Sensors for Web – Alternative for Traditional Mobile Applications

Alexandru Marius Obretin
Economic Informatics
Bucharest University of Economic Studies
Bucharest, Romania

***Abstract*— This paper is providing an in-detail analysis of existing mobile based software solutions for indoor localization and identifies several issues to be tackled. The objective this research initiative is assuming is to provide an alternative approach for traditional sensor-based applications, an alternative that behaves better in terms of resource consumption, maintenance costs and degree of generality. The principles hereby described have been integrated into a web application that has been tested in two different environments, providing indoor localization accuracy under a meter**.

Keywords:- *Signal Processing; Data; Information Systems.*

I. INTRODUCTION

The accelerated technological evolution from the past decades has encouraged the development of novel paradigms and fields where intelligent systems are able to address stringent problems. Continuous progress has elevated constructs like Artificial Intelligence, Internet of Things or Augmented Reality from buzzwords to actual information technology branches and along with the progress, the world has become more connected, the number of devices that communicate through the Internet has grown exponentially and data have become the most valuable asset on Earth according to The Economist [1].

The increased presence of technology in daily activities allowed companies with effective business strategies to better address client requirements, to improve existing infrastructures and accumulate wealth. An example where digitalization transformed an ordinary task is outdoor localization, where businesses like Google Maps, Apple Maps or Waze enhanced the existing global positioning system network of satellites to provide effective and reliable services that can accurately locate individuals within several meters precision.

While outdoor localization is no longer a challenge, the interests were shifted towards indoor localization. Academic research efforts and bold business ideas try to design systems capable of emulating the same facilities that are currently accessible outdoors in interiors. Market prospects are great, international events and conferences promote successful initiatives and promising results are expected from a field that would value 23.6 billion US dollars in 2023 according to some marketing analyzes of Future [2], Reports and Data [3] and IndustryARC [4].

II. CURRENT CONTEXT

According to a classification proposed by Lashkari et al. [5] current indoor positioning systems can be grouped in two categories:

- crowdsourcing – systems that imply an explicit interaction with the user, the information being harvested directly from the users
- crowdsensing – systems that require implicit user interaction, where the data is collected from sensors rather than the users

Besides the more extensive used techniques in indoor location that rely on external deployed equipment and wireless technologies to compose proximity based systems like WiFi, Zigbee, Bluetooth, acoustic signals, Radio Frequency Identification (RFID) or Ultra-wideband (UWB), this paper will focus more on those approaches that use software and common devices in detriment of dedicated hardware components. Comprehensive literature review of most important indoor location techniques that fall in the spectrum of the former category, like Received Signal Strength Indicator (RSSI), Angle of Arrival (AoA), Phase of Arrival (PoA), Time of Flight (ToF), Return Time of Flight (RToF), Line of Sight (LoS), might be found in Zafara et al. [6], Magdid et al. [7], Ferreira et al. [8] and Xiao et al. [9].

Exponents of the latter category are mostly represented by academic initiatives that try to address particular topologies specific to the university venue.

A map-independent indoor location mobile application designed for iOS that uses augmented reality for guidance and preset coordinates for the initial position and the desired destination has been proposed by Cankaya et al. [10].

A mobile application relying on built-in smartphone sensors that emulates a pedestrian dead reckoning system with adaptive tilt compensation to estimate accurate headings in various linear displacements has been detailed by Xu et al. [11].

Liu et al. [12] proposed a smartphone indoor positioning engine designed to be easily integrated with mobile location-based services, being a hybrid mixture of wireless signals and smartphone sensors.

SmartPDR, a smartphone application for pedestrian dead reckoning, capable of tracking pedestrians through typical dead reckoning approach using data from inertial sensors embedded in smartphones has been presented by W. Kang and Y. Han [13].

A vision-centered approach for a two-dimensional pedestrian indoor navigation system has been presented. The adopted strategy relies on a robust and fast computation of the camera orientation (3 dof) by tracking three orthogonal vanishing points.

A different application that uses the phone's camera in a vision-centered approach for a two-dimensional pedestrian indoor navigation system has been tested by Elloumi et al. [14]. This application relies on a strategy that is both robust and fast regarding camera orientation computations by tracking three orthogonal vanishing points.

An integrated three-dimensional mobile software solution has been proposed by Chen et al. [15] for indoor positioning as an alternative to traditional WiFi fingerprinting localization systems that are ineffective in terms of costs and performance when applied to complex indoor topologies.

A method for the real-time estimation of distance traveled based on double integral of acceleration has been tested by Troung et al. [16] inside a smartphone pedestrian dead reckoning system designed for commercial areas mileage estimates.

Statistical methods used for sensor signals noise filtering and error corrections have been discussed by Zhang et al. [17] in a research initiative intended to calculate displacement using the Micro Electromechanical System accelerometer integrated in a mobile phone.

A paper from Ayub et al. [18] described a Drift & Noise Removal Filter that is implemented by integrating the gyroscope, the magnetometer and the accelerometer available in a smartphone, approach which minimizes the drift and noise in output orientation. Their experiment details a numerical error correction phase intended to minimize the errors caused by gyroscope signal integration. Consequently, the orientation results obtained using the proposed architecture are smooth and less noisy compared to a digital compass. The technique described provides optimal resource consumption for computational efforts required in digital signal error correction and mileage estimation.

One common characteristic among all the aforementioned projects is the usage of mobile phone's built-in sensors. In order to achieve the goal of indoor localization and work as effective indoor positioning systems, these initiatives involve custom developed software applications that communicate with the sensors, read measurements, interpret the data and provide estimations. While most of them seem to provide accurate coordinates for targets indoors, they tend to be resource intensive. High frequency signal exchanges between the sensors and the application leads to a high throughput that needs to be filtered, processed, evaluated and persisted, all these operations being executed at application level, which actually drains the battery.

Moreover, when developing native mobile applications, usually two different programming languages should be considered for addressing the operating systems that dominate the market. Such approaches increase the development, testing and maintenance time, as often the sensors are differently optimized and calibration phases must be conducted accordingly.

III. ALTERNATIVE APPROACH

A different perspective lifts the overload of data filtration and processing from the actual device and provides a general-purpose application that works on both Android and iOS devices. To achieve these goals, instead of developing two distinct software solutions that address the same purpose, a web application would simplify the process.

An informational flow diagram of the application is detailed in Fig. 1.

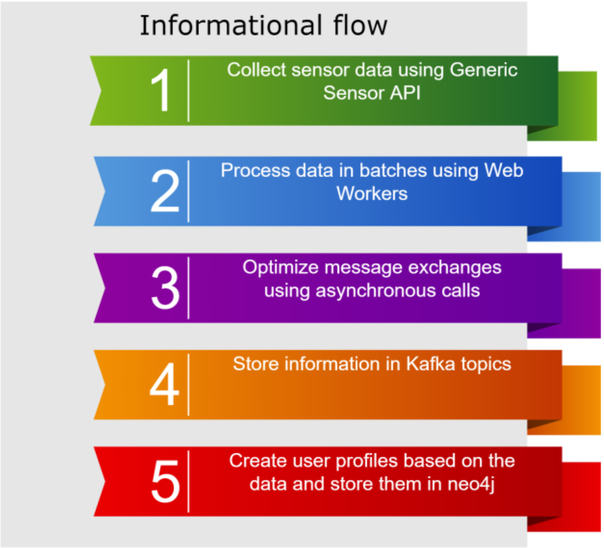
Informational flow
1
Collect sensor data using Generic Sensor API
2
Process data in batches using Web Workers
3
Optimize message exchanges using asynchronous calls
4
Store information in Kafka topics
5
Create user profiles based on the data and store them in neo4j

Fig 1:- Informational flow

But how can physical sensors on a mobile device can be accessed from a web application?

Starting from late 2017, a set of interfaces was proposed by Intel to expose the actual sensors to web applications. Concrete wrappers have been designed on top of the actual sensors, wrappers that inherit from a generic base class, which is designed to simply the process of accommodating new sensors in future and to grant a strict set of minimum requirements for any derived object. The base class is responsible for all common features a generic sensor could have, which reduces the implementation details in the wrapper classes, providing there only the logic that is relevant for a particular sensor. Starting from these architectural principles, several wrappers have been provided to interact with the accelerometer, the gyroscope, the magnetometer, the linear accelerometer sensor, the absolute orientation sensor, the relative orientation sensor and the ambient light senor. In terms of hardware components, there is no distinction between the absolute and relative orientation sensor. A smartphone usually has incorporated an accelerometer, a gyroscope and a magnetometer and these three are enough to emulate an attitude and heading reference system that could identify the angle between the frontal part of the phone and the magnetic North Pole. The implementation details in each wrapper class allow such a granular distinction and specialization.

Compared to DeviceMotion and DeviceOrientation events that are already accessible in web development and expose the accelerometer in the browser, the Generic Sensor API provides some additional enhancements:

- Open for extension design, providing the developer a set of interfaces that can easily be extended for increased code reusability.
- A mechanism for sensor availability detection.
- Configuration options for sensors, that allow different sampling rates depending on the requirements of the application.
- Synchronization between the sensor wrappers for accelerometer, gyroscope, linear acceleration sensor, absolute orientation sensor, relative orientation sensor, magnetometer and the screen coordinates.
- High precision sensor readings that enable synchronization between parallel operations.
- The Generic Sensor API is neither linked to the Window object, nor to the Navigator, providing the opportunity of integrating the interfaces with service workers for multithreaded web operations.
- Integration with Permissions API for better security and privacy.

Considering the enumerated bullets, this approach is suitable for developing generic, easy to maintain and extensible code that interacts with the sensors in a smartphone directly from the browser, using client side Javascript. So far, in the endeavor of providing similar functionalities with a native application, the Generic Sensor API is used for collecting sensor readings, this being one of the three operations that will happen at smartphone level.

One key aspect mentioned in the enhancements provided by the Generic Sensor API is the capability to run in conjunction with service workers. Traditionally, the user interface component of a web application is single-threaded, as a limitation that comes from Javascript. Does not matter how many CPU cores are available on the client machine that opens a website, all the operations run under a main thread in a sequential order.

This was a serious bottleneck for web development until web workers were introduced and with them the possibility for running in background, detached from the main thread, long-lasting scripts that would otherwise interfere with the responsiveness of the web page. The concept of web workers provided the ability of parallelization for intensive and expensive operations that would contrarily freeze the main thread, drastically impacting the user experience.

Although useful, the web workers come with a limitation. Running on a separate thread, they are unable to access the Document Object Model, hence they cannot update on their own user interface components. For example, assuming the application has to calculate the sum of the first thousand numbers in the Fibonacci sequence and then display the result in a card, a web worker would only be able to perform the computations, without being capable of changing the text on the card. Differently, if the main thread is used for the same purpose, the result might be displayed on the card when the operations are finished, but in the meantime the entire application will be blocked. The latter behavior is inconvenient, especially when a large number of operations are involved, as in mileage estimation, where the state is in a perpetual change. Nevertheless, in more complex scenarios where different tasks depend on each other, simply shifting time consuming operations on different threads might lead to errors if further actions rely on the result of the operation that is executed using workers.

An elegant solution for these impediments is to program the workers in an asynchronous manner, relying on premises that launch resource intensive operations on separate threads and wait for them to finish before executing a subsequent task. This way, neither the main thread gets blocked, nor the results remain unreachable. Asynchronous programming is a neat way of handling multiple actions in parallel, where proper usage of web workers allows message exchanges between the main thread and the workers providing solutions for transferring input data and results from one context to the other.

In this point, using the Generic Sensor API, the browser can communicate with the sensors in Javascript, relying on web workers that are spawned to listen to a particular component. Moreover, the web workers can collect sensor readings in batches and deliver results only when certain circumstances are met. Thus, neither the main thread that renders the user interface is responsible for data acquisition, letting it available for operations that really matter for the user, nor the web workers get overloaded. Modern hardware components that are integrated in smartphones are capable of high frequency measurements. According to specification details of most common accelerometers that are incorporated in communication devices they are able to provide readings ten times per second. If a web page would send to a server more than 600 requests per minute, as beside the measurements, other operations are also required, operations that would generate extra requests, sooner or later the server would close the connection with the client, being overwhelmed.

Communication in batches solves this issue and allows the developer to send more infrequent requests, as for mileage estimation there is no necessity for too often computations, considering some biological constraints that characterize human movements.

But in the virtue of creating a more efficient alternative to the native applications, the data filtration phase, required for normalizing noise in digital signals, and the data processing phase have to happen on a different device. Native applications usually do not deliver the raw data to an external component accountable for information analysis, but rather digest the data themselves.

In the current context, where the application is continuously generating events, another way of identifying sensor readings and an additional component that understands how to operate with them is mandatory. In a few words, event streaming refers to the process of gathering real-time data from various providers that include sensors, applications, services and others. The real-time and ongoing character of the process determines the naming streaming, as the data is presented as a stream of events: the value recorded by the accelerometer in the first second, then in the second, in the third and so on. Taken individually, these data are not meaningful and further manipulation, processing and interpretation is required, to acquire more context. Event streaming assures the information is accessible in a correct sequence and available retrospectively.

For this purpose, the proposed architecture is integrating Apache Kafka as an event streaming platform used for publishing and subscribing to sensor events and for storing and processing information at a later phase, as well. Being a secure, fault-tolerant, distributed and highly scalable solution, Kafka integrates well with the backend component of the application. The implementation details behind expose a producer-consumer architecture where communication is happening over TCP with server nodes, usually called brokers, responsible for managing a cluster of storage repositories, referred as topics, where information is categorized based on clear and predefined rules. The access is index based and multiple producers can write information in the same topic in a concurrent manner. The same topics can be simultaneously accessed for reading information, describing a convenient mechanism for input-output operations.

The indoor localization application is collecting information from the sensors using Generic Sensor API and web workers, the information is stored in batches to avoid unnecessary operations and when certain constraints are met, the web workers exchange asynchronous messages with a remote server to publish sensor recordings in dedicated topics.

In the meantime, the main thread is free for users to interact with the application and when requested, the mileage or the followed path could be displayed. Grace to the proposed architecture, all the data filtration, processing and interpretation steps are executed at server level, benefitting from better resources than a smartphone and when available, in a similar asynchronous approach, the path would be revealed on a floor plan with additional information presented to the user. This bidirectional communication can happen simultaneously, with a minimum impact on the phone's resources.

Up to this point all the three operations that are required at smartphone level have executed – data extraction, data delivery and result retrieval. Data filtration, data storage, data processing and data interpretation have been delegated to the server, drastically reducing the workload.

Last but not least, as Kafka is a temporary storage service, the data being erased after a particular time period, a database should be considered. In traditional native solutions, due to storage constraints that affect a mobile application, a light database is opted for. Often it is a relational database, that if not managed properly could occupy serious storage space from the available memory. Relational databases are common nowadays in most non-trivial applications, with different providers and options available, each one promoting better solutions for all the problems related to information storage.

What is usually misunderstood about the relational databases is related to the name itself. Inspired by Edgar Codd relational algebra, relational databases store information in a tabular way – a relation identifying in fact a table. Therefore, relational databases are appropriate for structured data where the most important characteristic is the data itself.

Discordantly, for an indoor positioning application that besides the localization service aims for more ambitious goals, like integrating a recommendation engine that would analyze user's behavior and try to determine patterns which will be further compared with some reference information in order to provide suggestions, the data collected from the sensors are not that important. Not even the information about the actual location. What really matters for such applications are the relationships between the user and different interest points inside a venue. Therefore, the information associated with such services is better represented as a graph structure, rather than tabular.

Consequently, the proposed architecture integrates at server level a neo4j graph database that is directly linked to the Kafka cluster and persists information about user's activity. This way, the information would be accessible at later time, long-term comparisons and analytics might be elaborated, backup procedures could be established, and additional social graph data processing techniques could be considered to extract more meaning from the harvested data.

The components diagram of the proposed application is detailed in Fig. 2.

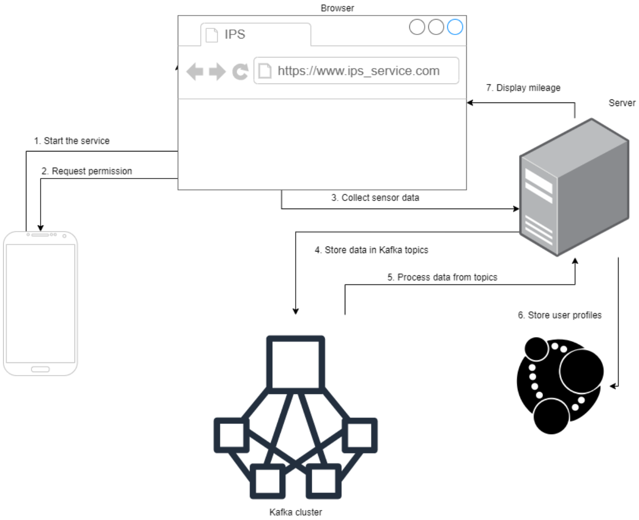

Fig 2:- Components diagram

IV. IMPLICATIONS

The architecture detailed in this paper has been implemented in a software solution that has been successfully tested in different contexts, providing indoor localization accuracy under a meter.

One the one hand, the hereby approach is infrastructure free, the initial costs required for deploying such service being far less than in traditional hardware backed indoor localization, where extensive grids of sensors are needed. Moreover, the maintenance of this architecture is more convenient, as the single point of failure is the software solution, that can be adjusted depending on the user's preferences. Additionally, the database integration allows easy debugging and effective issue solving capabilities, as the application might be requested to re-execute the data sample.

On the other hand, differently from traditional mobile applications designed for indoor localization, the proposed solution prioritizes the resource consumption, optimizes processes to prevent battery drain and provides generality, a web application being suitable for any smartphone, no matter the underlying operating system.

V. CONCLUSIONS

This paper introduced several software development choices that grant flexibility and help designing an effective indoor positioning system, as an alternative for the stakeholders interested in providing such capabilities in their venues without increased initial costs.

Although the results seem promising, there is still room for improvements, an integrated machine learning module used for data analysis or a recommendation engine backed by the patterns and behaviors identified from the long-term usage being two enhancements that would definitely add value to the final product and improve the user experience.

Last but not least, the analysis above emphasizes the benefits of interdisciplinary research in software development.

ACKNOWLEDGEMENT

This paper was co-financed from the Human Capital Operational Program 2014-2020, project number POCU / 380/6/13/125245 no. 36482 / 23.05.2019 "Excellence in interdisciplinary PhD and post-PhD research, career alternatives through entrepreneurial initiative (EXCIA)", coordinator The Bucharest University of Economic Studies.

REFERENCES

[1]. T. Economist, "The world's most valuable resource is no longer oil, but data," The Economist, 06 05 2017. [Online]. Available: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data. [Accessed 02 05 2020].

[2]. M. R. Future, "Indoor Positioning and Navigation System Market - Global Forecast 2023," Market Research Future, 11 2018. [Online]. Available: https://www.marketresearchfuture.com/reports/indoor-positioning-navigation-system-market-1775. [Accessed 02 05 2020].

[3]. IndustryARC, "Indoor Positioning and Navigation Market Research Report: Market size, Industry outlook, Market Forecast, Demand Analysis, Market Share, Market Report 2018-2023," 2018. [Online]. Available: https://www.industryarc.com/Report/43/global-indoor-positioning-navigation-market.html.

[4]. ReportsAndData, "Indoor Positioning System Market Size & Forecast, 2019-2027," 2019. [Online]. Available: https://www.reportsanddata.com/report-detail/indoor-positioning-system-market.

[5]. B. Lashkari, J. Rezazadeh, R. Farahbakhsh and K. Sandrasegaran, "Crowdsourcing and sensing for indoor localization in IoT: a review," IEEE Sensors Journal, pp. 2408-2434, 2019.

[6]. F. Zafara, A. Gkelias and K. K. Leun, "A Survey of Indoor Localization Systems and," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568 - 2599, 2019.

[7]. H. S. Maghdid, I. A. Lami, K. Z. Ghafoor and J. Lloret, "Seamless outdoors-indoors localization solutions on smartphones: implementation and challenges," ACM Computing Surveys (CSUR), vol. 48, no. 4, pp. 1-34, 2016.

[8]. A. F. G. G. Ferreira, D. M. A. Fernandes, A. P. Catarino and J. L. Monteiro, "Localization and Positioning Systems for Emergency Responders: a Survey," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2836-2870, 2017.

[9]. J. Xiao, Z. Zhou, Y. Yi and L. M. Ni, "A Survey on Wireless Indoor Localization from the Device Perspective," ACM Computing Surveys (CSUR), vol. 49, no. 2, pp. 1-25, 2016.

[10]. I. A. Cankaya, A. Koyun, T. Yigit and A. S. Yuksel, "Mobile indoor navigation system in iOS platform using augmented reality," in 9th International Conference on Application of Information and Communication Technologies (AICT), Rostov on Don, 2015.

[11]. S. Xu, R. Chen, Y. Yu, G. Guo and L. Huang, "Locating Smartphones Indoors Using Built-In Sensors and Wi-Fi Ranging With an Enhanced Particle Filter," IEEE Access, vol. 7, pp. 95140-95153, 2019.

[12]. L. Jingbin, C. Ruizhi, P. Ling, R. Guinness and H. Kuusniemi, "A Hybrid Smartphone Indoor Positioning Solution for Mobile LBS," Sensors, vol. 12, pp. 17208-33, 2012.

[13]. W. Kang and Y. Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization," IEEE Sensors, vol. 15, no. 5, pp. 2906-2916, 2015.

[14]. W. Elloumi, K. Guissous, A. Chetouani, R. Canals, R. Leconge, B. Emile and S. Treuillet, "Indoor navigation assistance with a Smartphone camera based on vanishing points," in International Conference on Indoor Positioning and Indoor Navigation, Montbéliard-Belfort, 2013.

[15]. G. Chen, X. Meng, Y. Wang, Y. Zhang, P. Tian and H. Yang, "Integrated WiFi/PDR/Smartphone Using an Unscented Kalman Filter Algorithm for 3D Indoor Localization," MDPI Sensors, vol. 15, no. 9, p. 24595–24614, 2015.

[16]. P. H. Truong, S. Kim and G. Jeong, "Real-Time Estimation of Distance Traveled by Cart Using Smartphones," IEEE Sensors Journal, vol. 16, no. 11, pp. 4149-4150, 2016.

[17]. R. Zhang, X. Liu and Z. Qu, "The Research of Distance Measurement Method Based on Mobile Phone MEMS Accelerometer," in 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2016.

[18]. S. Ayub, A. Bahraminasab and B. Honary, "A sensor fusion method for smart phone orientation estimation," in Proc. 13th Annu. Post Graduate Symp. Converg. Telecommun. Netw. Broadcast., 2012.