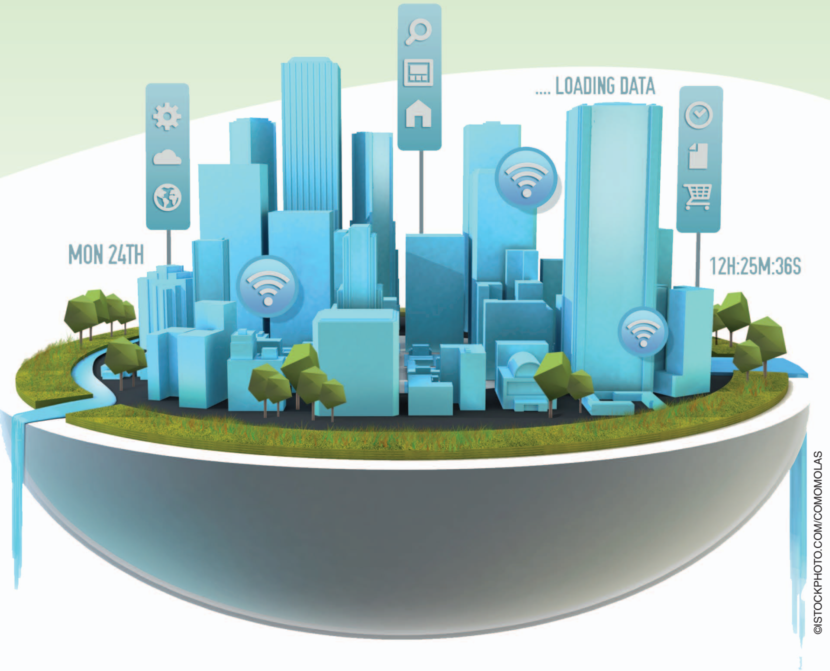
Smart ITS Sensor for the Transportation Planning Based on IoT Approaches Using Serverless and Microservices Architecture



©ISTOCKPHOTO.COM/COMOMOLAS

Luis Felipe Herrera-Quintero
Computer Science Program, Telecommunication Engineering Program,
Universidad Piloto de Colombia,
(e-mail: luis-herrera@unipiloto.edu.co)

Julian Camilo Vega-Alfonso
ITS Colombia, Colombia, (e-mail: klausbanse@gmx.net)

Klaus Bodo Albert Banse
Universidad de los Andes, (e-mail: jcvgalfonso@gmail.com)

Eduardo Carrillo Zambrano
Universidad Autonoma de Bucaramanga, (e-mail: ecarrill@unab.edu.co)

Digital Object Identifier 10.1109/MITS.2018.2806620
Date of publication: 20 April 2018

Abstract—Currently, there are many challenges in the transportation scope that researchers are attempting to resolve, and one of them is transportation planning. The main contribution of this paper is the design and implementation of an ITS (Intelligent Transportation Systems) smart sensor prototype that incorporates and combines the Internet of Things (IoT) approaches using the Serverless and Microservice Architecture, to help the transportation planning for Bus Rapid Transit (BRT) systems. The ITS smart sensor prototype can detect several Bluetooth signals of several devices (e.g., from mobile phones) that people use while travelling by the BRT system (e.g., in Bogota city). From that information, the ITS smart-sensor prototype can create an O/D (origin/destiny) matrix for several BRT routes, and this information can be used by the Administrator Authorities (AA) to produce a suitable transportation planning for the BRT systems. In addition, this information can be used by the center of traffic management and the AA from ITS cloud services using the Serverless and Microservice architecture.

I. Introduction

Intelligent transportation system (ITS)-based solutions strongly affect the traffic and transportation in cities or countries. Thus, governments and public authorities invest resources in several initiatives to promote and organize the ITS infrastructure [2]. As a result, public policy makers, transport planners, traffic engineers, research institutes and the private sector tend to develop or use new technology approaches that support several ITS services to lessen the energy consumption, congestion, and money required to build new transportation infrastructure [9].

A significant area in the transportation sector is mobility. For example, urban areas are facing important challenges in terms of diagnosis, planning and control. Thus, the availability of traffic information plays a key role in different processes such as preventing or handling traffic jams or calculating the average travel time, fuel consumption or air pollution. In addition, considering the miniaturization of electronics, a notably large quantity of hardware devices that can be used for traffic monitoring has been produced, such as inductive loops, video cameras, and ultrasonic sensors. However, most existing devices are expensive and sometimes larger in terms of hardware, so their installation and maintenance tend to be limited [3]. Nonetheless, the technology evolution from recent years has driven to introduce innovating solutions such as embedded and cheaper technologies, which revolutionize the telecommunications, computing systems and obviously transportation sector [2]. The embedded technologies provide benefits, such as reduction in cost, power consumption and size, flexibility and ease of installation. These benefits contribute to place devices and deploy systems on a large scale throughout the infrastructure of traffic systems to provide ubiquitous coverage.

These technologies are increasingly being combined into an Internet of Things (IoT). The IoT approach regards an infrastructure of interconnected objects, people, systems and information resources with intelligent services to enable them to process information of the physical and virtual worlds and react [22]. The IoT creates an intelligent and invisible network with several facilities such as monitoring several phenomena and controlling several devices and program tasks using IoT-enabled products based on embedded technology approaches. The IoT-enabled products generate a scenario where one can directly or indirectly communicate among them or with the Internet [23].

Above all, the ITS and technology are constantly evolving in recent times. Conventional technology-driven ITSs have been morphing into data-driven ITSs [12], which can offer new types of services using the vast amounts of available data [10], [13]–[15]. Furthermore, there is a global ITS working group of ISO TC 204 (International technical committee for ITS), which regulates and specifies the use of different applications and databases to exchange data among different systems [6], [11]. Simultaneously, a technical committee that supports the IoT development is JTC1, which focuses on information technology.

A previous study [1] designed and implemented a system that automatically construct the Origin/Destination matrix; however, to achieve that goal, we used several ITS sensors from several enterprises to gather the information. These ITS sensors were not based on IoT platforms and did not use cloud approaches to produce ITS services. After this study, we focused our efforts in improve our last approach [6], [36] by incorporating a new serverless and microservice architecture. It is worth noting that this paper continues to focus on the transportation planning for BRT (bus rapid transit) systems from the automatic construction of the O/D matrix using Bluetooth signals in the BRT stations. The novelty of this paper is based on the improvements of the serverless and microservice architecture to produce ITS cheap cloud services, where IoT approaches are used by the planning transportation authorities of the cities that deploy BRT systems.

The remainder of this paper is organized as follows. Section II illustrates the context of the proposed device. Section III describes the new proposal, where the IoT approach can be used in the serverless and microservice architecture; here, the paper describes each development level of the sensor and the platform architecture. Sections IV and V focus on the applications, deployments, related implementations and experimentation, and a case study with several tests based on the automatic generation of the O/D matrix is presented. Finally, Section VI describes the conclusions, and Section VII lists the acknowledgments.

II. Contextualization and Related Work

Currently, big cities such as Bogota in Colombia face major challenges in the field of transportation and mobility, such as traffic planning and management, which are notably important for the sustainable development of public transport in a country. In this area, the ITS is involved and offers several technological approaches.

There are various solutions for public transportation, but this study focuses on traffic planning for bus rapid transit systems (BRTs).

There are various solutions for public transportation, but this study focuses on traffic planning for bus rapid transit systems (BRTs). In fact, many cities are currently developing their BRT system, such as Santa Marta, Valledupar, and Sincelejo. Each BRT implementation should be developed as an integrated system with key attributes [20], which highlight the importance of good transportation planning.

In mass transportation systems, a highly important topic is the origin/destination (O/D) trip matrix, which represents the number of users with similar travel characteristics who move between zone pairs of the transportation network; the reference period is a fundamental input for most transportation planning and design problems [38]. Currently, in Bogota, these types of matrices are manually generated, i.e., people are deployed in the transportation network and must take data to build it. However, this fact causes many errors. Other countries (Spain, France, Italy, Chile, USA, Australia, etc.) have used several technologies such as Bluetooth [39], [40], [42]–[45] and a global standard protocol to wirelessly exchange information to support a matrix O/D construction. However, the Bluetooth technology in the O/D matrix construction context is used to simultaneously determine how a car moves through the transportation network and how people move through the public transportation network to build an O/D matrix. To do that, we focused on mobile phones that are equipped with Bluetooth technology. Because many people in Colombia have one of these phones, the O/D matrix can be built to improve the organization and planning for public transportation systems in Colombia.

Today, several developments focus on transportation planning; in fact, the Origin/Destination matrix tool is a widely used procedure for traffic planning. Thus, there is a solution for its automatic generation [20] based on the Bluetooth technology and the Service Oriented Architecture (SOA) paradigm, which implements a web service platform. Based on this solution, we incorporate the NoSQL approach to handle vast amounts of data produced by Bluetooth sensors [6].

In recent years, new approaches have emerged with a significant effect on ITSs. One of these approaches is the IoT. In accordance with ISO/JTC1, the IoT deployment and its capabilities focus on several sectors such as healthcare, information and communication technologies (ICT), manufacturing and heavy industry, finance and banking, food and farming, transportation, domestic, water, education, energy, entertainment and sports, public safety and military, retail and hospitality, and government [22]. In fact, several projects that aim to solve transportation affairs use the IoT approach, such as monitoring highways [23].

With the increase in complexity of any IoT system, new architectural styles have arisen, such as microservices, which improve the scalability, fault isolation, flexibility, etc. Each microservice can be independently deployed of one another, and the microservices are loosely coupled [30]. Serverless computing generates a new method to provide new computing resources in the frame of a higher layer of abstraction in cloud computing, where a piece of business function can be defined and executed on demand; the compute resources to execute it are properly allocated by the cloud provider [31].

Thus, there are many new technologies and advanced approaches that can be used to deliver ITS services, which can produce solutions for the planning transportation in BRT systems.

III. Proposal

A. Description

We propose approaches to generate the O/D matrix in previous studies and combine them for use based on a new Cloud and IoT architecture. Our system has versatility, low cost and fast deployment, which implies a single cheap device.

Because our system uses Bluetooth technology, it can fulfill four main tasks: gather Bluetooth signals, filter them, store data, and make the data accessible throughout standard interfaces to other devices and components. Our smart ITS sensor gathers data to generate the O/D matrix from the signals of Bluetooth-equipped devices that people have in the BRT systems. With our purpose, we can provide a reduced-cost, highly scalable, reliable and fast-deployment platform, which is a great solution for the traffic organization and planning in massive transportation systems.

In our proposal, we use IoT devices to design and implement a new ITS sensor, which focuses on gathering and processing data to generate the O/D matrix, using Bluetooth technology. As mentioned, this proposal builds on a

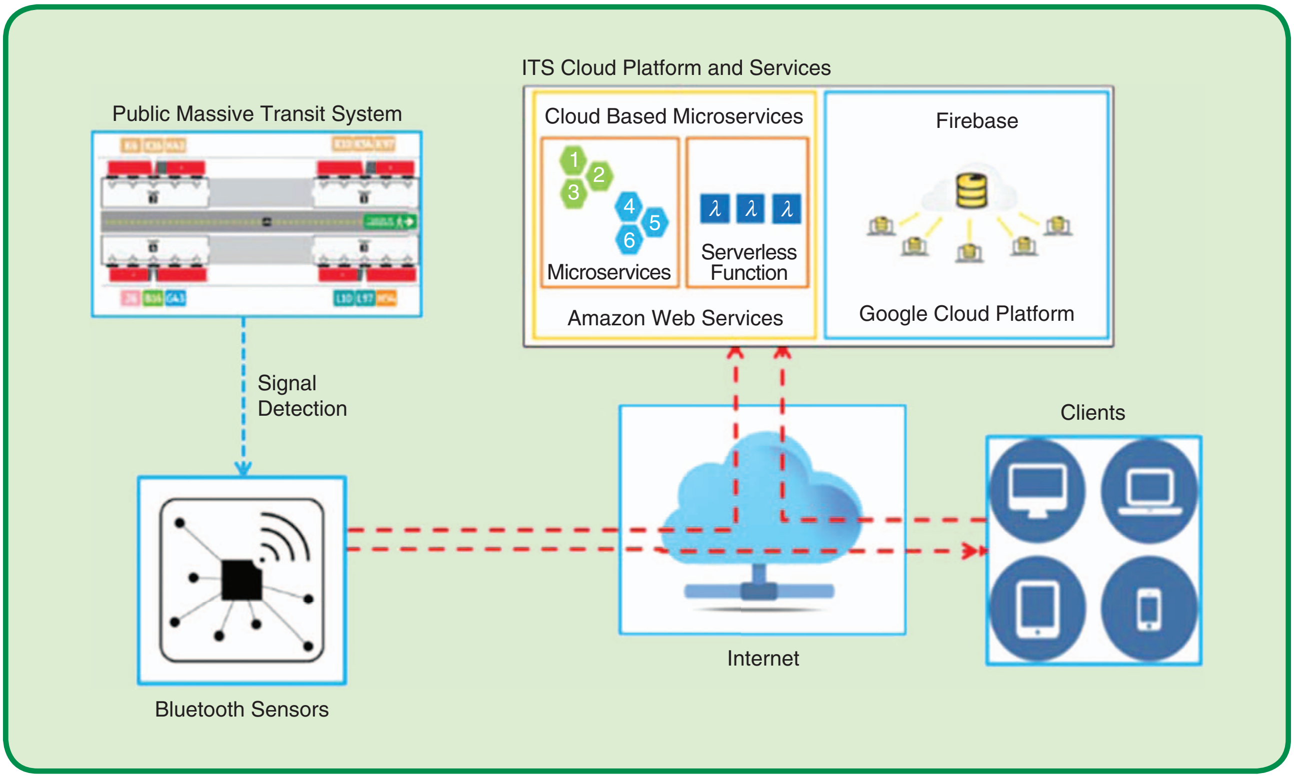


FIG 1 IoT system integration.

previous study [20] and uses its business logic (see figure 3) but replaces all components to use IoT-enabled devices. We create a cloud system that can generate the O/D matrix, and we propose a new architecture for the system based on micro-services and serverless functions. Then, the system operation is described.

The system has several devices that capture Bluetooth signals, use algorithms to filter these signals, and store the data in an embedded database. At a customizable interval, the data are sent to a cloud-based web service, which processes and stores the data into a NoSQL database also based on a cloud provider (see figure 1). Then, the data are accessible to the clients, application servers, and external components through standard interfaces such as web services. Finally, the serverless function uses the data provided by the sensors to automatically generate the O/D matrix.

B. System Architecture

The new design is component based, which provides a loose coupling among the components and high consistency. This improved system is made up of four parts as described in [20]: uptake signal, device classification and acceptance, device comparison and O/D matrix construction (see Figure 2).

As mentioned, the designed system has four subsystems. The first subsystem focuses on capturing Bluetooth signals. The second subsystem analyzes the captured filter data and decides which ones are accepted to be processed. The third eliminates duplicate records in a time range, and the last subsystem provides the results of an O/D matrix from a specific date and time. This system implements a web service platform and is component based.

Thus, this paper redesigns this architecture. It merges the capture, classification, acceptance, and subsystems into a unique subsystem that runs over an IoT-enabled device, and it even provides a local storage. Hence, the platform is composed of two subsystems and the system clients (Figure 3). The first subsystem, which is the sensor subsystem, is responsible of capturing, analyzing, filtering and storing Bluetooth signals and records; it also provides interfaces of configuration and administration. The second subsystem is composed of several components as follows:

- *Firebase:* Collects and stores the information prevenient from the first system.
- *Backend Apps:* Runs analysis and generation tasks for the O/D Matrix; provides standard application interfaces to enable the system clients to connect and retrieve data.
- *Device Management Services:* Set of micro services to administer and monitor the deployed devices.

In the process, two cloud providers are selected: Google Cloud Platform (GCP) [24] punctually Firebase and Amazon Web Services (AWS) [25], the latter to implement micro-services and server-less functions. Moreover, this architecture is easily replicable, at least in the major cloud providers such as Microsoft Azure [26] or even their IoT solution suites[27], [28].

C. Micro-Services and Serverless Function Implementation

Micro services are implemented using current technologies to simplify this job, i.e., Linux containers [32] provided by Docker [33]. For the Device Management Services (Figure 3), six micro services are developed and distributed in two containers (Figure 4). One container focuses on micro services for device management, and the other focuses on Microservices for Monitoring.

Then, the function to calculate the O/D matrix is placed on the AWS lambda service [34]. This service makes the calculations in response to events. In this case, the event that triggers the O/D matrix calculation is an HTTP request, i.e., when any client invokes the REST web service configured for this job. The AWS API gateway is put in front of the server-less function (Figure 5). After the calculations are performed, the results are stored in the database.

Both implementations tend to be notably similar (Figures 4 and 5), and the key difference lies in the computing resource provisioning. The micro-service model is a pre-provisioned and well-defined strategy for scaling when the workload peaks. The server-less function model is based on a per-execution basis, which implies that new resources are allocated for each execution. If the workload increases, the cloud provider clones the function and executes it in a notably new environment.

As mentioned, the proposal follows the SOA paradigm; therefore, the system platform uses Web services technology to integrate subsystems. Thus, the implementation follows the REST (Representational State Transfer) architectural principles to provide simple and intuitive interfaces to access the resources in the sensor and connect with the system clients. The implementation uses the IoT

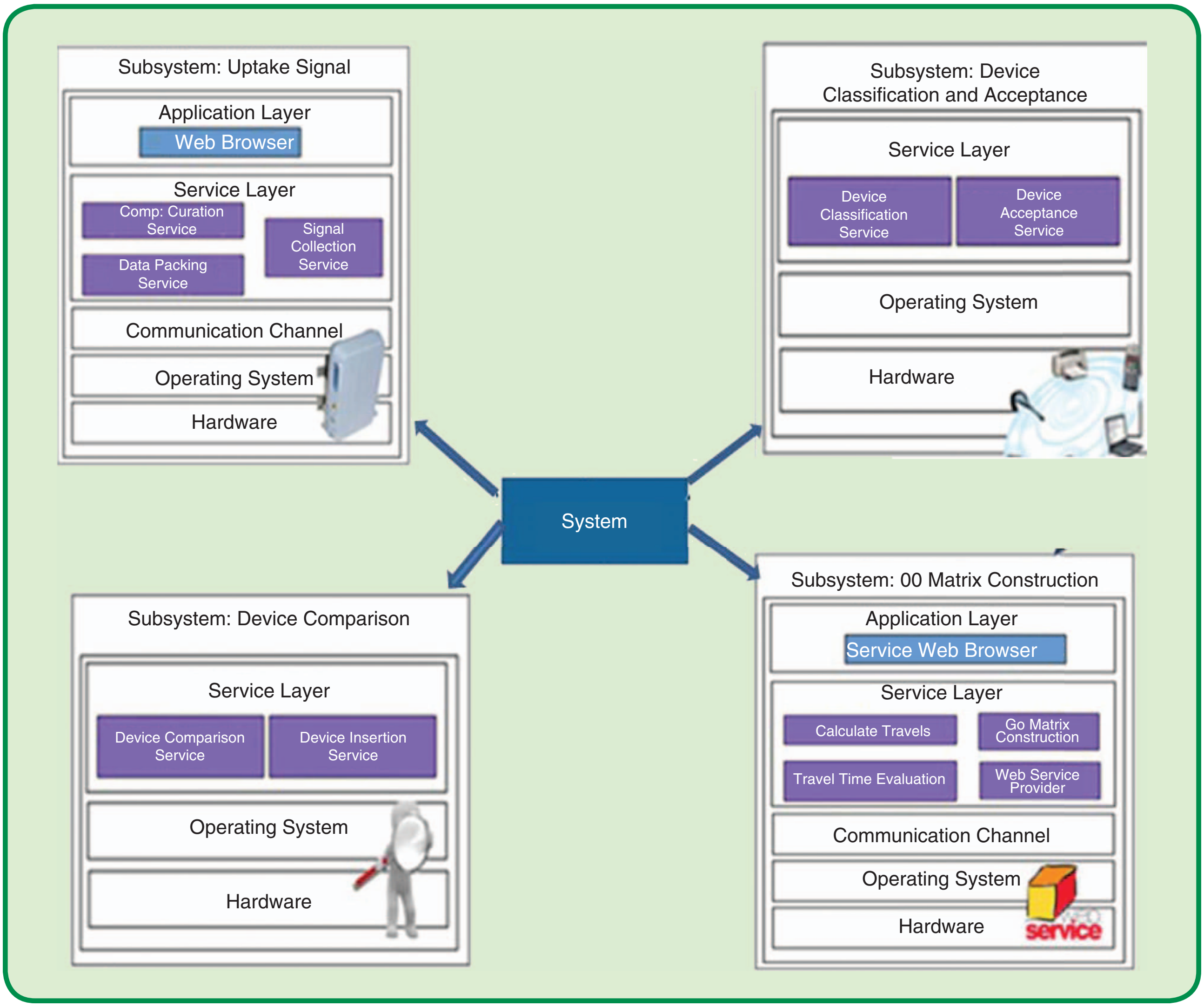


FIG 2 Previous architecture.

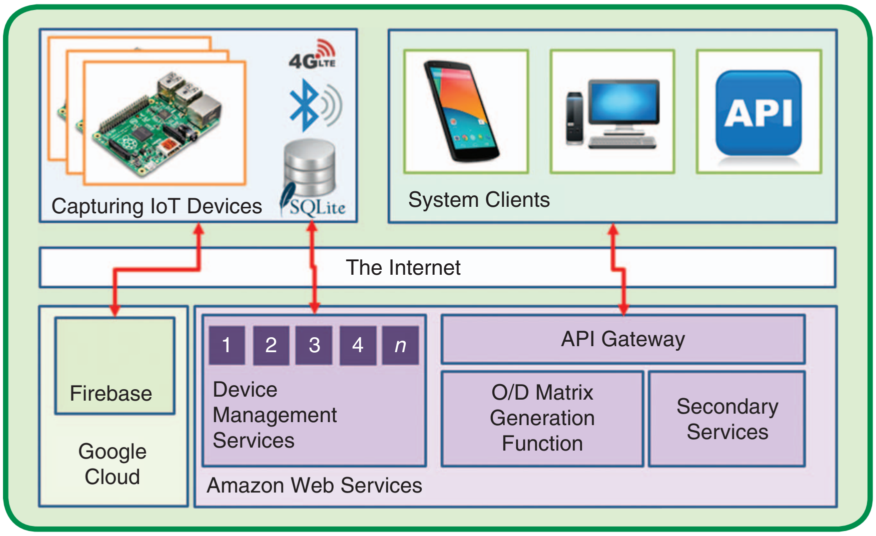


FIG 3 Proposed architecture.

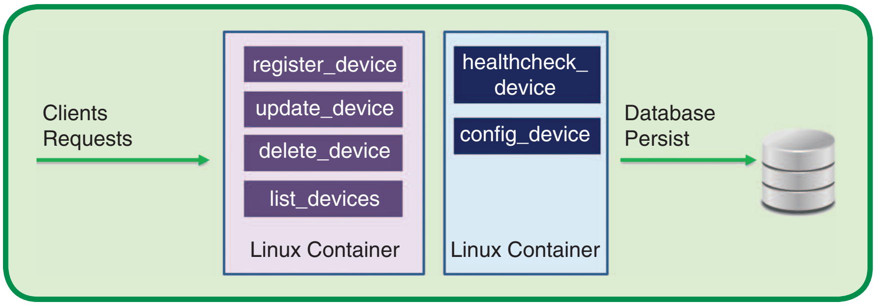


FIG 4 System micro-services.

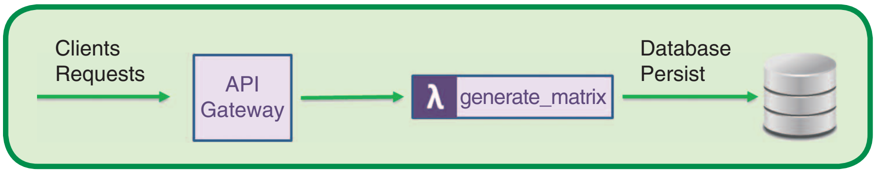


FIG 5 Server-less function structure.

device Raspberry Pi [21] to implement the sensor module. This product provides sufficient performance and characteristics for the implementation, and it is cheap and small (see figure 6).

D. Capture Device

The development of an IoT capture device is the major novelty in our proposal. It is composed of several technologic bricks, which supports the functionalities of the mentioned sensor subsystem. This device integrates an embedded web server, database and Bluetooth signal sniffer.

1. Hardware

The hardware mainly consists of two components: a single-board IoT device (Raspberry Pi) and a Bluetooth signal sniffer.

Table 1 describes the main specifications for the first generation of the raspberry pi, (currently on the 3rd generation), which was used in this project.

Several IoT-enabled devices are available in the market; thus, a comparison was performed before selecting the Raspberry Pi. In this case, several features were measured and compared: performance, amount of memory, community support, cost, etc. We synthesize those values in the following table.

2. Device Architecture

We designed and implemented an architecture to achieve the proposed functionalities for the capture IoT devices (see Figure 3). The IoT device architecture has different layers as follows (See Figure 7). In the top layer are web service interfaces to access the sensor functionalities; this interface also provides management options to set up the sensor. Lower, we develop different scripts, which interact with the interfaces provided by the operative system to access Bluetooth resources; those interfaces manage the Bluetooth adapter, perform Bluetooth inquiry and retrieve the captured signals, etc. (hcid). These scripts are event-driven; i.e., after the data are captured by the Bluetooth interface, the scripts perform the actions to automatically store that data in the local (Sqlite) database. Then, every 5 minutes, the same set of scripts collect the stored data and perform the routine to submit it to Firebase in the ITS Cloud Platform.

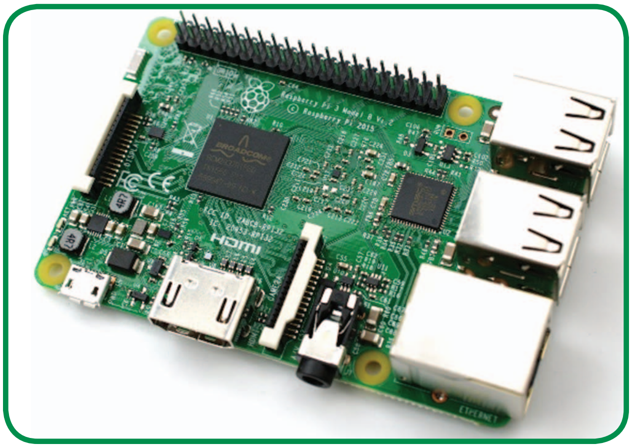


FIG 6 Raspberry Pi 3 aspect.

IV. Applications

To achieve our goal, two applications were developed. The first one is a web-based application (Figure 8) that offers a user interface for the traffic center or administrators and enables one to retrieve the current configuration of each sensor device and edit them. It also enables one to save its characteristics and location in the map.

Another functionality of this web application is the scan for Bluetooth devices in the range of the device, given a set of parameters such as the inquiry time or decision of whether to save on database (Figure 9).

The second one is a mobile application for the Android platform (Figure 10). This application enables the users from the transit authority to consult the O/D matrix for two points in a range of time. It also proves the correctness of the Backend Apps and its API end points (See Figure 3).

V. Case Study

The case study (Figure 11) contemplates three stages: the capture of Bluetooth signals by the IoT sensor; with the information processed and stored in said device, the intensive query of the data using the exposed web services; finally, the performance of load tests on the functionality that generates the O/D matrix when the data gathered from the devices are sent to the cloud platform. In the third stage, we test the use of serverless functions and the traditional web application approach.

1. Signal Gathering Test

In this stage, the proposed architecture for the IoT device is tested using a Raspberry Pi with a connected Bluetooth sensor. This test simulates that the IoT device is located in a station of a BRT system, where passengers carrying Bluetooth-equipped devices are circulating.

After a device enters the reading range of the Bluetooth sensor, the captured signal is processed and stored by the described applications. To achieve this test, the daily used devices of the people are used, as shown in the following table (Table 3).

Table 1. Raspberry Pi specs.

Characteristic	Value
CPU	Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
GPU	400 MHz VideoCore IV multimedia
Memory	1GB LPDDR2-900 SDRAM (i.e., 900 MHz)
USB ports	4
Video outputs	HDMI, composite video (PAL and NTSC) via 3.5 mm jack
Network	10/100 Mbps Ethernet and 802.11n Wireless LAN
Peripherals	17 GPIO plus specific functions, and HAT ID bus
Bluetooth	Version 4.1
Power source	5 V via MicroUSB or GPIO header
Size	85.60 mm × 56.5 mm
Weight	45 g (1.6 oz)
OS	Linux (raspbian, yoctolinux, ubuntu, CentOS, etc.), Windows 10 Microsoft, Android, Firefox OS
Price	36 US

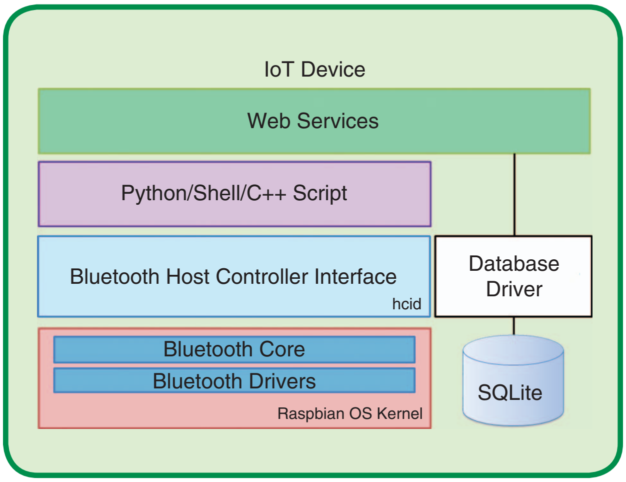
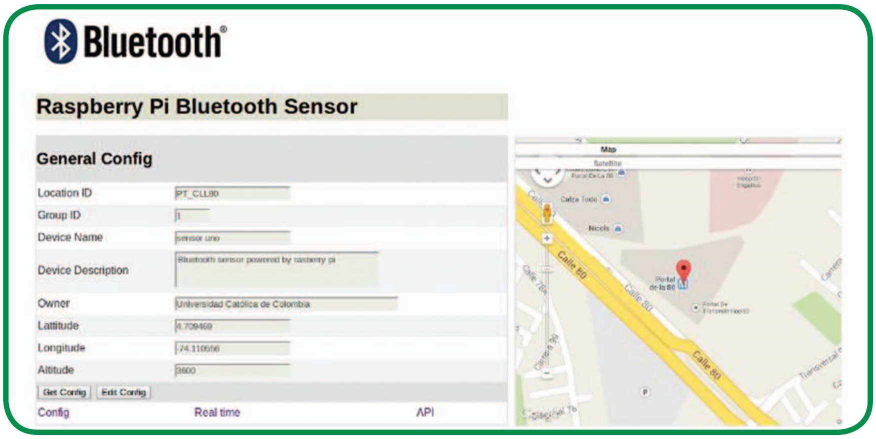


FIG 7 Capture device architecture.



Raspberry Pi Bluetooth Sensor

General Config

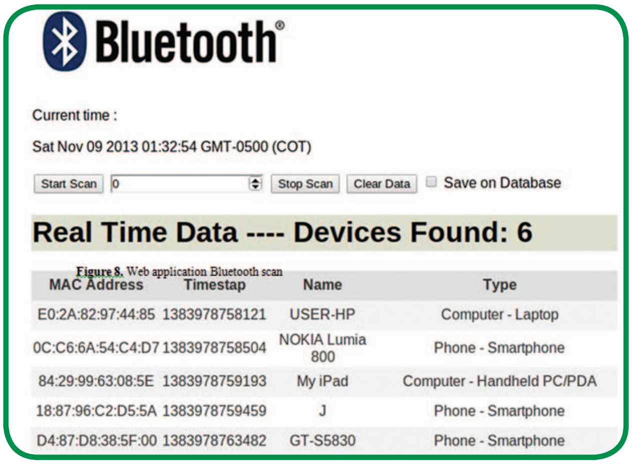
Location ID: PT_CLL80
Group ID: 1
Device Name: sensor uno
Device Description: Bluetooth sensor powered by rasberry pi
Owner: Universidad Católica de Colombia
Lattitude: 4.709469
Longitude: -74.110656
Altitude: 2600

FIG 8 Web application sensor set up.

2. Load Testing Iot Device

After all signals of the Bluetooth-equipped devices were captured and stored, we measured the response time of the web services in the Raspberry Pi (Figure 12). The web service was consulted many times using the Apache JMeter Tool [29], which simulates many clients; this service retrieves all stored records in the database in the Raspberry Pi.

With this test, we observe that the proposed capturing device can handle over a thousand requests with a



Current time :

Sat Nov 09 2013 01:32:54 GMT-0500 (COT)

Real Time Data ---- Devices Found: 6

Figure 8. Web application Bluetooth scan

MAC Address	Timestap	Name	Type
E0:2A:82:97:44:85	1383978758121	USER-HP	Computer - Laptop
0C:C6:6A:54:C4:D7	1383978758504	NOKIA Lumia 800	Phone - Smartphone
84:29:99:63:08:5E	1383978759193	My iPad	Computer - Handheld PC/PDA
18:87:96:C2:D5:5A	1383978759459	J	Phone - Smartphone
D4:87:D8:38:5F:00	1383978763482	GT-S5830	Phone - Smartphone

FIG 9 Web application Bluetooth scan.

12 - 5 - 2014

FIG 10 Mobile application.

Table 2. IoT device comparison.

Device	Price (USD)	CPU (MHz)	RAM (MB)	Max Storage (GB)
Raspberry Pi	35	1200	1024	64
Beagle Board	55	1024	512	4
PandaBoard	174	1024	2014	32
Intel Galileo	70	400	256	32

Table 3. Devices equipped with Bluetooth.

Device Class	Local Name	MAC Address
Laptop	User-hp	E0:2A:82:97:44:85
Smartphone	Nokia Lumia 800	0C:C6:6A:54:C4:D7
PC/PDA	My iPad	84:29:99:63:08:5E
Smartphone	HTC evo	18:87:96:C2:D5:5A
Smartphone	GT-S5830	D4:87:D8:38:5F:00
Wearable headset	222 Plantronics	00:19:7F:CB:AD:07
PC	Julian-PC	00:02:72:DA:2C:AD

considerable (50 and 100) level of concurrency without a decrease in performance. Furthermore, in this test, the proposed device handled 100% of the requests and the average time was 60 ms approximately.

3. Load Testing Proposed Architecture

In this case study, the proposed cloud schema and its response to thousands of IoT devices concurrently sending their data to the cloud were tested. The behavior of the serverless function was also compared to a traditional model (using a web server).

In the first test, the Apache JMeter Tool [29] was used. To simulate many devices that make requests to the system, 100 virtual server instances, each with 100 server threads, were launched, as shown in the next Figure (Figure 13).

The tested web service is in charge of saving the records in the NoSQL database. Figure 14 shows that the response time tends to increase to 700 ms and subsequently stabilizes, which is attributed to the cloud self-scaling in response to the high resource demand.

The second test was also performed using JMeter, but this time, we ran several requests against the service that calculates the O/D matrix. A server-less function must specify the memory to allocate per execution; 256 MB was set. In contrast, for the classic web service deployment using a web server, an AWS t2.micro [35] instance was used, whose specs

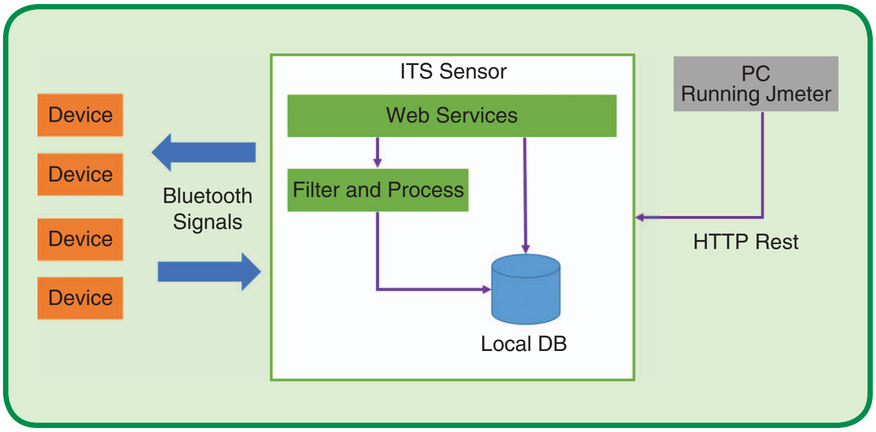


FIG 11 Case Study Test Overview.

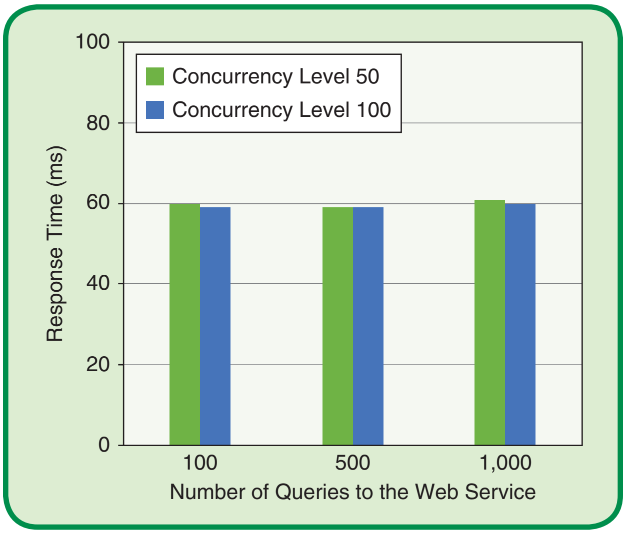


FIG 12 Query test.

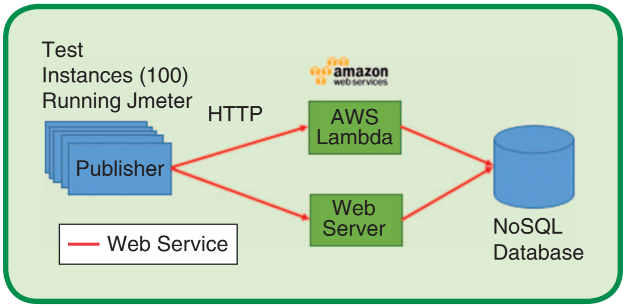


FIG 13 Serverless function test setup.

are 1 vCPU and 1 GB RAM. This instance is scaled in 1 unit if the response time exceeds 400 ms in a 5-min threshold (Figure 15).

There is a noticeable difference in average response time: the server-less function begins with a high response time, but it rapidly decreases with the increase in the number of requests per second. The classic web server deployment begins with a low response time and increases until ~600 ms of stabilization (Figure 16).

The first plot can be explained by the fact that each function invocation is run in a new environment; the first executions take more time; when the runtime is created, the actual function is loaded (the code), and computation resources are provisioned. In comparison, the server approach shows better response time in the first executions because the computational resources are previously provisioned and ready to be used. As a result of the process, the OD matrix is generated which can be observed in Table 4 and Figure 17. This matrix indicates the behavior of users of the BRT system in a measurement range, in a table the names of the sensors and the number of routes calculated between each pair. Additionally, a map of the city with the route of the users is generated.

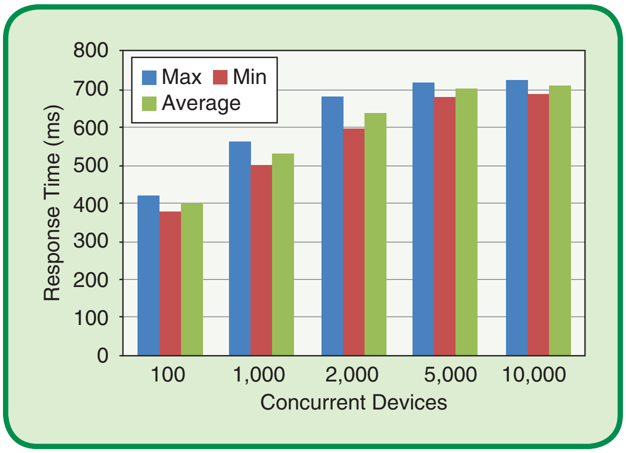


FIG 14 Cloud schema load test.

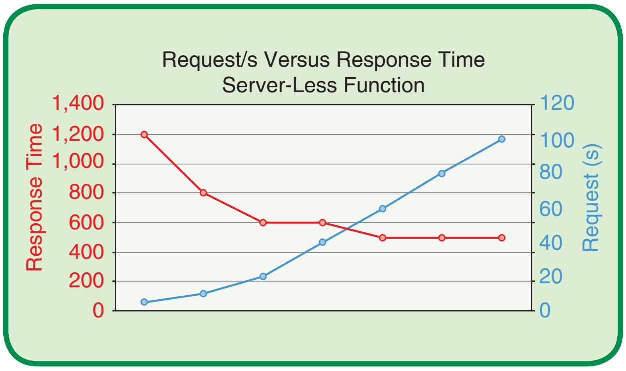


FIG 15 Serverless function load test.

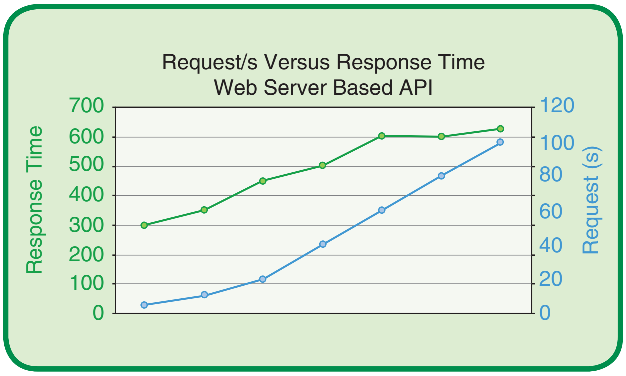


FIG 16 Web server load test.

Table 4. OD Matrix example.

Name of Sensors (O/D)	Alpha	Beta	Gamma	Delta
Alpha	–	300	100	50
Beta	0	–	0	150
Gamma	50	60	–	30
Delta	200	70	100	–

FIG 17 OD Matrix generated Map.

VI. Conclusions

This paper proposes a cheap and versatile serverless and microservice architecture approach that focuses on the Internet of Things for the transportation planning in BRT systems. The proposed model follows the SOA, which is a successful paradigm to integrate systems that implement the NoSQL approach. The NoSQL approach helps to handle vast amounts of data and integrate them into IoT-enabled devices, which are a revolutionary trend in the intelligent transportation field. The cloud providers Amazon web services and Google Cloud are used to produce the serverless and microservice architecture. Thus, this architecture reduces the operative charge in creating and deploying new functionalities and provides new provision for the resource for the same ones. In fact, the engineers in the surveillance and maintenance process can improve the daily IT processes for this system.

The proposed devices offer benefits compared to the previous innovating system because a single cheap device can accomplish the tasks that were meant to be performed by several devices such as computers, servers and sensors. Moreover, the proposed device assists the fast deploy of ITS solutions, reduces the cost, and increases the reliability because of the versatility, flexibility and easy access of the internet-of-things approach.

Future studies will design and implement message channels in the architecture or message brokers to communicate the microservices with several IT applications or legacy systems.

Acknowledgments

We acknowledge the Universidad Piloto de Colombia, ITS Colombia, Universidad de los Andes, Ministry of Transportation of Colombia, and Universidad Autonoma de Bucaramanga for their support to perform this research.

About the Authors

Luis Felipe Herrera-Quintero was born in Medellin, Colombia, in 1979. He received his PhD degree in ICT focused on Intelligent Transportation Systems (ITS). Currently, He is a senior lecturer on ITS and a research Professor at Universidad Piloto de Colombia. In addition, He is an ITS advisor at Ministerio de Transporte de Colombia. His research interests include ITS services delivery, Internet of Things, services-oriented computing, Smart cities. Mr. Herrera is member of the IEEE Intelligent Transportation Systems, Computer and Communication societies.

Julian Camilo Vega-Alfonso was born in Colombia in 1991. He received his engineering degree from the Universidad Católica de Colombia in 2013 and his Software engineering Master's degree in the ongoing year, 2017 from the Universidad de los Andes in Colombia. Currently he is software engineer at the industry. His research interests include ITS services architectures, cloud computing paradigm, big data solutions and IoT approaches.

Klaus Bodo Albert Banse was born in Germany in 1966, He is electrotechnical and telecommunications engineer and received the distinction of University Professor from Universidad de Cartagena, Colombia in 2003. He is guest lecturer on Intelligent Transportation System Design at Universidad de Medellin, Universidad Pedagogica and Technologica de Tunja in Colombia and several other universities in the region, as well as co-founder and currently president of the National and regional ITS associations ITS Colombia and ITS Latin America. He is also active as technical advisor on policy and project development in ITS and head of ITS design of the first intelligent highway deployments in the Andean Region.

Eduardo Carrillo Zambrano was born in Colombia in 1970. He received his Ph.D. degree in Information and Communications Technology from the Universidad de Valencia, Spain in 2004.

He is senior lecturer at Universidad Autónoma de Bucaramanga, Colombia. He is cofounder of Euro American Association on Telematics and Information Systems and Colombian Computer Society. His research interests include alternative energy sources, intelligent transportation systems and smart cities.

References

[1] Y. Cañon-Lozano, A. Melo-Castillo, K. Banse, and L. F Herrera-Quintero, "Automatic generation of O/D matrix for mass transportation systems using an ITS approach," presented at the IEEE Colombian Intelligent Transportation Systems Symp., Colombia, MO.

[2] L. F.Herrera-Quintero, F. Maciá-Perez, D. Marcos-Jorquera, and V. Gilart-Iglesias, "Wireless sensor networks and service-oriented architecture, as suitable approaches to be applied into ITS," in *Proc. 6th Euro American Conf. Telematics and Information Systems*, 2012.

[3] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, *Wireless Sensor Networks in Intelligent Transportation Systems*. Columbia, MO: Univ. of Missouri, 2009.

[4] X.Tao, C. Jiang, and Y. Han, "Applying SOA to intelligent transportation system," in *Proc. IEEE Int. Conf. Services Computing*, Orlando, FL, 2005.

[5] K. H. Chen, C. R. Dow, D. J. Lin, C. W. Yang, and W. C. Chiang, "An NTCIP-based semantic ITS middleware for emergency vehicle preemption," *Proc. IEEE Conf. Intelligent Transportation Systems*, 2008.

[6] A. M. Barragán-Charry, A. Forero-Zanabria, D. Valdez-Jimenez, K. Banse, and L. F. Herrera-Quintero, *Implementation of the New NoSQL Approach for ITS Applications*. Colombia, MO: Catholic Univ. of Colombia, 2013

[7] M. Papazoglou, P. Traverso, S. Dustdar, and F. Leyman, "Service-oriented computing: State of the art and research challenges," *IEEE Comput.*, vol. 40, Nov. 2007.

[8] "Final report of the service oriented architecture working group," ESafety Forum Working Group on SOA, Brussels, Germany, Tech. Rep. Feb. 2010.

[9] P. Kumar, S. Varum, and D. Reddy, "Advanced traveler information system for hyderabad city," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, Mar. 2006.

[10] A. Bahga and V. Madisetti, "Cloud-based information technology framework for data driven intelligent transportation systems," *J. Transp. Technol.*, 2013.

[11] J. Shibata, *ISO/TC204/WG3 (ITS Database Technology) Convener's Presentation to Joint CEN and ETSI Meeting (Cooperative Systems Standardization)*. Berlin, Germany, May 2010.

[12] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 2, 2011.

[13] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multiagent systems," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 2, 2011.

[14] D. Steil, J. Pate, A. Kraft, R. Smith, B. Dixon, L. Ding, and A. Parrish, "Patrol routing expression, execution, evaluation, and engagement," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 1, 2011.

[15] E. Schmmitt and E. Jula, "Vehicle route guidance systems: Classification and comparison," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Toronto, Canada, 2006.

[16] (2012). The promise of big data. Harvard School of Public Health. [Online]. Available: http://www.hsph.harvard.edu/news/files/2012/12/big_data.pdf

[17] L. Einav and J. Levin. (2013, Apr.). The data revolution and analysis. [Online]. Available: http://www.stanford.edu/~jdlevin/Papers/BigData.pdf

[18] (2013, Nov. 22). Duisburg Essen University embedded middleware in mobility applications. [Online]. Available: http://www.nes.uni-due.de/research/projects/emma

[19] (2013, Nov. 22). Embedded middleware in mobility applications. Univ. of Stuttgart. [Online]. Available: http://www.wissenschaftwirtschaft-bw.de/profile.232.htm

[20] Y. Cañon-Lozano, A. Melo-Castillo, K. Banse, and L. F Herrera-Quintero, "Web service platform for automatic generation of O/D matrix for mass transportation systems," presented at the 13th Int. Conf. ITS Telecommunications, Tampere, Finland.

[21] Raspberry Pi. [Online]. Available: http://www.raspberrypi.org/

[22] (2014, Aug. 24). ISO JTC-1, internet of things, preliminary report. [Online]. Available: http://www.iso.org/iso/internet_of_things_report-jtc1.pdf

[23] L. F, Herrera-Quintero, K. Banse, J. CVega-Alfonso, "IoT approach applied in the context of ITS: Monitoring highways through instant messaging," presented at the 15th Int. Conf. ITS Telecommunications, Copenhagen, Denmark.

[24] Firebase. [Online]. Available: https://www.firebase.com/

[25] Amazon Web Services (AWS): Cloud computing services. [Online]. Available: https://aws.amazon.com

[26] Microsoft Azure: Cloud computing platform & services. [Online]. Available: https://azure.microsoft.com/en-us/

[27] Azure IoT suite: An IoT cloud-based offering microsoft. [Online]. Available: http://www.microsoft.com/en-us/server-cloud/internet-of-things/azure-iot-suite.aspx

[28] How the AWS IoT platform works: Amazon web services. [Online]. Available: https://aws.amazon.com/iot/how-it-works/

[29] Apache JMeter: Application designed to test functional behavior and measure performance. [Online]. Available: http://jmeter.apache.org/

[30] S. Daya, N. Van-Duy, K. Eati, C. Ferreira, D. Glozic, V. Gucer, V. Gupta, S. Joshi, V. Lampkin, M. Martins, S. Narain, and R. Vennam, Microservices from Theory to Practice. Creating Applications in IBM Bluemix Using the Micro services Approach. IBM Redbooks.

[31] Server-less architectures: The evolution of cloud computing. MongoDB. [Online]. Avaliable: https://www.mongodb.com/blog/post/serverless-architectures-the-evolution-of-cloud-computing

[32] I. Melia, S. Puri, K. Owens, K. Tirumalai, S. Yellumahanti, L. Herrmann, M. Coggin, J. Fernandes, K. Craven, and D. Juengst. Cisco – Redhat. Linux containers: Why they're in your future and what has to happen first. [Online]. Available: https://www.redhat.com/en/files/resources/en-rhel-cisco-linux-containers-0091408.pdf

[33] What is docker. [Online]. Available: http://www.docker.com/what-docker

[34] Amazon web services: Lambda product details. [Online]. Available: https://aws.amazon.com/lambda/details/

[35] Amazon web services: EC2 instance types. [Online] Available: https://aws.amazon.com/ec2/instance-types/

[36] L. F. Herrera-Quintero, K. Banse, and J. C. Vega-Alfonso, "Smart ITS sensor for the transportation planning using the IoT and bigdata approaches to produce ITS cloud services," presented at the 8th Euro American Conf. Telematics and Information Systems, Cartagena, Colombia, 2016.

[37] J. Bogner and A. Zimmermann, "Towards integrating microservices with adaptable enterprise architecture," in *Proc. IEEE 20th Int. Enterprise Distributed Object Computing Workshop*, Vienna, Austria, 2016, pp.1–6.

[38] E. Cascetta and M. Postorino, "Fixed point approaches to the estimation of O/D matrices using traffic counts on congested networks," *Transp. Sci.*, vol. 35, no. 2, pp. 134–147, 2001.

[39] R. Albert. Proyecto piloto con sensores bluetooth en Vigo España y Chile información de tiempo de recorrido en tiempo real. [Online]. Available: http://www.trafficnow.eu/

[40] J. Zoto, R. J. La, M. Hamedi, and A. Haghani, "Estimation of average vehicle speeds traveling on heterogeneous lanes using bluetooth sensors," in *Proc. IEEE Vehicular Technology Conf.*, Quebec City, Canada, 2012, pp. 1–5.

[41] G. McGrath, J. Short, S. Ennis, B. Judson, and P. Brenner, "Cloud event programming paradigms: Applications and analysis," in *Proc. IEEE 9th Int. Conf. Cloud Computing*, San Francisco, CA, 2016, pp. 400–406.

[42] G. Michau, P. Borgnat, N. Pustelnik, P. Abry, A. Nantes, and E. Chung, "Estimating link-dependent origin-destination matrices from sample trajectories and traffic counts," in *Proc. IEEE Int.Conf. Acoustics, Speech and Signal Processing*, South Brisbane, QLD, 2015, pp. 5480–5484.

[43] J. Barceló, L. Montero, M. Bullejos, O. Serch, and C. Carmona. "Dynamic OD matrix estimation exploiting bluetooth data in Urban networks," in *Proc. 14th Int. Conf. Automatic Control Modelling & Simulation and Proc. 11th International Conf. Microelectronics Nanoelectronics Optoelectronics*, Stevens Point, WI, 2012, pp. 116–121.

[44] P. A. Laharotte, R. Billot, E. Come, L. Oukhellou, A. Nantes, and N. E. El Faouzi, "Spatiotemporal analysis of bluetooth data: Application to a large urban network," *IEEE Trans. Intell. Transport. Syst.*, vol. 16, no. 3, pp. 1439–1448, June 2015.

[45] B. Daniel, *Data Collection: Affordable Real-Time Traffic Adaptive Control President & CEO*. Denver, CO: Acyclica Inc.