Adapting distributed stream processing technologies for the automation of modern health care systems

Rhaed Khiati, Muhammed Hanif, Choonhwa Lee

Division of Computer Science and Engineering, Hanyang University, Seoul, South Korea

{seahawk32, honeykhan, lee}@hanyang.ac.kr

***Abstract*— With the increase in population, there is an increasing number of patients. Subsequently, we also see an increase in the amount of patient data that needs to be processed, further emphasizing the need for new systems and developments that can handle such large quantities of big data. To this end, this paper proposed a potential solution to this problem in the form of a system that can analyze a patient's data in real-time, providing doctors and other intended healthcare personnel with an immediate report of a patient's situation, allowing for a quicker response time, better treatment, and the first step towards a grand realized smart hospital system in the long-term. This system, aided by the rapid analysis of Apache Flink, produces the requested data to the doctor as intended, enabling for a swift response time to patient issues, thus highlighting a unique approach to the field, a contrast to other previous research in this field where there is a lack of said provisions.**

***Keywords*— Big Data, Smart Healthcare, Distributed Systems, Bioinformatics**

I. Introduction

Modernization is an inevitable trend, as technology continues to develop for a variety of fields and purposes. These developments have led to increased interest and investment in smart healthcare systems, and subsequently, smart hospitals viewed as the future of patient treatment. This increasing interest can be attributed to a variety of factors, but there are two critical factors that are of significant importance in contributing to this trend.

One factor that has led to these significant breakthroughs in the field has been the development and implementation of the Internet of Things, or the IoT, one of the most successful byproducts of what is widely known as the Fourth Industrial Revolution. To put it simply, the IoT is essentially an interconnection of smart devices, objects and the like that can exchange data through the internet without requiring any form of human interaction, and sometimes, even beyond the scope of the interaction of two machines. Smart technology can be considered as one of the most prominent examples of the practical use of IoT, as numerous devices (like lighting, home security, entertainment devices in a smart home) can be supported in a variety of ecosystems, and controlled by central devices that interact with said ecosystem, and of course, healthcare systems happen to be one of the more prominent beneficiaries of this new technology. Through the IoT, medical and healthcare systems are able to gather data on patients and other uses from a variety of devices like medical sensors and imaging devices to aid in patient treatment and provide an improved quality of life, while also simultaneously reducing costs, and as mentioned earlier, reducing patient and physician stress and enabling a better interaction between patients and healthcare professionals.

Another factor that has led to significant breakthroughs in the field has been the rise and advent of streaming frameworks. Streaming engines, most notably those under the Apache banner like Storm, Spark or Apex, have been integral in analyzing data in large quantities, in real-time, which has enabled prominent companies and developers to potentially predict how users' wants and needs, and how said users plan on utilizing their services and systems. In the case of smart healthcare, streaming frameworks have the capability to analyze the increased amount of patient data to provide healthcare officials with the information they require with reliable and valid results that save time, to further improve their treatment methods towards their patients, better communicate with their patients, provide the correct medication, identify ailments and illnesses as they occur.

With all of these recent factors and developments in the field, the stage is set for the next major revolution in the smart healthcare field, and the potential of combining the IoT with streaming frameworks can lead to some truly innovative results. The goal of this paper is to provide one such solution. In this paper, we intend to develop a prototype of a system proposed in the previous year, that showcased how streaming frameworks, and more specifically, Apache Flink, which is widely considered to be the most reliable and dependable of those under the Apache banner, can be utilized to create a fully-functional, smart healthcare system that can provide great benefits to both patients and healthcare professionals alike. The implementation will be in the form of a doctor's report for this paper, to again showcase the potential of this project, and highlight the additions that can be made moving forward. Section 2 of our paper will be a view on similar works to this project, and section 3 will provide an insight to the benefits that Flink could bring to these healthcare system will highlight the system architecture and design. In section 4, we will highlight our experimental results, conluding the paper in section 5.

II. Related Works

Durga Amarnath M. Budida and Ram S. Mangrulkar [1] proposed a system that could potentially solve health issues using the IoT. Their proposed system makes use of a variety of smart sensors (including ones related to pulse rates and temperature) that collect raw data that is then pooled into a data server, where the data is further analysed, then converted into a graph analysis that can be accessed at any time from the server. The final goal was the development of a system that would enable all users, including patients and doctors, to access the system (separated between the two through credentials), wherein users can view their required data that has been uploaded and displayed in the form of an SQL table, and as mentioned previously, plotted into a graphical form that could later be used for further analysis and testing, thereby bridging the gap between healthcare systems and the IoT, which was their intended objective. We found their work to be very close to the system we proposed, and relatable to a previous paper we wrote last year for said proposed system. The key difference here would be our reliance on streaming frameworks, though it was not made clear the method by which they analyse their data.

Guoqiang Sun et al. [2] conducted a research analysis on the development of a Mobile Intelligent Medical Information System, which utilized the IoT. In it, they gathered a sizable amount of research content, based on mobile technology (such as RFID and various means of wireless communication), usage of a patient ID card (dubbed an IC card) that relied on mobile NFC technology, and research that utilized the usage of indoor navigation technology. They conducted a variety of interviews, site investigations, and other such methods of information gathering, then finalized the paper by highlighting their intended system features, and an outline of their interface design. This paper notes several prominent features of smart healthcare systems like booking appointments, doctor-patient interactions and so on, and analyses how their system would end up benefitting the intended users.

Debojyoti Seth and Debashis Charkraborty [3] proposed a system that combines the usage of a Hidden Markov model and the IoT to create a hybrid driven smart healthcare system. In it, they explained that the hybrid system would synchronize itself with the bio-signals of patients that are acquired through a variety of sensors and other hardware such as an Electrocardiogram (in a non-invasive manner to protect the patient's right to privacy). Signals are acquired from areas of interest around the body, like the scalp and various muscles (depending on the patient), then run through a Markov model to predict possible user necessities and even future treatments. While the Markov model was not something considered for our work, it is a consideration for future experimentation.

Mauricio Gonzalez-Palacio et al. [4] proposed an IoT approach to improve sterilization procedures in hospitals, using electrical instrumentation and processing of steam quality, through delving into the nature of bacteria and their effects on hospitals, as well as a description of water steam and the sterilization process (and how the two are intertwined), and as such, may seem like an unrelated work, but the idea of utilizing the IoT for the improvement of a hospital system and structure still fits in line with the core of our work.

Muthuraman Thangaraj et al. [5] described a smart, autonomous hospital system, enabled by the IoT and by utilizing technology drivers. One of their goals was to sync hospital systems and clinical databases, among other related systems, with smart data objects, all of which provide a variety of functions, from enabling access to the system to patients, to health measuring devices that can report on the oxygen levels in a patient's blood or a patient's blood glucose level, the list goes on. The system also factors in the patient's ability to pay a fair fee for treatment, patient's comfort levels with regards to disclosing information, and so on, all through healthcare interoperability as they put it. After conducting some tests, they found that the system will be able to facilitate the connection between a hospital authority and medical devices, and through their user-friendly design and configurable infrastructure, hospitals should easily pick up and utilize their system quite efficiently, while conforming to all medical standards and configurations.

Paolo Bellagente et al. [6] presented a middleware approach for an IoT application to healthcare. They dive quite deeply into the usage of medical sensors as devices connected to the IoT, as we recommend later in the paper, and explore how the adoption of a Connected-Health-CH-paradigm can result in a reduction of costs and an improved quality of life for patients. They use the ISO/IEEE11073 data model as a reference for their work (which involves agents, in this case, producers, generating data and transporting messages between them and the managers, or in this case, the consumers), as well as the AMQP protocol to provide full interoperability between the intended clients (patients, etc.) and the MOM servers (Machine on Machine), and an AuthManager designed to ensure managers (in the case of the model) have rights to data access and configuration, in conjunction with the above protocol. Their results showed an enhanced flexibility and scalability with regards to the IoT scenario, at a lower cost and higher speeds.

Luliana Chiuchisan et al. [7] provided a case study on the potential of adopting the IoT technologies developed today for healthcare into modern healthcare systems. In it, similar to previous works, they highlight the benefits of attributing the IoT to healthcare systems, citing the Romanian Health System as an example where new technologies that adopt the IoT could be implemented to improve and match their standards to that of other, more advanced European healthcare systems. The authors also propose an Intensive Care Unit for monitoring patients, through sensors to monitor a patient's vital functions, as well as other devices such as a bedside monitor, an XBOX Kinect™ which can help detect movement, but only to prevent certain situations, and a sensory board to measure temperature, humidity and the like, among other devices and features.

Finally, Gennady Smorodin et al. [8] talk about the evolution of the IoT can lead to advancements in healthcare in Russia, particularly with the development of Augmented Reality methods that essentially transform patients into what

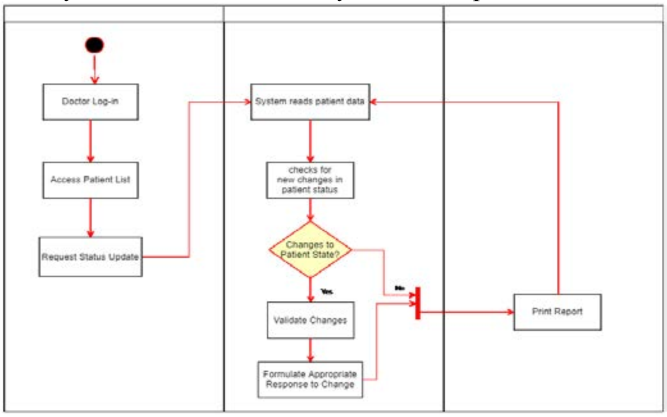

Figure 1. Use-case of a doctor requesting patient data in the system.

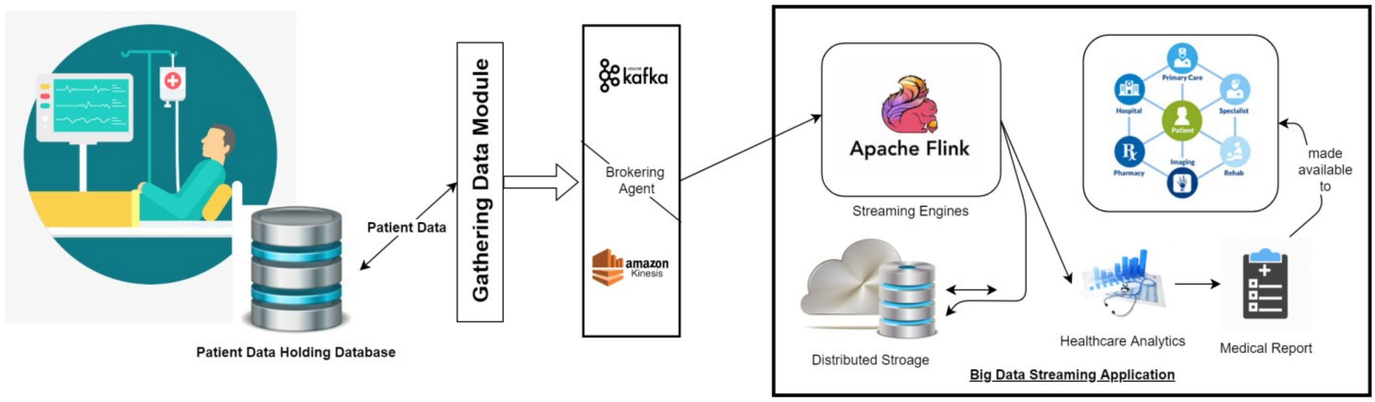

Figure 2. Our Synthesized Database. The initial fields considered for better understanding of users of the system.

they dub as an "e-patient", or a sort of digital patient with all of that patient's files digitized, and their health monitored within the IoT.

This section introduced a few of the related works that we drew inspiration from for the writing of this paper. Most of these works above rely on a variety of different technologies or focus on a specific aspect of a smart healthcare system, with the common linking thread between them being the usage of the IoT. However, our work differs from the following projects due to our utilization of streaming frameworks, or more specifically, our usage of Apache Flink for data analysis and the prcessing of patient data. That idea is relatively new in our field, and although there have been a few theoretical concepts of such an addition to healthcare systems, there has not been a complete practical implementation of such a concept, which we intend to implement in this project moving forward.

III. System Design

A. System Scope and Objectives

This system is intended to be a partial implementation of a system we proposed previously. The system will collect the patient data from a dynamically changing database, analyse said data in real-time (including in the case of any surprising changes or prompts to the patient state), then present the intended user (in this case, a physician or healthcare official) with a report of the patient's current state, and the suggested steps moving forward, whether said case is life-threatening or not. Figure 1 showcases an outline of how the system will function, where the a healthcare official, a doctor in this case, would log in, access the hospital's patient list, and requests the information of a specific patient for any potential changes. The system will search through the patient's data for any status updates to the patient's condition. If a change to status has occurred, that data is validated, analysed, and formulates a proper response for the doctor to follow, and prints a report. If no changes have occurred, then the system prints the report, stating that no changes have been made. Based on our past work, and the related works, the objectives of this project should be to:

-Provide doctors with a method to treat patients as needed
-Reduce doctor's stress levels
-Allow varying degrees of access to patient records depending on the user
-Maintain large amounts of big closed data and the capability to analyse it accordingly

B. System Architecture

In this sub-section, we will introduce the proposed general system architecture and the working flow of interaction of each component to each other and outside systems in a seamless way. Data is collected from a database that is continually updated, pooled together, and then fed into the data broker, which in the case of this project is Apache Kafka. The data broker will sift through the gathered data to ensure its validity, and upon completion, Kafka pipelines the data towards the streaming framework, which in this case, is Apache Flink, where the data is thoroughly analysed, before the results are sent back to the system, and the healthcare official who requested the information, while also being saved in a distributed storage as needed. Analysis of data will continue, even in the sudden case of a patient's change of state, where the data will be updated instantly. Once received the system prints this obtained data in the form of a medical report that is made available to all intended stakeholders, which in the case of this project, are the relevant healthcare officials. This data can be accessed by said stakeholders at any time, and when required, the process will automatically repeat itself. Figure 2 above illustrates the process in its entirety, starting from the module collecting and gathering data, siphoning that data towards the data broker, which reads and validates the required patient data, before pipelining the data towards the streaming framework, Flink in the case of our project, typically through a project manager, a step that will be explained later in the paper. From there, Flink reads and analyses the patient data as seen in the figure, and once that process is completed, Flink outputs the data in the form of healthcare analytics, and from there, makes the data available to the user who requested it, namely the doctor, while also storing the data for later use.

IV. Experimental Evaluation

The proposed system prototype is designed to siphon data from an established patient database, which is fed through to Flink by way of the broker Kafka, to analyse a patient's data in real-time, then produce a form for the intended user once the analysis is complete. This design is made to be a simulation of a case where a doctor needs to request a patient's data for any change in condition or receives in the case of a sudden change in a patient's medical status. For the sake of this simulation, we have synthesized our very own database to control all factors that could occur during analysis, and to reduce the number of external factors that could affect the data analysis, as the available open-source databases available online provide a substantial number of unrelated fields that would potentially affect the final result, as explained above, or is irrelevant for this experiment.

Patient Name	Age	Gender	DOB (M/D/Y	Insurance	Reason for Hospitalizati	Condition
Mark Redford	19	Male	05/25/00	Yes	Broken Leg	Stable
John Cash	43	Male	07/31/76	Yes	Slight Fever	Stable
Nina Holo	21	Female	09/12/98	Yes	Blood Diagnosis	Stable
Troy Day	34	Male	01/05/85	No	Erratic Breathing	Critical
Cassie Apollo	13	Female	03/15/06	Yes	Slight Fever	Stable
Christina Belval	65	Female	06/23/54	Yes	Dislocated Shoulder	Stable
Lisa Oh	27	Female	12/29/92	No	Head Trauma	Stable
Rashid Davis	22	Male	08/21/97	Yes	ACL Tear	Critical
So-Min Kim	16	Female	05/13/03	Yes	Slight Fever	Stable
Johann Craus	51	Male	04/24/68	No	Erratic Breathing	Critical
Kris Ducard	31	Male	07/08/88	Yes	Bronchitis	Stable
Samantha Thon	18	Female	09/12/01	Yes	Head Trauma	Stable
Zainab Ruman	45	Female	05/14/74	No	Internal Bleeding	Critical
Corrian Merriwe	23	Female	09/24/96	Yes	Lung Failure	Critical
Tilda Chrome	35	Female	10/13/84	Yes	Slight Fever	Stable
Jonathan Marku	19	Male	04/29/00	Yes	Heavy Cough	Stable
Michael Fink	56	Male	08/24/63	No	Eye Infection	Stable
D'shawn Harris	35	Male	05/21/84	No	ACL Tear	Critical
Nathan Truemar	27	Male	11/13/92	Yes	Erratic Breathing	Critical
Maya Yurikawa	18	Female	11/14/01	Yes	Broken Arm	Stable
Sergei Vladokov	40	Male	12/31/79	No	Hamstring Tear	Stable
Tim Werner	25	Male	02/14/94	No	Strained Back	Stable
Ivan Irishikov	26	Male	06/04/93	Yes	Head Trauma	Stable
Roman Ahmado	80	Male	07/15/39	No	Internal Bleeding	Critical
William Thomas	67	Male	09/18/52	No	Anemia	Stable
Judy Teach	14	Female	11/11/05	Yes	Shock-induced Coma	Critical

Figure 3. Synthesized Database used for the evaluation purposes: The initial fields considered for better understanding of users of the system.

As seen above in Figure 3, we synthesized a patient clinical database, based on online healthcare records, with an initial amount of approximately 50 simulated patients (the names were selected randomly, with no relation with anyone of the same name), with a set of fields expected in a typical healthcare system database, including current condition, diagnostic of disease, presence or lack of insurance (which we found to be an essential point in the related works), among others. Information in this database was fed to Flink for analysis through our data broker, Apache Kafka. Though there are a variety of data brokers to choose from, such as Amazon Kinesis and the like, we are dealing in our paper with a local data cluster, making Kafka the more ideal choice for this project [9].

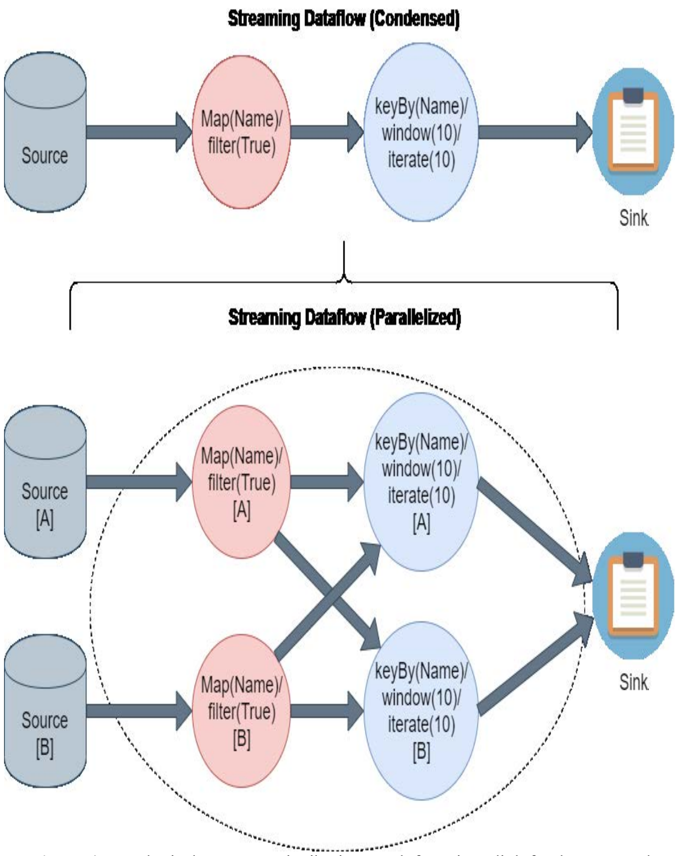

Figure 4. Topological Operator Distribution graph for using Flink for the proposed system.

As explained, Kafka begins reading the data from the local cluster to confirm its validity, and since the simulation was trying to mimic how a doctor would act, we focused on having Kafka read the more crucial fields, namely the condition and reason for hospitalization, as they are the ones who are most guaranteed to change in the case of a situation happening to a patient. Kafka typically reads data in the form of consumers and consumer objects, where the data as mentioned earlier being the objects that will be read by Kafka, split between at least two consumer groups (at least two key fields to be read) that read and validate the data, and print the output requested. From there, the next step, as mentioned in the System Design section, is to pipeline the data through to our streaming framework, Apache Flink. Similar to the choice of data broker, there are a variety of different streaming frameworks, most of which like under the Apache banner, that would have been very suitable for this project, but we chose Apache Flink for a number of reasons, namely because of its reputation as being an open-source streaming platform, capable of handling both batch and stream data processing, utilizing a framework that can handle data streams at extremely high speeds, while providing a low fault tolerance, high throughput, proper flow control, excellent memory management and the ability to be integrated across multiple coding platforms, such as Java, Scala and Python, among other reasons [10].

PATIENT REPORT

PATIENT NAME
Mark Redford

PHYSCIAN IN CHARGE
Dr. Samuel Han

DATE
07/24/19

REASON FOR HOSPITILIZATION	Broken leg, caused by a collision with a fellow player during training.
CONDITION	Stable.
MEDICAL HISTORY	Fever, Mild Cold.
PRESCRIBED MEDICATION	Aspirin, Mild Painkillers.
CHANGES TO CONDITION	None.
RECOMMENDED ACTIONS	Continue to monitor on an infrequent basis until patient's release.
DOCTOR'S NOTES	Patient's leg was not severely damaged and will recover on schedule. He otherwise remains in peak condition, and no other actions are recommended at this point in time. Would recommend more rest, even post release, for two weeks at the very least.

Figure 5. Medical report prototype displayed for the doctor.

To facilitate the data pipeline, we utilize a software project management tool, Apache Maven [11], which supports multiple programming platforms, and while Flink is flexible and can run across a variety of platforms like Python and Scala, we are primarily working with Java. Maven helps simplify the build process while providing the information we need with regards to our project experimentation. Afterwards, Flink receives the validated data from Kafka, while also reading the required patients data directly from the database, as shown in Figure 4, in preparation for data analysis. Using the patient name, Flink maps each element of the patient's record, based on both sources, to a streaming dataflow, consisting of both streams as well as transforming operators, while simultaneously filtering through the data as that happens to confirm that there have been no changes made. Each of these operators are split into subtasks that all act independently of each other and in parallel, and in the case of our project, we have five subtasks, namely, mapping the data which we mentioned, filtration, setting a specific window of time for data to arrive and be analyzed (a session window with a 10 second activity gap), a key for sorting purposes and an iteration command so that the analysis will loop back after completing a set number of times (10). Master processes coordinate the distribution of subtasks, while the worker processes execute them in their respective task slots allocated to each worker, which contains the assigned tasks as well as a subset of resources and a memory slot. Once the process is complete, the system prints a form to the doctor's device, wherein the doctor in this simulation can verify the patient's state, as seen in Figure 5, and determine the next course of action. Finally, to simulate a case where a sudden change happens to a patient when the doctor isn't present, during the analysis of data, we manipulate the database manually by altering some of the information (changing the condition from stable to critical, etc.), to produce a form, notifying the user (doctor) that something has happened, and the suggested course of action.

V. Conclusion

Healthcare systems are and will continue to be a topic of vast importance and innovation, and the prototype we developed was designed with that intention in mind. Though the idea seems simple enough, the process that was undergone during testing was long and rigorous, and at the end of it, all gave us a powerful insight into the next step of this project.

This prototype still requires a bit of tweaking and enhancement, of course, but this first step is promising, and could potentially aid doctors in their endeavors with regards to patient care and treatment. On top of that, the idea of adding the concept of the IoT to the project would significantly enhance the capabilities of not just this small prototype, but the project as a whole.

This paper was developed as a sub-section of a larger smart healthcare system project. The principle motivation behind this system is to provide doctors and healthcare officials with the basis of an efficient system that they could make use of to provide fast and efficient treatment for all. Future considerations are to continue development on the next sub-sections of the larger project, with the prospect of realizing a fully-functional system later on.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Science and ICT (No.2017R1A2B4010395).

References

[1] D. A. M. Budida and R. S. Mangrulkar, "Design and implementation of smart HealthCare system using IoT," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-7.

[2] G. Sun, F. Yu, X. Lei, Y. Wang and H. Hu, "Research on Mobile Intelligent Medical Information System Based on the Internet of Things Technology," 2016 8th International Conference on Information Technology in Medicine and Education (ITME), Fuzhou, 2016, pp. 260-266.

[3] D. Seth and D. Chakraborty, "Hidden Markov model and Internet of Things hybrid driven smart hospital," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-7.

[4] M. Gonzalez-Palacio et al., "Internet of things baseline method to improve health sterilization in hospitals: An approach from electronic instrumentation and processing of steam quality," 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, 2018, pp. 1-6.

[5] M. Thangaraj, P. P. Ponmalar and S. Anuradha, "Internet Of Things (IOT) enabled smart autonomous hospital management system - A real world health care use case with the technology drivers," 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, 2015, pp. 1-8.

[6] P. Bellagente, A. Depari, P. Ferrari, A. Flammini, E. Sisinni and S. Rinaldi, "M3IoT — Message-oriented middleware for M-health Internet of Things: Design and validation," 2018 IEEE International

Instrumentation and Measurement Technology Conference (I2MTC), Houston, TX, 2018, pp. 1-6
[7] I. Chiuchisan, H. Costin and O. Geman, "Adopting the Internet of Things technologies in health care systems," 2014 International Conference and Exposition on Electrical and Power Engineering (EPE), Iasi, 2014, pp. 532-535.
[8] G. Smorodin et al., "Internet of Things: Modem paradigm of health care," 2017 21st Conference of Open Innovations Association (FRUCT), Helsinki, 2017, pp. 311-320..
[9] Posta, C. (n.d.). What is Apache Kafka? Why is it so popular? Should I use it?. [online] TechBeacon. Available at: https://techbeacon.com/app-dev-testing/what-apache-kafka-why-it-so-popular-should-you-use-it [Accessed 30 Sep. 2019].
[10] Pal, K. (2016). The Importance of Apache Flink in Processing Streaming Data. [online] Techopedia.com. Available at: https://www.techopedia.com/2/31476/trends/big-data/the-importance-of-apache-flink-in-processing-streaming-data [Accessed 30 Sep. 2019].
[11] K, A. (2018). What is Maven and what are its benefits? - The Linux Juggernaut. [online] The Linux Juggernaut. Available at: https://www.linuxnix.com/what-is-maven-and-what-are-its-benefits/ [Accessed 30 Sep. 2019].

Rhaed Khiati received his B.S. degree in software engineering from Alhosn University, Abu Dhabi, UAE, in 2015. He is currently pursuing his master’s degree in computer science from Hanyang University, Seoul, South Korea. His research interests include Cloud & Distributed Computing, Big Data Analytic Engines, Stream Processing Frameworks and Smart Healthcare Technologies.

Muhammad Hanif received his B.S. degrees in computer and software engineering from University of Engineering and Technology (UET), Peshawar and his master and Ph.D. degree in Computer Science and Engineering from Hanyang University, Seoul, Republic of Korea. He is currently working as a post-doctoral research associate in department of Computer Software Engineering at Hanyang University, Seoul, South Korea. His current research interest includes Cloud & Distributed Computing, Big Data Analytic Engines, Stream Processing Frameworks, Distributed Scheduling, Service Orchestration, and Block-Chain Technologies.

Choonhwa Lee received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree in computer engineering from the University of Florida, Gainesville, FL, USA, in 2003. He is currently a Professor with the Division of Computer Science and Engineering, Hanyang University, Seoul. His research interests include cloud computing, peer-to-peer and mobile networking and computing, and services computing technology.