server is not implemented in the Android App, it has been included in the diagram to show the protocol framework communicating with that server.

- Online and offline communications: is the component responsible for communicating with the server of the tracking system to receive their orders and to send results and events. This component implements two types of communications, HTTP for NDCE and SMS for DCE.
- Events and operations: this component is responsible for executing the operations defined by the protocol for a node, as well as events that are regularly sent to the server.



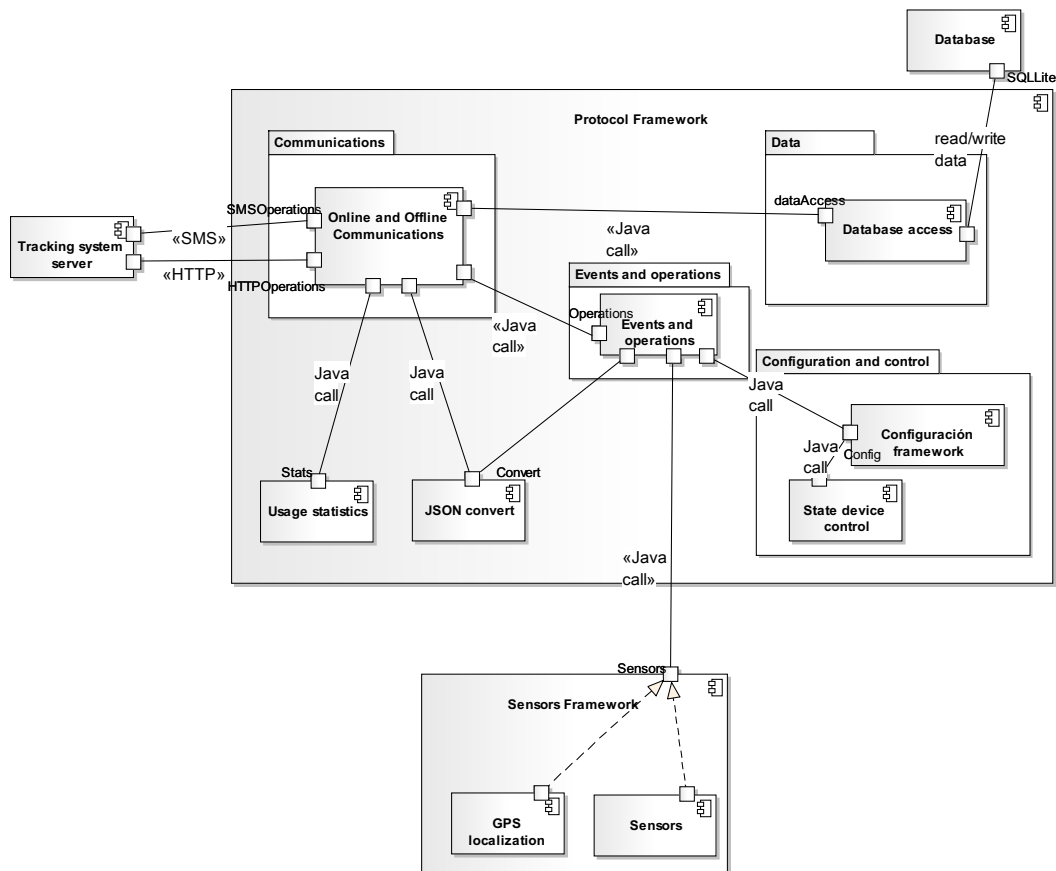


Figure 4: Android App framework components diagram

In the Protocol framework the principal components are:

- Database access: this component is responsible for facilitating the inclusion and deletion of data used by the framework in the database. In this work, SQLite (Owens and Allen 2010) has been chosen since it is the most extended database manager in the development of mobile apps.
- Configuration framework: its function is to store and to give access to the configuration parameters for the framework, as for example the IP address of the server or the power mode (see Table 3). This allows managing the power consumption of the nodes, improving the performance of the system.

- Device state controller: its main function is to monitor network related status of the device (connecting them, disconnect them, whether it has 3 G connection...). In this way, it also optimizes the energy consumed by devices.

Table 3 shows the different energy related operation modes in which a node can be established. These modes are defined in the protocol specification.

	<b>Operation Mode</b>	<b>Configuration</b>
0	WAIT: The device is waiting to be configured.	GSM: ON / GPS: OFF
1	RUN 1: Extreme Low Power mode	smsPollTime: 86.400.000 ms (once at day) smsTransmitPeriod: 86.400.000 ms (once at day) locationRefreshRate: 86.400.000 ms (once at day)
2	RUN 2: Very Low Power mode	smsPollTime: 86.400.000 ms (once at day) smsTransmitPeriod: 86.400.000 ms (once at day) locationRefreshRate: 900.000 ms (15 min.)
3	RUN 3: Low Power mode	smsPollTime: 900.000 ms (15 min.) smsTransmitPeriod: 900.000 ms (15 min.) locationRefreshRate: 900.000 ms (15 min.)
4	RUN 4: Always Locate mode	GSM: ON / GPS: Always Locate Mode  smsTransmitPeriod: 60.000 ms (1 min.) locationRefreshRate: 60.000 ms (1 min.)
5	RUN 5: Custom	smsPollTime: Adjusted by command smsTransmitPeriod: Adjusted by command locationRefreshRate: Adjusted by command

Table 3: Power Modes

Figure 5 shows the class diagram of the Framework Protocol. The principal modules are:

- Message: it is the module responsible for receiving, sending and reading messages. Once a message has been received, it should be analysed to know which operation of the module "Operations" should be executed.
- Operations: this module implements the operations of a node, as it specified in the Protocol. This is a class that acts as a facade for invoking the

corresponding operation. Each of them has been implemented as a class, with an execute() method that performs the operation and returns a String encoded in JSON with the results.

- Events: this is the module that implements and controls the events of a specified node in the Protocol. The EventController class is an Android service that provides operations to invoke events, and that controls periodic execution of them. A separate class represents each one of the events managed.
- ModeController: this module is responsible for switching on/off the mobile network periodically depending on the energy configuration stored in the class SharedPreferences.

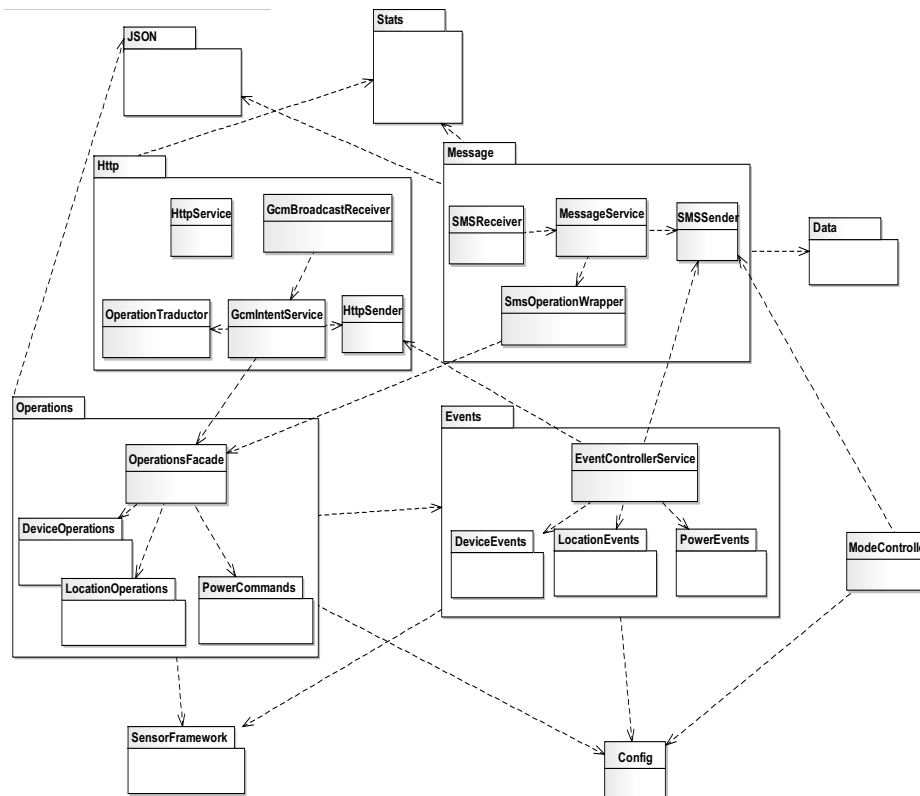


Figure 5: Protocol framework structure

In the sensors framework there are two components, the device sensors and GPS location:

- GPS location: it is responsible of reading the device location using its GPS, also be allowed to choose a refresh time to save energy. GPS has been managed as a separate module from the other sensors since it is the base for supplying LBS services.
- Device sensors: it is the component in charge of reading the other device sensors.

### **SERVER side technological implementation**

Figure 6 shows the Server General Scheme. "Node" components communicate with the server through the HTTP protocol in the case of NDCE and via SMS messages in DCE. On the other hand, "Client" refers to a software component that is able to invoke operations on the server. Regardless of the communication channel used, all the actors of the system must use the specification of the communication protocol to encode operations.

In DCE, two options can be used to communicate between server and mobile devices: a SMS Gateway, provided by an external company, which allows sending and receiving SMS to/from nodes. The External company provides a web services API, that allows the server of our system communicate via HTTP (SMS Gateway in Figure 6). The second option is to deploy an App for a mobile device, that also

allows sending and receiving SMS to/from nodes, but communication with the server is done in HTTP (Mobile App in Figure 6).

The "Framework" is composed of several components:

- HTTP server: handles the communications with the rest of the system, i.e. send and receive orders and information to/from the nodes and external customers.
- The database manager: it handles the appropriate operations of reading and writing the database according to the operations requested by the nodes or customers.

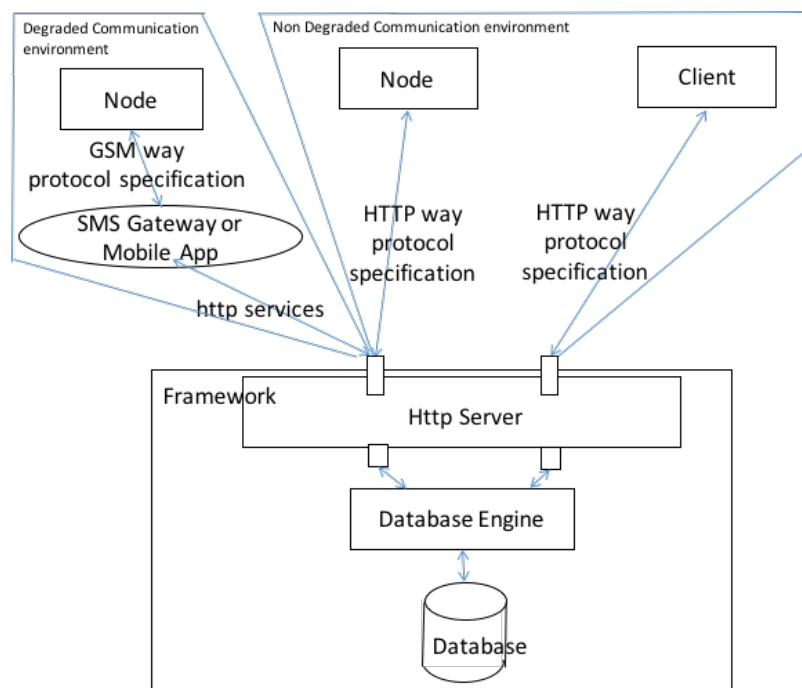


Figure 6: Server components

Figure 7 shows the server's data classes diagram. The class **Node** represents a node of the system with the corresponding attributes and methods. The class **Measurement** is an abstract class used to encapsulate any type of measure in a node, such as location (**Location**), energy (**Power**) or information from a sensor

(SensorData). The class Sensor represents possible sensors associated with each of the nodes. At the same time, the sensors have associated measurements of the SensorData class.

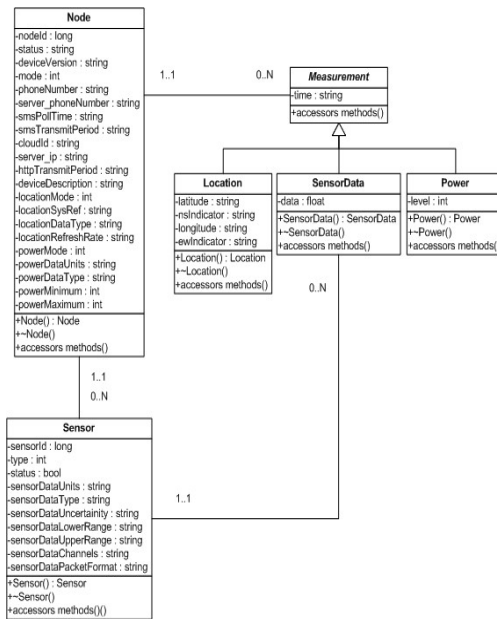


Figure 7: Server structure

The implementation has been done over the Spring IO framework (Pollac, Mark Gierke, Oliver Risberg, Thomas Brisbin, Jon Hunger 2012) ('Spring Documentation, Guide and Specification' 2016), which is based on Java technology. Figure 8 shows the Spring components which comprise the system. There is a Service Layer with several Controllers, which are the equivalent to the Java servlets, and a Persistence Layer, where there are located the Repositories, responsible for database access.

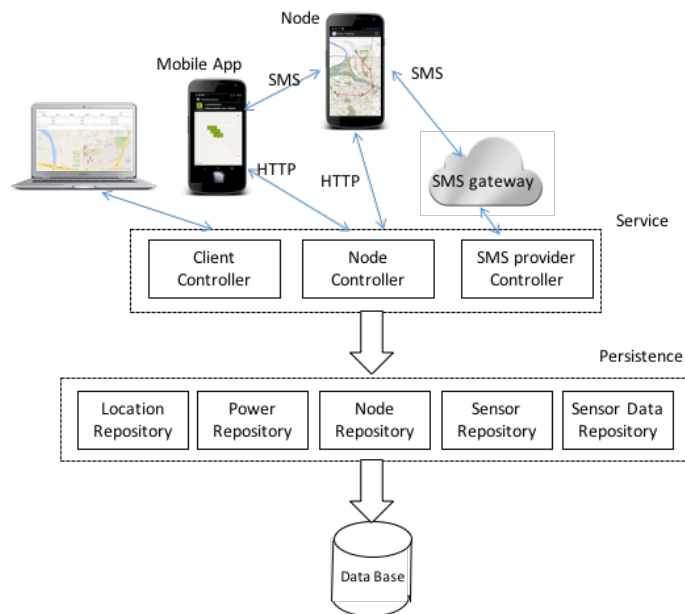


Figure 8: Server implementation over Spring IO

In the persistence layer, a Java class is created for each one of the objects for implementing the data model. Once the domain classes have been created, it is used Hibernate ('Hibernate' 2016) to persist the entities. Hibernate is responsible for performing the ORM (Object/Relational Mapping) mapping to transform classes to database tables.

Once the persistence layer is implemented, the next step is to implement the service layer. This layer consists of several web services that can be accessed to perform appropriate operations of the communication protocol in the server. In Spring, this is translated to a Controller which defines the accessible resources and implements its functionality. A first Controller is in charge of serving external clients' operations; a second, the Node Controller, for the operations of the nodes; and a third controller serves as an intermediary between the messages arrived from the SMS provider and the Node Controller. In this third Controller, when a text message arrives, it is translated from JSON to an HTTP request, and this request is sent to the controller of the nodes. This way, the receipt of SMS messages is

abstracted, so if the SMS provider is changed, it would be only need to create a new resource to receive the messages, and the translation process would be the same, without affecting the rest of the system.

To extend the functionality of the framework to NDCE environments, it is evident the need of communicating with mobile nodes whose IP address is not always known (unlike GSM where the number phone is always the same).

Therefore, it was decided to use the Google Cloud Messaging (GCM) service to send Push notifications (Bell, Bleau, and Davey 2011) to Android nodes. This service allows sending and receiving messages between the server and the nodes without knowing the IP addresses of the nodes, which may change over the time. The Push Notifications service associates with each node a unique identification ID, in such a way that the server sends messages to the GCM service with the node ID.

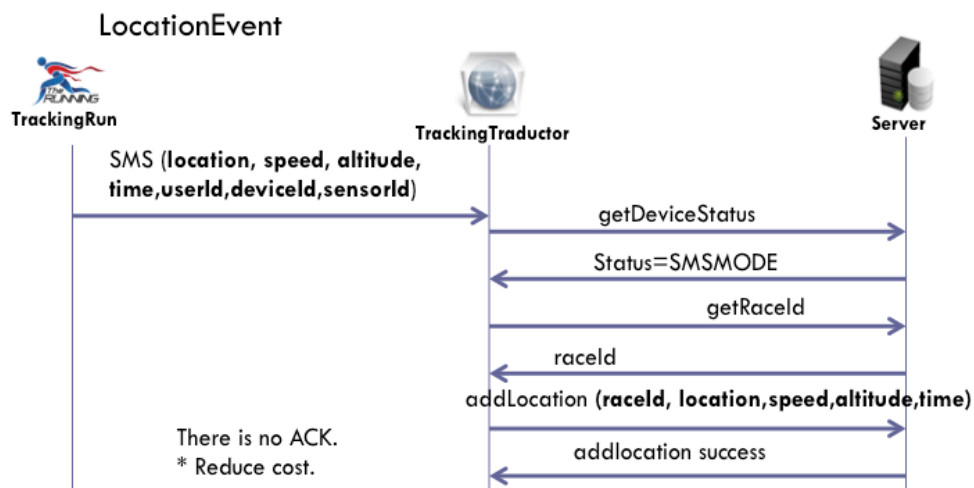


Figure 9: Mobile communication with the tracking system: conversion of SMS in Web services

The tracking system functionality in DCE is based on sending and receiving text messages. To choose the infrastructure that would support such functionality, we consider the following alternatives:



- Using an SMS Gateway. This option is available in companies that offer solutions to send and receive SMS through a SMS gateway, and the integration is performed using an API.
- To develop a Mobile App to translate the SMS to HTTP requests to the server (and vice versa). The SMS are sent to the mobile number of the intermediate node, which executes an App that interpret and translate the SMS to an HTTP request and forward that information to the server.

In a first phase of development it was decided to use the SMS Gateway, but this method had some problems: it was unstable due to the enormous number of SMS that the system needs to send and receive, and it was not possible to know with certainty the phone number used by the SMS Gateway company, which turns into a problem when sending SMS messages from remote nodes to the server.

Thus, it was decided to implement an App for mobile devices that integrate the Framework Protocol and perform an intermediate role between the server and the nodes. "TrackingTraductor" is the App that enables communication between end nodes and the server when the former are in DCE. The App must run in a mobile phone with internet connection (see Figure 8). This is the communication system used for DCE. The node sends the SMS message to the phone that is running the App "TrackingTraductor", and this App transforms the SMS into a web service that is sent to the server (see Figure 9 that presents the conversion for the scenario presented later).

## **Use case specification and construction**

In order to act as a proof of concept, this section describes the construction of an example that makes use of the libraries presented in the previous sections, applied to the tracking of sporting events.

The App "TrackingRun" can monitor in real time the participants in sporting events in DCE. Once logged in "TrackingRun", either via Facebook, Google Plus or through our server, the user will select a training mode ("Walk", "Running" and "Cycling") and may start storing data such as the speed, distance, elapsed time, maximum and minimum altitudes reached and the route taken through the positions obtained by the GPS. "TrackingRun" communicates with the server using the "Location based offline services via SMS" Protocol. In NDCE it uses http requests. In DCE communicates via SMS that it sends to the phone that is running the App "TrackingTraductor". To enable the location in an area without 3G coverage, the app imported the Frameworks Protocol and Sensors through SensorLib and NodeLib libraries (see Figure 10), which basically carry the modules: "http", "message", "events" and "operations" ( see Figure 5) whereby you can send operations from server to nodes and events from nodes to server.

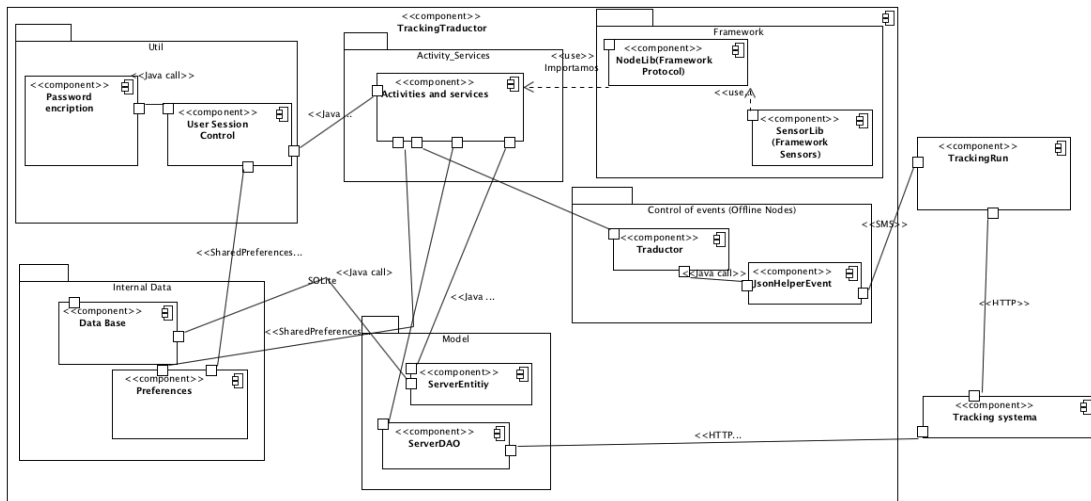


Figure 10: Tracking Run Component Diagram

To accommodate our needs, they have also implemented new events and modified some others. Events in SMS format for sending information over the Offline Mode races and their corresponding fields are:

- DeviceDescriptionEvent: Android device unique id, user id, email, password (encrypted with an AES block cipher with a 128-bit key) and the type of record. The information contained in this event serves to register the device in the race.
- StartRaceEvent: instant start-up time, race mode ("Walk", "Running" or "Cycling"), author of the race, weight and height.
- LocationEvent: latitude, longitude, time point of measurement.
- EndRaceEvent: completion time instant and additional data.

These SMS are sent to the mobile number of the intermediate node executing the "TrackingTractor". In order to make the validation of the developments, a number of tests were conducted with nodes in different zones of the province of Zaragoza, Spain.

Three kind of tests have been performed to verify application behavior:

- Online tests: With data roaming enabled, trainings were carried with 3G coverage, and it was verified that the data stored on the server (and visible on the phone) were correct.
- Simulated offline Tests: With data roaming off and offline mode enabled, an environment without 3G coverage was simulated and careers were performed that had expected behavior. It was verified, that both the SMS in the intermediate app and the server were the correct
- Real offline Tests: Even with the data roaming enabled, we had no 3G coverage, so to communicate it was obligatory to activate the offline mode.

After consulting through Opensignal (Opensignal 2016), places near Zaragoza with little coverage we moved to one of them to carry out more real offline tests. The images of Figure 11 shown on the left side the App "TrackingRun" and on the right the "TrackingTraductor" App.

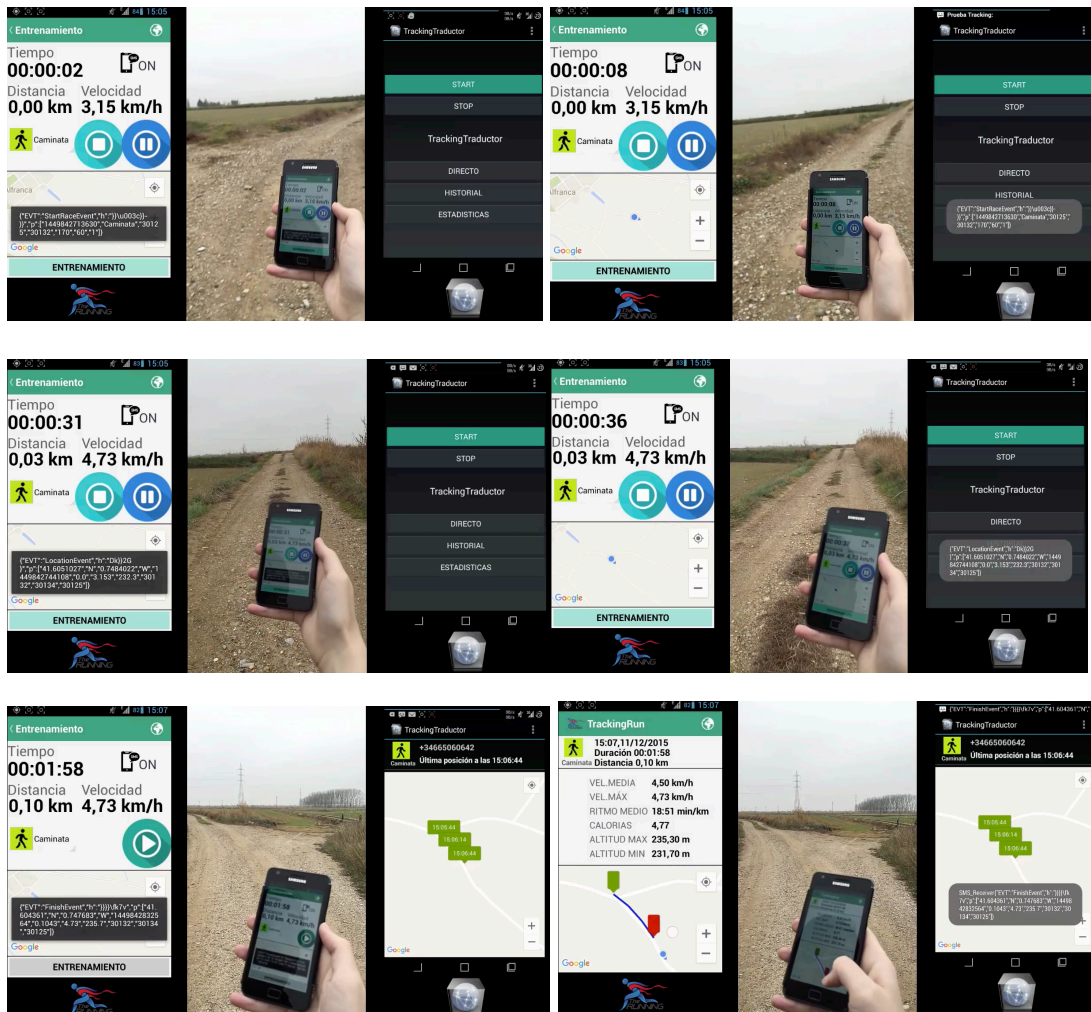


Figure 11: TrackingRun and TrackingTraductor Apps

The first line of images shows the sending of a "StartRaceEvent" by the "TrackingRun" App, and its reception by the "TrackingTraductor" App. The second line of images shows the sending and receiving of a "Location" event. The third line of images shows the sending and receiving of an "EndRaceEvent". This test was performed in the coordinates (41.605379, -0.750768), an area closes to "la Alfranca" in the province of Zaragoza, Spain. The full video of the experiment can be seen at <https://youtu.be/ozaGEzfO3G8>.

## **Openness of the technology**

As it has been mentioned in the introduction, this paper has also the objective of provide open access to the technology developed in order to give opportunities for its reuse by third parties without license cost.

LBSnon3G library is released under the 3-clause BSD license ('OpenSource BSD-3-Clause' 2016), so it may be used for commercial purposes. The license does not require anyone to cite LBSnon3G, but if you use LBSnon3G in a published research work we encourage you to cite this article. In addition, it has also been included an example of use them that could help developers for building their own solutions. The software is available at Github (Tom Preston-Werner PJ Hyett 2013) repository [https://github.com/IAAA-Lab/LBS\\_non3G/](https://github.com/IAAA-Lab/LBS_non3G/)

## **Conclusions**

In the XXI Century, when new technologies (5G) provide the capacity for support mobile communications with download speeds of up to 35 gigabits per second (Qualcomm 2017) there are large areas in the World (even in the developed countries) where it is not possible to have more than 75% of the time access to mobile connections that around 200 kilobits per second (minimum 3G speed). These are the DCEs, where people live, work or spend their free time. They need (and even they have the right) the provision of digital services that could help them. This paper has presented an approach to work on the way of providing support for developing solutions to these needs. This approach includes a set of software utilities (based on libraries and examples of use) that can be used by software developers for creating

their solutions over the base of a proven experience. This is the main difference of this approach in relation to other technological solutions (many of them available at industrial level): is its conceptualization as platform for building solutions, trying to avoid the most usual approaches based on ad-hoc solutions that have a lot of maintenance problems, as well as they do not used to reuse past experience for building better and more robust solutions.

Working over the experience of a previous work in developing light protocols that do not need broadband for communication, this paper has presented an approach that extends a validated protocol for IoT environments by including mobile devices as elements of the communication process. This solution makes use of the capability for operating with the SMS that these mobile devices provide. In this way, the libraries developed can identify which communication channel can be operated and then decide automatically the adjustment of the protocol for using 3G (or upper) or SMS.

The final contributions of this paper include the design and the corresponding source code that implements a protocol that can manage transparently the use of different network protocols for server side applications (based on Java and Web browsers) and for mobile device platforms with Android operating system. This source code is now available in a Git repository for free use (without any charge). In addition, the paper has presented a detailed design of the software utilities built in order to give developers the possibility of an easier modification of them according with their own necessities. This has been validated in the example presented. In addition, it has also been used in two additional systems currently under development with the collaboration of two software companies.

One additional advantage found with this approach in order to be able to provide services in DCEs is that the SMS protocol is asynchronous. When a software application running in the server side needs to send a message to a mobile application that is out of a 3G area, it has only to send the corresponding SMSs and to check if they have been received by the communication network. After that, all the responsibility of the communications is delegated to a proven third party infrastructure (in general a commercial infrastructure) that will check if it is possible to transfer the message to the mobile device or not. In case that it would be not available (out of coverage), the network infrastructure will buffer the information till the device would be accessible. This makes easier the task for building the software because avoid the necessity of checking the availability or not of the mobile device.

This work has created a solid foundation to continue working on the creation of applications that make use of the software utilities developed in any environment where a tracking system is needed. There are many application contexts in which it is possible to make use of the developed technology: companies of distribution of merchandise, monitoring and management of fleets of vehicles, individual navigation or monitoring of geolocation in movable objects. All these developments will improve the services and the people standard of living across the world in the era of the Digital Earth.

## **ADKNOWLEDMENTS**

This work has been partially supported by the Aragon regional government (project GCP-2016-0035-00, program PDR cofounded by FEDER from EU) and the Spanish national government (project RTC-2016-4790-2, program Retos Colaboración).



## References

- Al-Hujran, Omar. 2012. 'Toward the Utilization of M-Government Services in Developing Countries: A Qualitative Investigation'. *International Journal of Business and Social Science* 3 (5): 155–60.
- Alam, Mahbul. 2014. 'Simple Low Cost Online Monitoring System for Municipal Waste Collection Authority of under Developing Countries'. *International Journal of Electronic Governance* 7 (1). Inderscience Publishers: 72–80.
- Ali, Anum, Ghalib A. Shah, and Junaid Arshad. 2016. 'Energy Efficient Techniques for M2M Communication: A Survey'. *Journal of Network and Computer Applications*. doi:10.1016/j.jnca.2016.04.002.
- Asensio, Angel, Alvaro Marco, Rubén Blasco, and Roberto Casas. 2014. 'Protocol and Architecture to Bring Things into Internet of Things'. *International Journal of Distributed Sensor Networks* 2014. doi:10.1155/2014/158252.
- Banks, A., Gupta, R. 2014. 'MQTT Version 3.1. 1. OASIS Standard.' <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>.
- Bell, Kris M, Darryl N Bleau, and Jeffrey T Davey. 2011. 'Push Notification Service'. Google Patents.
- Bellini, Alexandre, Carlos E Cirilo, Vinícius R T Ferraz, Josué G Araujo, Juliana L Duque, Luana P Annibal, Rafael S Durelli, and Cesar Marcondes. 2010. 'A Low Cost Positioning and Visualization System Using Smartphones for Emergency Ambulance Service'. In *Proceedings of the 2010 ICSE Workshop on Software Engineering in Health Care*, 12–18.
- Bhattacharya, Kamal. 2015. 'From Giant Robots to Mobile Money Platforms: The Rise of ICT Services in Developing Countries'. *IEEE Internet Computing* 19 (5). Los Alamitos, CA, USA: IEEE Computer Society: 82–85. doi:doi.ieeecomputersociety.org/10.1109/MIC.2015.99.
- Butler, H., M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub. 2016. 'The GeoJSON Format'. *Internet Engineering Task Force (IETF)*. doi:10.17487/RFC7946.

- Chou, Hsuan-Yi, and Nai-Hwa Lien. 2014. 'Effects of SMS Teaser Ads on Product Curiosity'. *International Journal of Mobile Communications* 12 (4). Inderscience Publishers Ltd: 328–45.
- Chutijirawong, Narain, and Prasert Kanawattanachai. 2014. 'The Role and Impact of Context--Driven Personalisation Technology on Customer Acceptance of Advertising via Short Message Service (SMS)'. *International Journal of Mobile Communications* 12 (6). Inderscience Publishers: 578–602.
- Comisión Nacional del Mercado de las Telecomunicaciones. 2005. 'Comisión Nacional Del Mercado de Las Telecomunicaciones'.
- Consortium, Open Geospatial. 2012. 'Open GeoSMS Standard - Core'. [https://portal.opengeospatial.org/files/?artifact\\_id=44146](https://portal.opengeospatial.org/files/?artifact_id=44146).
- Delgado, Bueno, and Molina Martínez. 2015. 'Software Application for Calculating Solar Radiation and Equivalent Evaporation in Mobile Devices'. *Agricultural Water Management* 151. Elsevier: 30–36.
- E-UTRA. 2016. 'TS 36.300 v13.4.0 Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2", 2016',.
- ETSI. 2005. 'European Digital Cellular Telecommunications System (Phase 2); Point-to-Point (PP) Short Message Service (Sms)',.
- Ferreira, Fábio, Flávio Dias, João Braz, Ricardo Santos, Roberto Nascimento, Carlos Ferreira, and Ricardo Martinho. 2013. 'Protege: A Mobile Health Application for the Elder-Caregiver Monitoring Paradigm'. *Procedia Technology* 9. Elsevier: 1361–71.
- Ficek, Michal, Tomáš Pop, and Lukáš Kencl. 2013. 'Active Tracking in Mobile Networks: An in-Depth View'. *Computer Networks* 57 (9). Elsevier: 1936–54.
- Fienen, Michael N, and Christopher S Lowry. 2012. 'Social. Water. A Crowdsourcing Tool for Environmental Data Acquisition'. *Computers & Geosciences* 49. Elsevier: 164–69.
- Google. 2014. 'Google Cloud Messaging'. <https://developers.google.com/cloud->

messaging/.

- Hedman, Jonas, and Stefan Henningsson. 2015. 'The New Normal: Market Cooperation in the Mobile Payments Ecosystem'. *Electronic Commerce Research and Applications*. Elsevier.
- 'Hibernate'. 2016. <http://hibernate.org>.
- Hu, D, F Sun, L Tu, and B Huang. 2013. 'We Know What You Are--A User Classification Based on Mobile Data'. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, 1282–89. doi:10.1109/GreenCom-iThings-CPSCom.2013.223.
- Ingenu, Ingenu. 2017. 'RPMA Technology'. Accessed April 1. <http://www.ingenu.com/technology/rpma/>.
- ISO/IEC\_2014. 2014. 'ISO/IEC 19464:2014 Information Technology – Advanced Message Queuing Protocol (AMQP) v1.0 Specification'. <https://www.iso.org/standard/64955.html>.
- ISO/IEC\_2016. 2016. 'ISO/IEC 20922:2016 Information Technology – Message Queuing Telemetry Transport (MQTT) v3.1'. <https://www.iso.org/standard/69466.html>.
- Kala Fleming, Peninah Waweru, Muuo Wambua, Elizabeth Ondula, Lianna Samuel, undefined, undefined, undefined, and undefined. 2016. 'Toward Quantified Small-Scale Farms in Africa'. *IEEE Internet Computing* 20 (3). Los Alamitos, CA, USA: IEEE Computer Society: 63–67. doi:doi.ieeecomputersociety.org/10.1109/MIC.2016.58.
- Karippacheril, Tina George, Fatemeh Nikayin, Mark De Reuver, and Harry Bouwman. 2013. 'Serving the Poor: Multisided Mobile Service Platforms, Openness, Competition, Collaboration and the Struggle for Leadership'. *Telecommunications Policy* 37 (1). Elsevier: 24–34.
- Langer, Akhil, Rohit Banga, Ankush Mittal, L Venkata Subramaniam, and Parikshit Sondhi. 2014. 'A Text Based Drug Query System for Mobile Phones'.

- International Journal of Mobile Communications* 12 (4). Inderscience Publishers: 411–29.
- Liu, Chang, Qing Zhu, Kenneth A Holroyd, and Elizabeth K Seng. 2011. ‘Status and Trends of Mobile-Health Applications for iOS Devices: A Developer’s Perspective’. *Journal of Systems and Software* 84 (11). Elsevier: 2022–33.
- Lombera, Isaí Michel, Louise E Moser, P M Melliar-Smith, and Yung-Ting Chuang. 2013. ‘Mobile Decentralized Search and Retrieval Using SMS and HTTP’. *Mobile Networks and Applications* 18 (1). Springer: 22–41.
- Lora Aliance. 2016. ‘Lora Aliance’. <https://www.lora-alliance.org>.
- Mut-Puigserver, MacIà, M. Magdalena Payeras-Capellà, Josep Lluís Ferrer-Gomila, Arnau Vives-Guasch, and Jordi Castellà-Roca. 2012. ‘A Survey of Electronic Ticketing Applied to Transport’. *Computers and Security* 31 (8): 925–39. doi:10.1016/j.cose.2012.07.004.
- OMG. 2015. ‘Data Distribution Service for Real-Time Systems Specification, 1.4 Edition’. <http://www.omg.org/spec/DDS/1.4/>.
- Opensignal. 2016. ‘Opensignal’. <https://opensignal.com>.
- ‘Opensource BSD-3-Clause’. 2016. <https://opensource.org/licenses/BSD-3-Clause>.
- Owens, Mike, and Grant Allen. 2010. *SQLite*. Springer.
- Poetsch, Thomas, Kepeng Li, Markus Becker, and Koojana Kuladinithi. 2013. ‘Transport of CoAP over SMS’. *Transport*.
- Pollac, Mark Gierke, Oliver Risberg, Thomas Brisbin, Jon Hunger, Michael. 2012. *Spring Data Modern Data Access for Enterprise Java*. O’Reilly Media.
- Qualcomm. 2017. ‘Qualcomm Showcases 5G Leadership by Announcing Its First 5G Modem Solution’. <https://www.qualcomm.com/news/releases/2016/10/17/qualcomm-showcases-5g-leadership-announcing-its-first-5g-modem-solution>.
- Ratasuk, Rapeepat, Nitin Mangalvedhe, Amitava Ghosh, and Benny Vejlgaard. 2014. ‘Narrowband LTE-M System for M2M Communication’. In *IEEE Vehicular Technology Conference*. doi:10.1109/VTCFall.2014.6966070.

- Ratasuk, Rapeepat, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. 2016. 'NB-IoT System for M2M Communication'. In *2016 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2016*, 428–32. doi:10.1109/WCNCW.2016.7552737.
- Reinau, Kristian Hegner, Henrik Harder, and Michael Weber. 2015. 'The SMS--GPS-Trip Method: A New Method for Collecting Trip Information in Travel Behavior Research'. *Telecommunications Policy* 39 (3). Elsevier: 363–73.
- Řezník, Tomáš, Bronislava Horáková, and Roman Szturc. 2013. 'Advanced Methods of Cell Phone Localization for Crisis and Emergency Management Applications'. *International Journal of Digital Earth* 8947 (May): 1–14. doi:10.1080/17538947.2013.860197.
- Saint-Andre. 2011. 'Extensible Messaging and Presence Protocol (XMPP): Core.' <https://tools.ietf.org/html/rfc6120.html>.
- 'Social.Water Source Code'. 2012. <https://github.com/mnfienen-usgs/Social.Water>.
- 'Spring Documentation, Guide and Specification'. 2016. <http://spring.io/>.
- Susanto, Tony Dwi, and Robert Goodwin. 2011. 'User Acceptance of SMS-Based E-government Services'. In *Electronic Government*, 75–87. Springer.
- Tom Preston-Werner PJ Hyett, Chris Wanstrath. 2013. 'GitHub'. 2008. <https://github.com>.
- Vinoski, Steve. 2006. 'Advanced Message Queuing Protocol'. *IEEE Internet Computing* 10 (6): 87–89. doi:10.1109/MIC.2006.116.
- Walter, Kai. 2010. 'Development of an Early Warning Information Infrastructure Using Spatial Web Services Technology'. *International Journal of Digital Earth* 3 (4): 384–94. doi:10.1080/17538947.2010.486871.
- Wan, Zheng. 2009. 'Personalized Tourism Information System in Mobile Commerce'. In *Management of E-Commerce and E-Government, 2009. ICMECG '09. International Conference on*, 387–91.
- Weightless.org. 2017. 'Weightless - Setting the Standard for IoT'. Accessed April 1. <http://www.weightless.org>.

- Wilde, Erik, and A Vaha-Sipila. 2010. 'Uri Scheme for Global System for Mobile Communications (Gsm) Short Message Service (Sms)'.  
Yan, Zheng, Xueyun Li, and Raimo Kantola. 2015. 'Controlling Cloud Data Access Based on Reputation'. *Mobile Networks and Applications* 20 (6): 828–39. doi:10.1007/s11036-015-0591-6.  
Zarpou, Theodora, Charoula Drosopoulou, and Maro Vlachopoulou. 2013. 'Mapping the Tourism Mobile Applications: What, How and Where'. In *Proceedings of the 6th Balkan Conference in Informatics*, 206–12.