International Journal of Engineering and Geosciences (IJEG),
Vol; 5, Issue; 1, pp. 015-025, February, 2020, ISSN 2548-0960, Turkey,
DOI: 10.26833/ijeg.587023

AN IOT ARCHITECTURE FOR FACILITATING INTEGRATION OF GEOINFORMATION

Ümit Işıkdağ [1]*

[1] Mimar Sinan Fine Arts University, Department of Informatics, Istanbul, Turkey
(umit.isikdag@msgsu.edu.tr); **ORCID 0000-0002-2660-0106**

*Corresponding Author, Received: 04/07/2019, Accepted: 09/09/2019

ABSTRACT: Background: GeoInformation, is very valuable for a range of fields ranging from location based services and navigation to smart cities and homes. On the other hand today many fields benefit from Internet of Things (IoT) implementations, where the machine-to-machine and machine-to-human transmission of GeoInformation frequently occurs. This transmission usually occurs in multi-source/multi-target and multi-platform IoT environments. **Problem Statement:** In many cases real-time GeoInformation stays in its own island of automation, and thus its real value cannot be uncovered. This happens mainly due to inefficiencies and problems that occur in the storage, sharing and exchange of real-time GeoInformation as a result of multi-source/multi-target and multi-platform nature of the IoT architectures. **Research Approach:** Integration appears as a critical paradigm which should be focused in order to store, manage and transfer of GeoInformation efficiently in these complex environments. In this context, the focus of the study was to test the applicability of different technologies and integration methods for acquisition, transmission and visualisation of multi-source GeoInformation through implementing an IoT Integration Testbed Architecture which is utilizing low-cost hardware (to acquire information), graph databases(to store information) and standard IoT protocols (to exchange information). The implementation explained in this paper covers acquisition of real time GeoInformation from a set of real and virtual sensors, storage of this GeoInformation in Graph Databases, exchange of information through two different communication models (request/response and publish/subscribe) based on standard IoT protocols, and visualization of information by web pages, web mapping services and using a GIS software. **Results:** The implementation results demonstrated a proof-of-concept on how multi-source GeoInformation acquired from different type of IoT nodes can be integrated, stored and visualised on different platforms by utilising a standard IoT communication paradigms and multiple communication models.

Keywords: *IoT; GeoInformation; Integration; Graph Database; Arduino; Raspberry Pi; REST*

1. INTRODUCTION

Today we are living in a more connected world, where seamless communication between humans and machines (M2H) and between machines themselves (M2M) is becoming an inevitable need. On the other hand, more and more information is becoming geographically referenced, and GeoInformation appears a key element of the today's information infrastructures, along with the increasing need for location based services. The use of geoinformation for enabling and facilitating smart city environments covers a wide range of fields including 3D modelling of terrain, remote sensing for building boundary extraction, urbanization studies, energy efficiency for buildings, tourism, and 3D modelling of cultural heritage (Goksel and Dogru,2019; Mulazimoglu and Basaraner,2019; Karsli et al,2018; Dogan and Yakar,2018; Erdem and Ince,2018; Yemenicioglu et al,2016). In smart cities, devices operating in many different platforms, specifically in smart city environments are providing georeferenced information and a considerable amount of this information is in real time. In fact, although this real time information is very valuable for different domains, most of this information stays in its own island of automation, and thus the real value of information cannot be uncovered. Information integration would be a key facilitator in this case, to bridge the isolated islands of GeoInformation in order to reveal the value information, and a well-established integration would make GeoInformation available for big data analytics, machine/deep learning and other AI related tasks. The Internet of Things (IoT) paradigm can help very much in acquisition and presentation of real-time GeoInformation.

IoT is a newly emerging model of communication which focuses on facilitation of information exchange between all different types of Things. The Internet of Things envisions a connected world of billions of smart things, devices, smartphones, smart cars, smart homes, and smart cities (Black and White, 2017). In an IoT environment Things may be real or virtual, moving or steady but things are active participants in the whole system. Things communicate with each other (things-to-things communication) and also communicate and interact with humans (things-to-human communication.) (Gaikwad et al.,2015) Although in the beginning it was thought that connecting everyday objects to the Internet would have no real purpose and use, but today connecting Things to the Internet enables boundless possibilities for smart environments, workspaces and cities. According to CISCO, the number of things connected to the internet exceeded the number of people on earth during 2008; furthermore, the company foresees that the number of internet-connected things will reach 50 billion in 2020 (Spalazzi et al.,2014) Connected Things provide many interesting opportunities if the information travelling between them, and between them and humans, can be handled, stored, managed and presented efficiently. On the other hand once information is georeferenced (i.e. represented in the form of GeoInformation), its' value rises exponentially. As IoT architectures are multi-source, multi-platform and multi-domain, integration is the critical paradigm which would unlock the full potential of the GeoInformation transferred, exchanged and shared in the IoT architectures.

In order to contribute to the integration of GeoInformation in IoT environments, the aim of the study was to test the applicability of various integration methods for acquisition, storage, transmission and visualisation of multi-source GeoInformation. This is accomplished through implementation of an IoT Integration Testbed Architecture by utilizing low-cost IoT hardware, graph databases, standard and commonly used IoT protocols, and multiple communication models.

The methodology of this research was composed of two steps. The first one was the review of literature on IoT concepts and standards, and on IoT and GeoInformation, and the second step was the development of the proof-of-concept IoT integration architecture, which included development of use cases and software components.

Following the review of basics of, IoT and the role of GeoInformation in IoT environments, the paper provides details of the developed Testbed architecture, by providing in-depth information about each software component developed to test the GeoInformation integration in an IoT architecture.

2. IOT & IOT PROTOCOL STACK

The core concept of the IoT is the integration of physical objects into a global information network, where information about these objects are represented in the Internet. This makes a direct interaction with and between the physical objects possible (Jung et al.,2012). An IoT architecture connects anything to the Internet and in an IoT environment Things can exchange information according to the agreed protocols in order to achieve intelligent identification, positioning, tracking and monitoring (Xiaoying and Huanyana, 2011) IoT is a complex interconnected system of sensors, actuators, smart devices, software applications that communicate together to accomplish a task (Elkhodr et al.,2015). It is believed that IoT architectures will make Internet even more immersive and pervasive, where concepts like Persistent Computing, Omnipresent Computing and Ambient Intelligence will become more popular. Furthermore, by enabling easy access and interaction with a wide variety of devices such as, home appliances, surveillance cameras, monitoring sensors, actuators, displays, vehicles, the IoT will foster the development of a number of applications that make use of the potentially enormous amount and variety of data generated by such objects to provide new services to citizens, companies, and public administrations (Zanella et al., 2014). Different domains ranging from home/industrial automation, to mobile healthcare and assistance and intelligent energy management will benefit from IoT architectures. Libelium (2019) classified the domains where IoT Technologies can facilitate as Smart Cities, Smart Environment, Smart Water, Smart Metering, Security & Emergency, Retail Sector, Logistics, Industrial Control, Smart Agriculture, Smart Animal Farming, Domotic and Home Automation, and e-Health. Futuristic transport applications such as robot taxis will benefit much from IoT architectures. The autonomous UAV's are also in the same group. The possibilities offered by IoT architectures can be considered as limitless (Spalazzi et al.,2014).

IoT information integration can be defined as

combining data from different connected devices/sensors to produce more accurate, complete, and dependable information. This integration provides advantages as information acquired from multiple sensors helps, in understanding the surrounding environment more accurately and in seeing the big picture about certain events and environments, which is not possible using individual devices/sensors separately. In addition, the combination of information of multiple sensors produces new knowledge, which helps to build a context awareness model that helps to understand situational context.

IoT architectures makes use of both standard internet communication protocols and protocols developed for resource-constrained architectures. Protocols for communications in resource-constrained architectures are mainly developed by IETF. The literature presents many approaches for grouping and classifying the standards and protocols used in IoT architectures. IoT messaging standards and communication protocols (i.e. known as the IoT Protocol Stack) can be grouped into in three levels.

Level 1: The first level of the stack is composed of physical and datalink layers. The physical layer is the lowest layer in the communication protocols stack which deals with bit-level information transfer (Technopeida, 2019). The role of the datalink layer which resides above the physical layer is to encode/decode and organize the data bits prior to their transportation in form of frames in the network between the two nodes (Techtarget, 2019). The IoT paradigm benefits from several wireless technology standards at this level. These include; WiFi, low-energy networking standards such as LoRaWAN (Long Range Wide Area Network), IEEE 802.15.4, ZigBee, Bluetooth Low Energy (BLE) and finally high-energy LTE-A, (Long-term Evolution Advanced) which is (4G) cellular communication standard allowing data to be transferred in high-speed using a cellular network.

Level 2: The second level of the stack is composed of the network and the transport layers. The network layer provides paths for routing of data packets. The layer's role is finding the best path for the transfer of the data between the devices (Plixer,2019). The transport layer divides the data into segments where each data block has a port and sequence number, for ensuring the data reaches the target application in a correct order. This layer is also responsible for adjusting the speed of data transmission between the devices depending on the data sending and receiving capacities of the end-nodes. There are several standards and protocols used in IoT implementations at network layer including IP, 6LoWPAN, RPL and IPv6. At the transport layer the key standards are TCP and UDP. User Datagram Protocol (UDP) is the transport protocol used for achieving higher speeds for IoT applications.

Level 3: The third level of the IoT protocol stack is composed of session, presentation and application layers. The session layer opens, maintains and closes the sessions between multiple devices and determines which packets belongs to which data stream. Presentation is the layer where encryption and decryption (of critical) data occurs. The application layer is the layer where interaction with the user occurs. This layer aims to provide services to the end-user. Many well-known Internet protocols such as HTTP, FTP, SMTP, Telnet, DNS, and DHCP operate at this layer. There are several other protocols worth to mention in this level such as COAP and MQTT. Constrained Application Protocol (CoAP) is a protocol designed in IETF for resource-constrained devices based on RESTful architectural principles. Message Queue Telemetry Transport (MQTT) is another key protocol for IoT applications. MQTT is designed to operate on Low Power Networks. MQTT supports publish/subscribe (Pub/Sub) model of communication, while implementing a Message Broker. Many applications ranging from smart energy meters to monitoring applications makes use of MQTT today. The protocol fits well for M2M communication, when power consumption and bandwidth is a bottleneck. Also, MQTT is suitable for resource constrained devices that use unreliable or low bandwidth links. MQTT-SN was defined specifically for sensor networks and defines a UDP mapping of MQTT and adds broker support for indexing topic names (Al-Fuqaha et al.,2015; Salman and Jain,2017)

Another very commonly used architectural term in IoT architectures is the REST. REST stands for REpresentational State Transfer, and is neither a communication protocol nor a data encoding standard, but it is an architectural style for communication. REST has a key role and impact on IoT architectures. The REST architectural style utilises HTTP methods to enable and facilitate interaction in M2M and M2H communication. REST makes use of methods defined in RFC 2616 (HTTP) protocol, such as GET to request and acquire a representation of a web resource (i.e. in form of any type of data such as text, image), POST to create a web resource, PUT to change the representation of a web resource (i.e. update a resource), and DELETE to remove a web resource. The structure of a RESTful URI (Uniform Resource Identifier) should be straightforward, predictable, and easily understood (Rodriguez, 2018).

The IoT architecture explained in this paper implemented some of these messaging standards and communication protocols as:

- Wi-Fi for Level 1
- TCP/IP for Level2
- HTTP and MQTT for Level3
- RESTful architectural style is implemented for the integration of some components

Furthermore both (HTTP) publish/subscribe and (MQTT) request/response models of communication is implemented and tested within the architecture.

3. GEOINFORMATION AND IOT

Three main functions of IoT architectures are, acquiring information from the environment, conducting actions, enabling communication and interaction. Through the use of the sensors and actuators the real time information about a state and location of a Thing can be collected/stored, and also an action can be performed at a certain location. The georeferenced information is valuable for IoT architectures either when acquiring information about states of "Things" that is located at a certain region or when taking an action to change the state of "Thing" at a certain location. Furthermore, an action in an IoT environment may not only change a state of a "Thing" but also may change its location. In addition, an action can also change the form of a "Thing" or a "Space", and also the connectedness attributes (topologic relations) between the "Spaces". Thus, GeoInformation exchange is essential for IoT architectures to get informed about form, state, space and location, and topologic relation changes.

Internet of Things architectures can collect real time information regarding the state of many objects/utilities and environmental factors in the urban built environment, such as power grids, railways, bridges, tunnels, roads, wastewater systems, buildings, indoor facilities, street lights, parking spaces, trash containers, air quality, traffic, snow levels, water levels. Combining cloud computing, next-generation communication networks, and intelligent data mining technology, the Internet of Things helps in forming the smart city by making physical and informational resources integrate systematically (Zhou and Zhang, 2011) IoT technologies will change the geography research in 4 dimensions. Firstly, a new space will emerge, that extends beyond real geographical space through the real time information & interactions that are made possible with IoT. This new space will have impacts on Economic Geography, Social and Cultural Geography. The IoT will bring new human-land relationships through intelligent identification, positioning, tracking, monitoring of humans, furthermore real time information about land will be available seamlessly. The IoT will enable a more efficient logistics process lowering the distance/time barrier in movement of "Things" between spaces. Intelligent transportation refers to smarter use of transportation networks through utilisation of real time GeoInformation. Intelligent transportation would be very much facilitated by the IoT and will help humans to travel more quickly and safely. The IoT will help the formation of new behavioural geography where IoT will help in navigation and also travel decisions of humans, based on real time GeoInformation acquired about places. Furthermore, a person's health condition, eating and buying habits can also be tracked by indoors and outdoors by making use of GeoInformation. Personalised advice can be provided to humans in these areas based on the GeoInformation gathered by the sensors (Ming and Ling, 2012). In terms of urban geography, the IoT will contribute to the gathering and diffusion of, real time information about humans, spaces, land and movement of humans and Things. IoT architectures will help in localisation and positioning in urban spaces. Today, the movement of "Things" and humans generate mass amount of data via connected wireless devices such as smartphones (Keng and Koo,2014). This GeoInformation can later be organised to determine the semantics of locations. Real-time GeoInformation gathered by IoT architectures can be used for managing emergency response operations, for example information related to temperature, CO density, light density can aid the evacuation crew in fire situations (Liu and Zhu, 2014).In order to facilitate the processes that require the use of GeoInformation, IoT architectures, standards and protocols provide unique opportunities for information integration. Recent work in the field include Kamilaris and Ostermann (2018) which provides a survey that investigates the opportunities of location-aware IoT, and examines the potential of geospatial analysis in this research area. Rieke et al. (2018) presented a selection of approaches developed in different research projects to overcome the gaps that retain certain geospatial applications from using real-time information. Laska et al (2018) presented an architecture for real-time processing of spatiotemporal IoT stream data. Pozzebon et al. (2018) proposed a wireless sensor network framework for real-time monitoring of height and volume Variations on sandy beaches and dunes.

4. OVERVIEW OF THE INFORMATION INTEGRATION ARCHITECTURE

The aim of this study was to test the applicability of various technologies and integration methods for acquisition, transmission and visualisation of GeoInformation by implementing an IoT Testbed architecture that is utilizing low-cost hardware and graph databases. As hardware related costs form the key cost item in the IoT implementations, first objective of the study focused on testing the applicability of the low-cost hardware. As many IoT architectures would have a need to store many relationships between the nodes, the second objective of the study was focused on testing the applicability of the graph databases in an IoT Architecture. Previous studies indicating the potential and uses of Graph Databases for GeoInformation storage and retrieval include Agoub et al. (2016) and Nguyen et al.(2017).As the data in a graph database is more connected than standard SQL database, the query response times of graph databases are much faster, especially when there is a lot of relationships between objects. In parallel with these aim and objectives of the research, the user requirements are identified and unified in a generic high-level single use-case scenario. The scenario is developed with a focus group of 15 experts from academia and industry working in the field of IoT and GeoInformation Management. All use cases identified during the user requirements definition stage are illustrated in Figure 1. The use case scenario developed involves several use cases which can be grouped under "interaction for generation of virtual sensor values", "observation of sensor values through different interfaces" and "observation of sensor values through MQTT messages". The generation of sensor values involves the generation of transient and persistent values. The observation of sensor values can be done using web interfaces and also utilizing GIS software. This group of use cases also involves background processes such as update of a graph database and generation of views. Another group of processes involves the observation of sensor values through the MQTT messages. The architecture developed based on these use cases consist of several hardware and software components which are explained in this section. Figure 2 illustrates the deployment diagram of the IoT Integration Testbed Architecture implemented in this study.

Hardware Components: The hardware components of the system include a Data-Tier Web Server (Processor/Controller: x64 architecture / OS: Windows), an Arduino Yun (Processor/Controller: Atmel ATmega32U4 + Atheros AR9331 / OS: LininoOS), an Arduino Uno (Processor/Controller: ATmega328P), a Raspberry Pi 2 (Processor/Controller: ARMv7 / OS: Raspbian), and a Middle-Tier Web Server (Processor/Controller: x64 architecture / OS: Windows). The boundary boxes shown in the Figure 2 illustrates the software components hosted by each hardware component in the architecture.

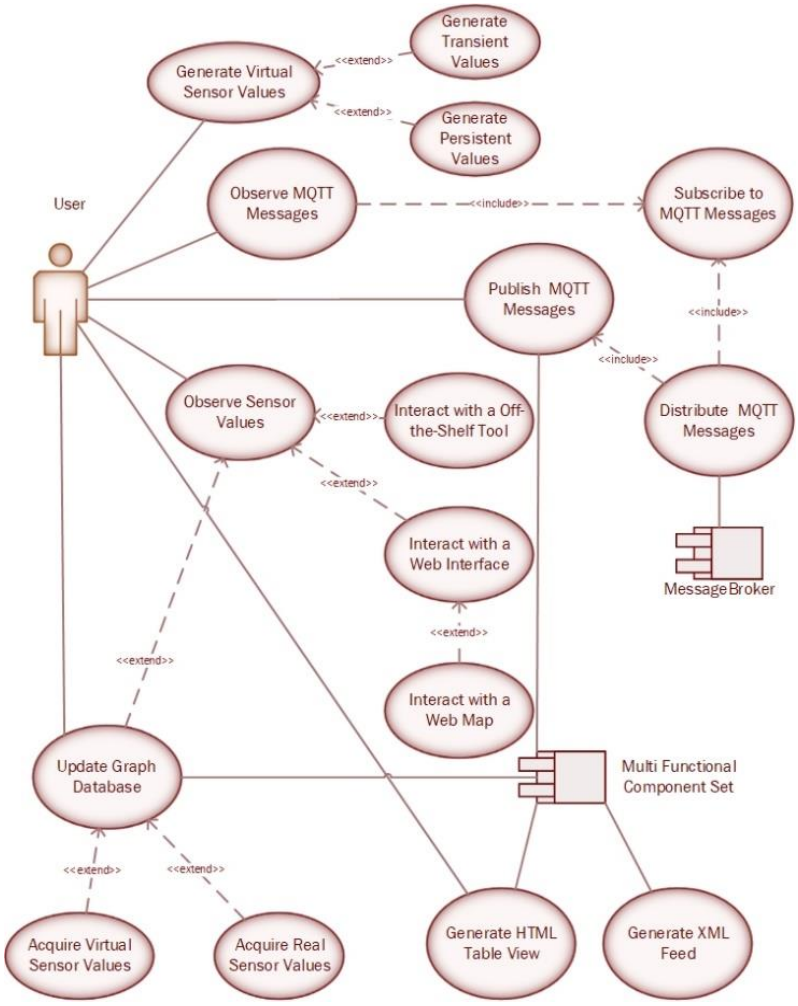

Figure.1 Use Case Diagram

Data-Tier Components: The data tier components of the system are, a Graph Database (Neo4J) server with Spatial Extension, XML and GeoRSS files residing in Data-Tier Web Server, IoT Node DB (which is an SQLite Database) and transient XML and HTTP files (generated on demand) residing in Arduino Yun , a transient XML file (generated on demand) residing in Arduino Uno.

Software Components (Developed): The software components of the system include Persistent Virtual Sensor (PVS) Generator, Graph Database Updater Component for Virtual Sensors, Cypher Query Translator (for Neo4J), Arduino Yun Local Database Updater & Data Viewer , Arduino Yun Graph Database Updater, Arduino Uno Simple XML Feed Generator, MQTT Message Publisher/Subscriber for Arduino Uno and Arduino Yun, MQTT Message Publisher/Consumer, Transient Virtual Sensor (TVS) Generator, GeoJSON Transformer, GeoRSS / GeoJSON Visualizers, Table View Generator, Statistical Data Generator.

Software Components (Utilized/Used): The software components that are utilised for the implemented architecture include an Apache Web Server on Data-Tier Web Server, NodeJS on Middle-Tier Web Server, uHTTPd Web Server on Arduino Yun, Mosquitto MQTT Server on Raspberry Pi 2, Arduino Client Library for MQTT (on Arduino Uno and Arduino Yun), QGIS Software on a client PC.

Output Files: The implemented architecture provides outputs in form of HTML and several other output file formats including XML, GeoRSS and GeoJSON. XML is a general purpose markup language, developed with the purpose of facilitating the exchange of data across different systems. As XML does not offer predefined tags, the structure (schema) and tags of XML files that are generated in this architecture were defined by the author. GeoRSS is a lightweight approach which extends RSS feeds, the standard provides an interoperable way for encoding geolocation to enable the geo-tagging of feeds. GeoJSON is a JSON (JavaScript Object Notation) compliant encoding of GeoInformation. RFC 7946 is the current standard that provides current GeoJSON specification. GeoJSON supports the following geometry types including Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon.

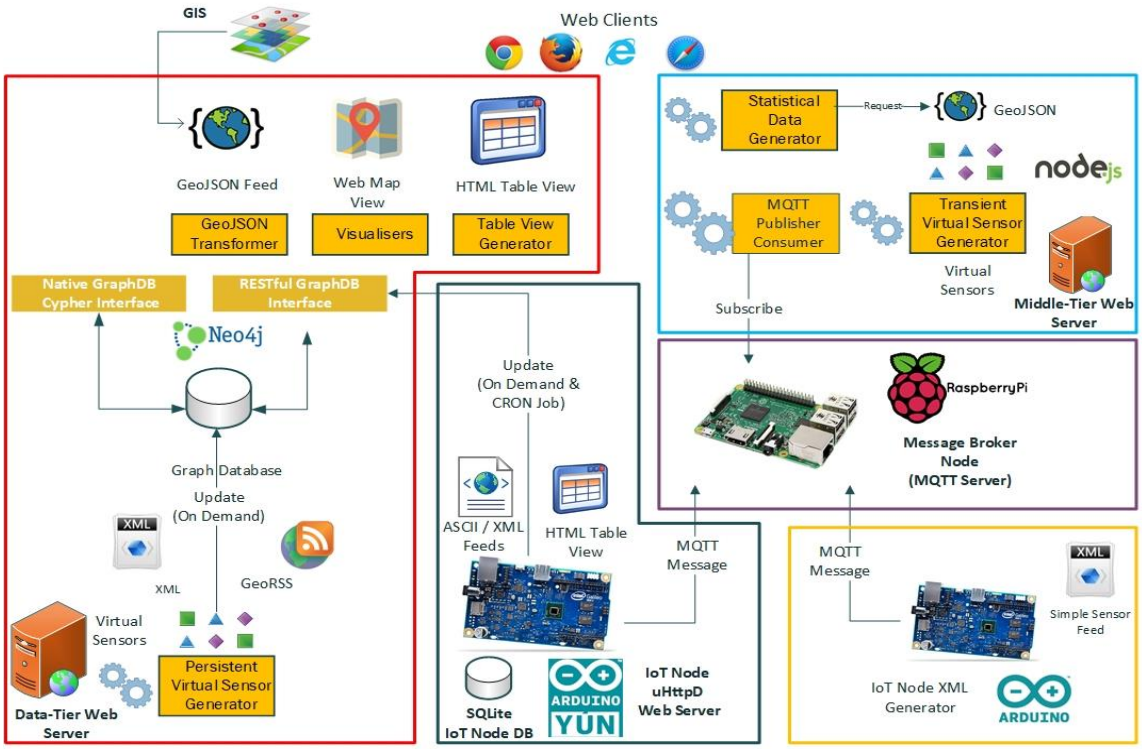

Figure.2 Deployment Diagram

5. COMPONENTS OF THE INFORMATION INTEGRATION ARCHITECTURE

This section will provide the details and the roles of the different software components that were developed and tested during the study. Figure 2 can be used as guide to observe the communication and interactions between these components.

Persistent Virtual Sensor (PVS) Generator: This component is developed to simulate the data coming from various sensors operating in multiple geographic locations, in order to test the data integration in the scenario of this architecture. The virtual sensor generator is composed of 2 software components that generate readings of 100 virtual single board computers (SBCs) where each virtual SBC contains readings from 500 virtual sensors. In total, 50.000 sensor readings can be generated as result of an HTTP Request. The types, names, values and coordinates of the sensors are generated randomly based on parameter value pool, i.) provided by a file authored by the end-user and ii.) also as the parameters of an HTTP Request. For instance, the end-user can specify i.) a set of geo-bounding-box coordinates where these virtual sensors would be located in as HTTP GET Request parameters, or ii.)a value pool of sensor types in a file. One of the PVS components generates a simple XML Feed, the other one generates a GeoRSS Feed, where the information about the location of the sensor is provided as a GeoRSS file. Both outputs are persisted in files.

Graph Database Updater for PVS Generator: The component is developed to test applicability and performance the RESTful calls for spatial object creation in Graph Databases by utilizing the mass GeoInformation in Sensor Feeds. This component processes/parses the all GeoInformation in XML and GeoRSS feeds generated by the Virtual Sensor Generator, then converts them into in-memory-objects and later makes use of HTTP POST requests to persist these objects as spatial objects in the Neo4J Graph Database. The storage of spatial objects in Neo4J is made possible by the utilisation of the Spatial Extension of the Neo4J database. The existence and structure of the persisted spatial objects can be observed by logging into the native user interface of the Neo4J. This component can be utilized on user demand or as a scheduled OS task.

Arduino Yun Local Database Updater and Data Viewer: This component is developed to test the capabilities of Single Board Computers (SBCs) in terms of real-time data acquisition, data storage and data presentation. The Arduino Yun SBC in this setup provides two interfaces, the first one is to the real-time a Sensor Feed generated by programming the Atmel ATmega32U4 microcontroller. This feed is served in form of an ASCII text or Simple XML on a specific port of the Arduino Yun LininoOS (port:80). The second interface is realized by uHTTPd web server running on another port (port: 82) of LininoOS. The Arduino Yun Data Updater and Viewer is set of components residing in this uHTTPd web server (port: 82). There is also an SQLite instance (i.e. IoT Node DB) running on the same operating system. The component operates on demand and once called, the Arduino Yun Data Updater and Viewer component acquires real time senor information (from port: 80) provided by the real time Sensor Feed (in ASCII text or Simple XML) on the same device, parses this information, and uses SQL to update the local SQLite Database, and following this, acquires records from the same database, and visualises it in form of a an HTML table at port:82.

Arduino Yun Graph Database Updater: The purpose of this component is to test the applicability and performance of the RESTful calls for spatial object creation in Graph Databases for storing real-time sensor information through the use of the hardware resources of SBCs such as Arduino Yun. The component parses real-time GeoInformation provided by a light level sensor connected to the Arduino Yun, and uses RESTful approach to persist this information in form of a spatial object in the Neo4J Graph Database. The spatial objects in the Graph Database can be observed by logging into the native user interface of the Neo4J. This component is invoked periodically through a CRON Job on the LininoOS, but also can be utilized by the user demand.

MQTT Publish/Subscribe Component for SBCs: This component is developed with the aim of testing the capability and performance of Single Board Computers (SBCs) for the IoT publish/subscribe model of communication by sending and receiving real-time GeoInformation provided by the sensors. The components are implemented in an Arduino Yun and in an Arduino Uno. The components are based on the Arduino Client for MQTT Library (2016) and Developed using Arduino IDE and runs on the Atmel ATmega32U4 / ATmega328P microcontrollers. The component is not illustrated in the deployment diagram (Figure 2) as it is running directly on the microcontroller level of the SBCs, instead the deployment diagram illustrates the messages sent by the component.

Once the component is compiled and run on Arduino Yun and Uno, these devices begin to broadcast MQTT messages in plain text (on port: 1883) to a MQTT Broker (implemented in the Raspberry Pi). The developed component also allows Arduino Yun and Arduino Uno act as subscribers of an MQTT Broker, so they also listen, receive and dump messages coming from the MQTT Broker. The operation of these components and the MQTT Broker in Raspberry Pi are observed and tested in a client PC, with the help of an MQTT client software, MQTT.fx.

Arduino Uno Simple XML Feed Generator: This component is developed to test the data acquisition and presentation capability of a very basic IoT node, i.e. an Arduino Uno equipped with an Ethernet Shield. The component is developed using Arduino IDE and runs on the Atmel ATmega328P microcontroller. The component is not illustrated in the deployment diagram as it is running directly on the microcontroller level of the Arduino, instead the deployment diagram illustrates the file generated by the component. The job of the component is very straightforward. The component can be called on user demand and once called, it acquires Light Level information from an LDR sensor using an analogue pin of the Arduino Uno, and generates a Simple XML Feed (a micro feed) and presents it to the user.

RESTful MQTT Publisher/Consumer for Async Frameworks: Asynchronous event driven server-side frameworks such as NodeJS, are popular for the IoT Server Side / IoT Cloud architectures. The aim of this component is to test the applicability of utilizing such a framework i.) for connecting to a MQTT Queue and also ii.) for publishing to /and consuming messages transmitted by the MQTT Queue. The developed component is a NodeJS Service, exposing a REST style URI. Using the service the end-user can subscribe to an MQTT Broker over HTTP, by providing the MQTT Broker's IP and the name of the MQTT channel. Once an URI such as [http://192.168.1.11:8081/mqtget /192.168.1.25/ch?channel=test/umit] is provided with an HTTP REQUEST, the component connects to the MQTT Broker for example on 192.168.1.25 (i.e. Raspberry Pi 2) and starts to listen a channel e.g. test/umit. Then, when the service provided by the component receives a message from the MQTT Broker, it parses this message and publishes it into a Web Page.

Depending on the user's preference, the contents of the message would also contain GeoInformation (i.e. location of sensors) as well as the semantic information (such as sensor type and value). The developed component also provides the capability of sending a custom MQTT message to an MQTT broker. The message is published through an HTTP GET method, by providing the MQTT Broker's IP and the name of the MQTT channel and Message Content. A URI such as [http://192.168.1.11:8081/mqtsend/192.168.1.25/payloa d?channel=test/umit&message=1050] would publish a message to the MQTT Broker, e.g. on 192.168.1.25, and to the targeted channel, e.g. test/umit, delivering a payload such as a sensor reading, e.g 1050.

Transient Virtual Sensor (TVS) Generator: The aim of this component is to provide additional set of virtual sensors by utilizing the asynchronous event driven server-side frameworks (such as NodeJS) in order to help the testing of multi-source information integration. The component is developed as a NodeJS service. Unlike with the Persistent Virtual Sensor Generator, the virtual sensors on this tier are in form of GeoJSON, and are transient, i.e. not persistently stored in files or a database. Coordinates of the sensors are generated randomly based on parameter value pool provided by the end-user. The number of sensors to be generated is also defined by a parameter. Once an HTTP GET request is sent through a RESTful URI, the component provides a (server side) response in form of GeoJSON. This response contains a "Feature Collection" of "Points" representing the location and properties (attributes) of the Virtual Sensors.

GeoJSON Transformer: This component is developed to test the data transformation capability in an IoT architecture through utilizing the spatial query capabilities of a Graph Database. The developed component generates a transient server-side response in form of a GeoJSON, as result of a web HTTP GET query sent through a URI. The query parameters include coordinates of a point and a radius value (provided by the user). Once the query is made, the component parses this query, utilizes the RESTful API of the Neo4J to transfer the query to the DB. The Neo4J DB, then makes use of its Spatial extension to respond to this query. As a response, Neo4J sends information related to sensors that can be found within the given radius of a given point. Once the query is responded by the Neo4J, the response is sent to the GeoJSON transformer in form of a JSON. The GeoJSON transformer then parses this JSON file, converts it into GeoJSON and responds back to the end-user. The component operates based on user demand/call.

GeoRSS and GeoJSON Visualizers: These two components are developed with the aim of testing how the integrated GeoInformation provided by multiple SBCs and sensors can be visualised over the web. The developed components are able to visualise both transient and persistent information. The visualizer components are developed in form of Web Maps, by making use of

well-known web mapping environment, Google Maps. The API of Google Maps is utilized to visualise the GeoRSS representation of Persistent Virtual Sensors (PVS) through getting sensor information from a set of files and transient real-time information generated by GeoJSON transformer. The second component utilizes GeoJSON Transformer to get the real-time information from sensors. As both visualizers use the same user interface, Figure 3 provides the sensor representation generated by GeoJSON visualizer as an example for sensor representation in these visualizers.

Table View Generator: The component is developed to test how non-graphical views can be generated to present integrated GeoInformation acquired and stored within this architecture. The component generates an HTML page with a table, showing the GeoInformation stored in the Neo4J Graph database. In the first stage an HTTP GET query sent through a URI. The query parameters include coordinates of a point and a radius value (provided by the user). As result of the same steps of interaction explained in GeoJSON transformer, Neo4J sends information related to sensors that are found within the given radius of a point as a response. The Neo4J response is then parsed and an HTML page containing a Table is generated by the component as the final output. The page containing the Table is sent to the end-user as the response.

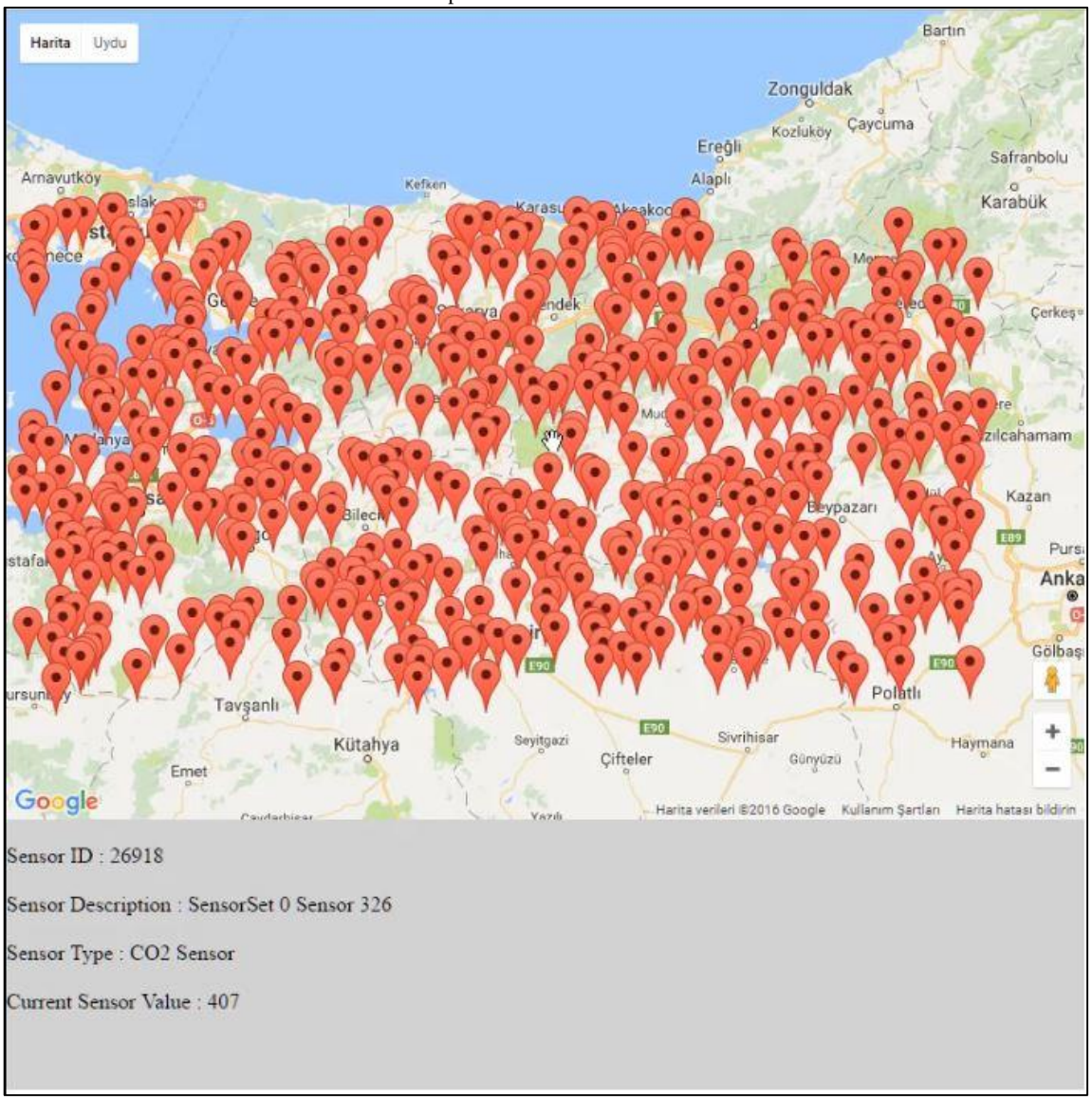

Figure 3. Visualisation of Sensor Information using Web Maps

Statistical Data Generator: The end users of IoT applications usually require summary statistics regarding the distribution of sensor types and sensor values. The component is developed to test how such information can be generated based on information presented by the components of this architecture. Statistical Data Generator makes use of information generated by the Transient Virtual Sensor (TVS) Generator. Given information about a number of sensors in form of GeoJSON, the component calculates summary statistics such as mean, standard deviation and variance for each sensor type. The calculated statistics are then presented in form of a JSON file. The component is developed in form of a NodeJS App. The component operates on demand.

The statistical data generator will be extended in the future research in order to cover some machine learning tasks including automatic classification of sensor values furthermore the future research will focus on implementation of LSTM networks which offer a great potential for prediction of the future trends of sensor values.

GIS Based Visualisation: It is also important to test the interaction options for the end-users. Thus, in the final phase of the implementation the information in GeoJSON and GeoRSS formats is acquired directly from i.)the GeoRSS representations and ii.)the transient real-time information in form of GeoJSON generated by the GeoJSON transformer.

Figure 4 provides the representation of this information inside QGIS software. The test revealed that values of sensors along with their type and other semantics attributes can be visualised and queried successfully using a GIS.

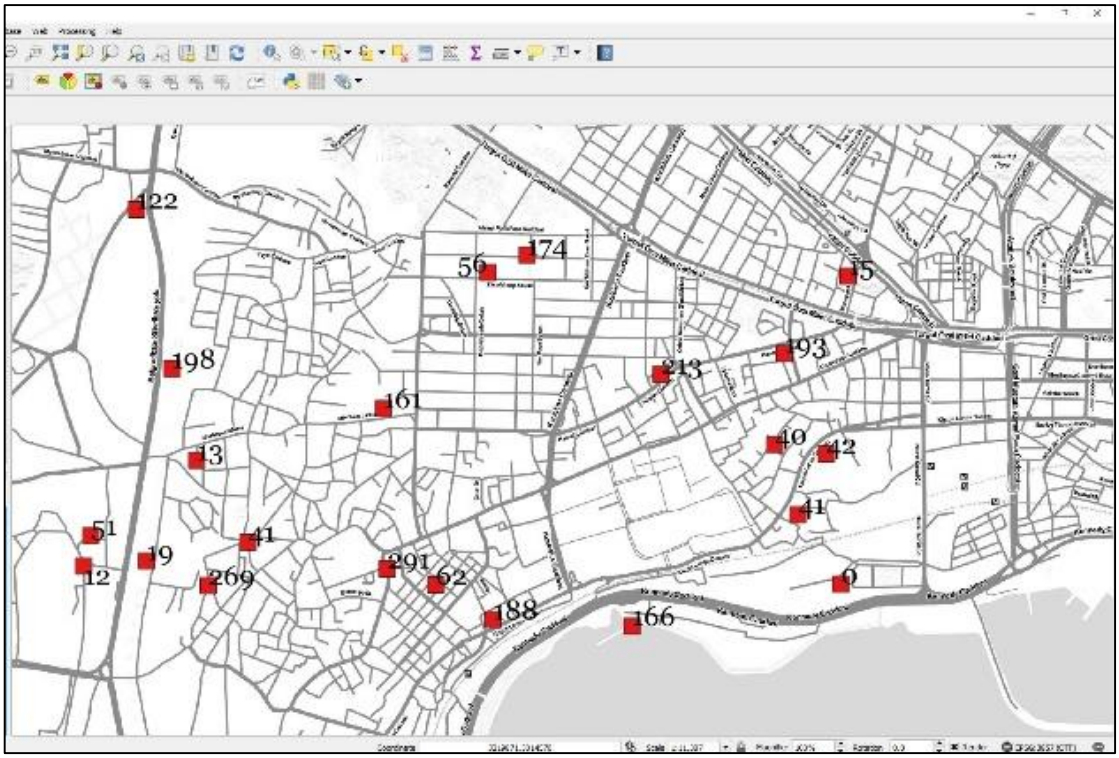

Figure 4. GIS Based Visualisation of Sensor Values

5. CONCLUSION

The research investigated the implementation of various technologies and integration approaches to test the applicability of a GeoInformation integration scenario. The scenario involved the acquisition of real time information from multiple sensors using low-cost hardware (including SBCs), transfer of this information from these devices to traditional data storage mediums such as files or and to modern data storage mediums such as graph databases with spatial information handling capability, development and implementation of RESTful interfaces to the data, and the development of the visualisation components.

The tests also included development of virtual sensor data generator components in order to help the testing of multi-source data integration capability, and mainly the support for non-real-time information fusion. In addition to that two communication models, publish/subscribe and request/response were tested for transferring information between the different IoT nodes.

The implementation results present a proof-of-concept on how multi-source GeoInformation acquired from different type of IoT nodes can be integrated, stored and visualised on different platforms.

Another finding of the research is that the combination of low-cost IoT hardware and graph databases can provide a unique opportunity for storage of large amount of GeoInformation and can respond to different kinds of spatial queries.

The key and unique novelty and scientific contribution of the study is providing a proof of concept architecture for establishing multi source-multi user interfaces, and multiple database/file integration by making use of multiple communication models(i.e. N to N to N. integration).

The research showed that Graph Databases have a unique potential to store and serve real-time GeoInformation. Although not used as commonly as the object-relational databases to store the spatial data, the graph databases have a great potential in dealing with vast amounts of geospatial data. In addition some operations that would be useful in IoT applications such as building up graph based topological data structures (such as space/state or space/event models) and analysis such as shortest path can be facilitated by the use of native graph databases.

Along with the data storage capabilities of the graph databases, this research also tested the automatic generation of interoperable data models such as

GeoJSON on demand, as response to an HTTP request, and the success achieved in this direction provides a proof on the applicability of Graph Databases and RESTful interfaces as a base layer for Geo-IoT web service architectures.

Another novel finding of the research was that publish/subscribe and request/response models of communication can be implemented alongside each other to transfer and interact with real-time GeoInformation.

The architecture can be extended with addition of various types of components and sensors. Other Single Board Computers (SBCs) that can be integrated with this architecture include Onion, Beagle Board, Orange Pi, Banana Pi, Asus Tinker Board, and Latte Panda. The sensors that can be integrated to this architecture is not limited with air quality and there are various types of sensors that can be integrated including traffic congestion detection sensors, water quality monitoring sensors, flood detection sensors, access control sensors, vehicle, human (elderly/child/patient) and product tracking sensors and devices, sensors for smart agriculture and farming, and fire and other hazard detection sensors.

The current visualisation options in the implementation is very limited and this can be expressed as the main weakness of the research. The future research will include new visualisation options using different data 2D data types such as polygons. Furthermore interpolation techniques such as Kriging can be used to estimate values between the sensors, and these estimated values can be visualised using raster visualisation techniques.

The future research will focus on integration of 3D GeoInformation with this architecture as 3D GeoInformation is mostly used 3D City and 3D Building representations, and integration of 3D GeoInformation would provide different query and visualisation opportunities. Visualisation tools such as Virtual Globes can be used for more efficient 3D Visualisations by making use of 3D GeoInformation models.

Due its current focus, this Testbed only tested integration options using standard and commonly used IoT protocols/architectures. On the other hand, in the recent future, implementation of GeoInformation oriented communication protocols and standards such as GeoMQTT, and OGC Sensor Things API, OGC Sensor Observation Service and OGC SensorML will be tested.

REFERENCES

Agoub,A.,Kunde,F.,Kada,M. (2016) Potential of Graph Databases in Representing and Enriching Standardized Geodata, DGPF, Available online at: https://www.dgpf.de/src/tagung/jt2016/proceedings/papers/20_DLT2016_Agoub_et_al.pdf

Akbulut, Z , Özdemir, S , Acar, H , Dihkan, M , Karslı, F . (2018). Automatic Extraction of Building Boundaries from High Resolution Images with Active Contour Segmentation. International Journal of Engineering and Geosciences, 3 (1), pp.37-42. DOI: 10.26833/ijeg.373152

Al-Fuqaha,A., Guizani,M., Mohammadi,M., Aledhari,M., Ayyash,M. (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications IEEE Communication Surveys & Tutorials, 2015 (17) pp. 2347-2376

Arduino Client for MQTT (2016) Available online: https://pubsubclient.knolleary.net/index.html (accessed on 10-June-2016)

Black, I. and White, G. (2017) Citizen Science, Air Quality, and the Internet of Things. In Internet of Things and Advanced Application in Healthcare, Reis,C.I. Maximiano,M.dS. Eds; IGI Global,USA,2017

Doğan, Y , and Yakar, M . (2018). GIS AND Three-Dimensional Modeling for Cultural Heritages International Journal of Engineering and Geosciences , 3 (2) , pp.50-55 . DOI: 10.26833/ijeg.378257

Elkhodr,M., Shahrestani,S., Cheung.H. (2015) A Smart Home Application based on the Internet of Things Management Platform. Proceedings of the 2015 IEEE International Conference on Data Science and Data Intensive Systems, Dec. 2015. Available online: http://ieeexplore.ieee.org/document/7396548/

Erdem, N and Ince, H . (2016). The Proposal of the Building Application for more Benefiting from Solar Light. International Journal of Engineering and Geosciences , 1 (1) , pp. 7-14 . DOI: 10.26833/ijeg.285215

Gaikwad,P.P., Gabhane,I.P., Golait,S.S. (2015) A Survey based on Smart Homes System Using Internet-of-Things. Proceedings of the International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Apr. 2015., Available online: http://ieeexplore.ieee.org/ document/7259486/

Göksel, C , Doğru, A . (2019). Analyzing the Urbanization in the Protection Area of the Bosphorus. International Journal of Engineering and Geosciences, 4 (2) , pp.52-57 . DOI: 10.26833/ijeg.446912

Jung,M., Reinisc,C., Kastner,W. (2012) Integrating Building Automation Systems and IPv6 in the Internet of Things. Proceedings of the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, July 2012, Available online: https://ieeexplore.ieee.org/document/6296937

Kamilaris, A., Ostermann, F.O. (2018) Geospatial Analysis and the Internet of Things, International Journal of GeoInformation 7(10), pp.269, Available online: https://doi.org/10.3390/ijgi7070269

Keng, D. and Koo,S.G.M. (2014) Spatial Standards for Internet of Things, Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings 2014), Green Computing and Communications (GreenCom 2014), and Cyber-Physical-Social Computing (CPSCom 2014), Sept. 2014, Available online at: https://ieeexplore.ieee.org/document/7059675

Laska, M., Herle, S., Klamma, R., Blankenbach, J. (2018) A Scalable Architecture for Real-Time Stream Processing of Spatiotemporal IoT Stream Data-

Performance Analysis on the Example of Map Matching, International Journal of GeoInformation 7(10),pp.238. Available online at: https://doi.org/10.3390/ijgi7070238

Libelium Web Site. (2019) Available online: http://www.libelium.com/resources /top_50_iot_sensor_ applications_ranking (accessed on 02-Jan-2019)

Liu,S. and Zhu,G. (2014) The Application of GIS and IoT Technology on Building Fire Evacuation, Procedia Engineering,2014 (71) pp. 577-582

Ming,L.A. and Ling yan,W. (2012) The Study on Geography under Internet of Things, IEEE International Conference on Computer Science and Automation Engineering, June 2012, Available online at: https://ieeexplore.ieee.org/abstract/document/6269452

Mülazimoğlu, E , and Başaraner, M . (2019). User-Centred Design and Evaluation of Multimodal Tourist Maps. International Journal of Engineering and Geosciences , 4 (3) , pp.115-128 . DOI: 10.26833/ijeg.535630

Nguyen, S. H., Yao, Z., Kolbe, T. H. (2017) Spatio-Semantic Comparison of Large 3D City Models in CityGML Using a Graph Database, ISPRS Annals Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W5,pp. 99-106

Plixer. (2019) Available online : https://www.plixer.com/blog/network-monitoring/network-layers-explained/ (accessed on 02-Feb-2019)

Pozzebon, A. Andreadis, A. Bertoni, D. Bove, C.(2018) A Wireless Sensor Network Framework for Real-Time Monitoring of Height and Volume Variations on Sandy Beaches and Dunes, International Journal of GeoInformation 7(10),pp.141.Available online: https://doi.org/10.3390/ijgi7040141

Rieke, M. Bigagli, L.,Herle, S., Jirka, S. Kotsev, A.,Liebig, T., Malewski, C., Paschke, T., Stasch, C. (2018) Geospatial IoT-The Need for Event-Driven Architectures in Contemporary Spatial Data Infrastructures, International Journal of GeoInformation 7(10) pp.385 Available online: https://doi.org/10.3390/ijgi7100385

Rodriguez,A. (2018) RESTful Web services, IBM Developerworks, Available online : https://developer.ibm.com /articles/ws-restful/ (accessed on 25-July-2018)

Salman,T and Jain, R. (2017) A Survey of Protocols and Standards for Internet of Things, Advanced Computing and Communications, 2017,1, Available online : https://arxiv.org/ftp/arxiv/papers/1903/1903.11549.pdf

Spalazzi,L., Taccari,G., Bernardini,A. (2014) An Internet of Things Ontology for Earthquake Emergency Evaluation and Response. Proceedings of 2014 International Conference on Collaboration Technologies and Systems (CTS), May 2014, Available online: http://ieeexplore.ieee.org/document/6867619/

Technopedia. (2019) Available online: https://www.techopedia.com/definition/8866/physical-layer (accessed on 10-Mar-2019)

Techtarget. (2019) Available online: https://searchnetworking.techtarget.com/definition/Data-Link-layer (accessed on 20-Mar-2019)

Xiaoying,S. and Huanyana,Q. (2011) Design of Wetland Monitoring System Based on the Internet of Things Procedia Environmental Sciences,2011(10),pp. 1046-1051.

Yemenicioglu, C , Kaya, S , Seker, D . (2016). Accuracy of 3D (Three-Dimensional) Terrain Models in Simulations. International Journal of Engineering and Geosciences , 1 (1) , pp.30-33 . DOI: 10.26833/ijeg.28522

Zanella,A, Bui,N., Castellani,A., Vangelista,L., Zorzi,M. (2014) Internet of Things for Smart Cities. IEEE Internet of Things Journal 2014 (1),pp. 22-32

Zhou,Q. and Zhang,J. (2011) Research Prospect of Internet of Things Geography Proceedings of the 19th International Conference on Geoinformatics, Jun. 2011, Available online at: https://ieeexplore.ieee.org/document/5981045