International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at :www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET

Applications of Java in Real-Time Data Processing for Healthcare

Chinmay Mukeshbhai Gangani
Independent Researcher, USA

ABSTRACT

People's personal health can be effectively monitored thanks to modern technologies. One option for tracking data on personal vital signs is to use sensors that are based on Bluetooth Low Energy (BLE). In order to assist diabetic patients better manage their chronic condition on their own, we present a customized healthcare monitoring system in this research that makes use of a BLE-based sensor gadget, instantaneous data processing, and machine learning-based algorithms. For the Internet of Things (IoT)-based smart e-Healthcare to be served effectively, real-time service has become essential. Numerous approaches have attempted to advance this area of technology, but they have fallen far short in integrating open and lightweight IoT-based frameworks. Hundreds of linked sensors must be deployed for scientific applications in healthcare, including body area networks, in order to track a host's health. The continuous stream of data gathered by all those sensors, which must be analysed instantly, is one of the main obstacles. Moving the gathered big data to a cloud data centre for reporting on progress and record keeping purposes is often required for follow-up data analysis. With less management, cloud computing offers a company a good structure and a decent cost. Monitoring and treating chronic illnesses and possible crises are the main goals of recent developments in sensor communication, sensor sensing, and microelectronics. Through message interceptions and the application of criteria based on the syndromic surveillance paradigm, we assess the quality of data in such systems using sophisticated event processing made possible by the Event Swarm programming framework. We think this is the first study to report on applying syndromic surveillance criteria to legacy clinical data streams in real-time. Our approach's viability is shown by our design and execution, which also illustrates the advantages of enhanced HIT system operational quality, including increased patient safety, lower risks in healthcare delivery, and perhaps lower costs.

Keywords :- Current Technology, Internet of Things (IoT) Real Time, Monitoring and Managing, Microelectronics, Sensor Sensing, Reduced Costs, Real-Time Application, Smart E-Healthcare, Bluetooth Low Energy (BLE), Streaming Data.

I. INTRODUCTION

The introduction of big-data technologies has continuously advanced healthcare research. Scientific applications in healthcare often use streaming input data produced by several dispersed sensors. These data are then processed by cutting-edge big-data frameworks and platforms. For instance, the Body Area Network, [1], which is often used to access, track, and assess an individual's health state in real time, has long been infamous for requiring a lot of computational power to analyse gigabyte of data in real time [1, 2]. Among other things,

these data are gathered via properly designed sensors that sample the real-time signals of blood pressure, body temperature, heart rate and respiration, chest sound, and cardiovascular health. Traditional paralleled processing frameworks like Hadoop MapReduce, [2], Pregel, and Graph Lab are physically and functionally restricted in their ability to handle stream large data in real-time. Their designs are mostly fabricated to access and handle the static input data, which is the main source of difficulties [2, 3]. When the input data comes in a stream flow, no built-in iterative modules may be utilized. Furthermore, situations where the streaming data being used come from diverse sources and have varying arrival speeds are beyond the capabilities of the current frameworks [2, 3]. When a person's behaviours vary, scientific applications in healthcare adjust the frequency of data collection. For instance, compared to while a person is jogging or swimming, much less data may be gathered when they are sleeping [3].

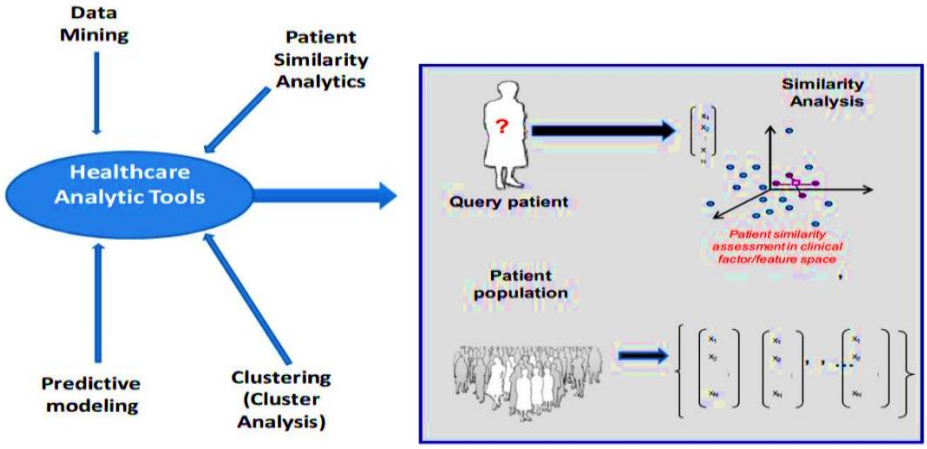

Fig. 1 Big Data Analytics in Healthcare. [2, 3]

Their lack of compliance with the real-time component is similarly evident in real-life scenarios. Numerous open-source, lightweight frameworks based on IoT protocols, including Node.js, Johnny-Five, SperialPort, PubNub client, EON.js, Chart.js, [5], Express Server, and Socket.io, may offer strong substitutes for the IoT-based smart e-Healthcare devices and services that are currently in use [4, 5]. The following innovations are made possible by this paper:

1. Using lightweight technologies for Internet of Things-based e-healthcare in a context with limited resources,
2. The Node.js framework is implemented using a number of client-specific Java script tools, such as Chart.js, EON.js, [4, 5], and Johny-Five, among others,
3. Combining e-health sensors in a low-footprint situation with features like Java script,
4. Using the Standard Firmata algorithm to enable interaction with the Internet of Things sensor device, and
5. Two distinct methods for determining the same result, such as BPM, from the same intermediate layers of the suggested architectures are shown, offering a boilerplate-oriented intervention component [5].

A fresh approach to diabetes care is provided by recent developments in ICT (information and communication technology) and innovative biosensors that can monitor a patient's status in real time. Continuous glucose monitoring (CGM) sensors and self-monitoring blood glucose (SMBG) portable devices allow diabetic individuals to track changes in their blood sugar levels and react promptly by taking the necessary action [5, 6]. The findings demonstrate that keeping an eye on patients' blood sugar levels may help them better manage their condition and enhance diabetic care. The greatest option to enhance diabetes control is a glucose monitoring system that consists of sensors that are the gateway (smartphone), and a cloud system. It uses a smartphone as a gateway to get sensor data from a sensor node that is affixed to the body [18]. Bluetooth Low

Energy (BLE) is the best option for wireless technology and low-power operation for the sensor node, [5, 6], which are necessary for the connection between the node that collects data and the smartphone.

1.1 Real-Time Data Processing

Companies, governments, industrial sectors, and technological research are now producing and sharing an enormous quantity of data [4, 5]. Ninety percent of the 2.5 quintillion bytes of data that are generated globally each day (one exabyte is equivalent to one quintillion bytes, or one exabyte is equivalent to one billion gigabytes) are unstructured [5].

This is unquestionably the big data are due to the startling volume of complicated data generated by every device, anywhere, at any time [6]. Big data analytics may be used as an operator for vendors' competitive advantage because of the expanding data trend. The amount of healthcare data, which includes various and changeable text formats, sounds, and pictures, is growing daily, making its processing and storage a vital but difficult problem. The healthcare sector uses big data analytics in a number of ways [6, 7]. Fahim et al. created a system that may encourage people to have active lives and suggest beneficial solutions to them.

The suggested methodology made use of a cloud system, machine learning, and big data infrastructure to quickly retrieve the vast amounts of sensory data from smart devices for suggestions. To address the issues related to real-time large data analysis, [5], framework for a lambda architecture-based cardiovascular disease (CVD) prediction system. The suggested system might use a range of health data sources, including clinical, genetic, and lifestyle data, to forecast illnesses and ailments [6, 7] .

1.2 Machine Learning–Based Algorithms for Diabetes

T2D is a progressive disease where the body progressively loses the ability of the pancreas to generate adequate insulin and/or becomes resistant to the typical effects of insulin [5, 6]. Since the majority of T2D patients show no symptoms, screening for pre-diabetes and diabetes should be taken into consideration in those who ≥ [7].

1.3 45 years of age, particularly in those with a BMI ≥

(25 kg/m2). This is most likely due to the fact that as individuals age, they tend to lose muscle mass, exercise less, and gain weight [7, 8]. Therefore, it is essential to have an autonomous prediction model that alerts individuals to their risk of acquiring diabetes in the future so they may take preventative measures. Based on a patient's present state, diabetes may be categorized using machine-learning algorithms [8] . By building datasets with DB tuples and the class labels that correspond to them, supervised learning classification models are well-liked machine-learning techniques that are used to build a classification model from a set of trained models to be used to predict class labelling for the [8, 9] provided data. In general, the following actions must be taken by supervised learning classification models:

1. Data collection;
2. Feature extraction;
3. Selection of a machine-learning algorithm;
4. Model construction using the selected algorithm; and
5. Evaluation of the algorithm accuracy.

The literature still contains a variety of research, [9, 10], despite the fact that IoT data analytics is a relatively young topic. Here is a summary of a few relevant studies. An automated human pain assessment technology that uses facial expressions to track discomfort in real time. Application model for future IoT healthcare systems [8]. The model's advantages, disadvantages, and applicability to a wearable Internet of Things healthcare system are examined. new uses for IoT in healthcare. The research describes the technological requirements and looks at the whole end-to-end solution for every application. In a prior investigation, an armband was used to monitor the end user's forearm muscles' real-time EMG signals. After developing and integrating the hardware and software architecture, [4, 5], they used two sets of trials that included a sequence

of instructions carried out in several rounds to test the system. According to the findings, the system that was presented scored 85.7% and 92.9% on the two tests for accuracy.

1.4 The Proposed Real-Time Data Analytics Architecture for Smart Healthcare

As seen in Figure 1, the suggested real-time data analytics architecture for smart healthcare is divided into two main sections: the horizontal domain (Kafka, Spark, [5, 6], NodeJS, and MongoDB) and the vertical domain (SDN-based WSN and an RFID structure). The WSN's source sensing node (SN) senses patient-related ECG data and sends it to the gateway, which uses a TCP connection to send it to the Kafka platform [4 ,6]. Three connected consumers receive incoming data from the Kafka messaging system, as shown in Figure 1. The Spark platform for real-time data processing is the first Kafka consumer [5, 6]. The MongoDB database, which houses the incoming stream, is the final consumer. The second is a NodeJS web application that displays the patient's data in the web browser.

1.5 Radio-frequency identification

RFID is a popular IoT-enabled device for gathering data [7, 8]. The RFID reader integration with the SN is shown in Figure 2 for the Riverbed node model. Although the RFID.

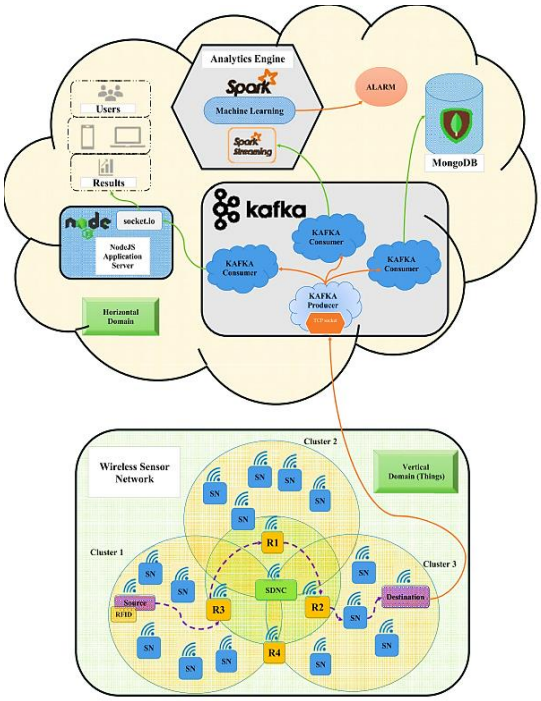

Fig. 2 An overview of the suggested architecture for Internet of Things (IoT) data analytics. [8, 9]

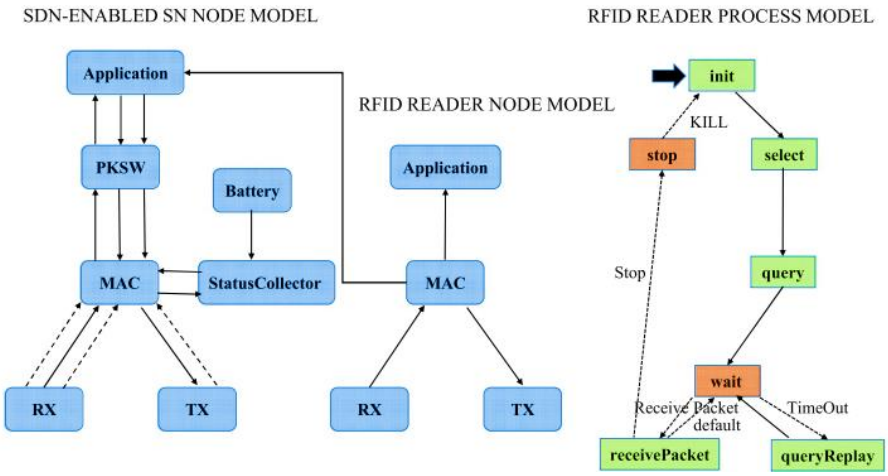

Fig. 3 RFID (radio-frequency identification) reader node architecture and RFID reader process model for sensing nodes (SN). [8]

In labs and other healthcare settings, health messaging is well-established and robust. The findings, evaluations, and other communications that these institutions provide are usually formatted according to the Health Level 7 (HL7) v2 guidelines [8, 9]. Clinical (human) interpretation is now the main use for such data. However, since this messaging is organized, it is possible to connect a CEP engine to the stream of HL7 v2 messages that a laboratory sends and receives. After that, CEP may be used to automatically identify clinical concerns of interest and problems with data quality, [9], as well as to monitor communications in real time. Such monitoring may lower the chance of patient injury and increase the safety of the HIT systems. In order to detect anomalous behaviour's that point to failure or problems with the quality of the data, we use CEP methods, [10], which are represented in the Event Swarm framework, to older streams of HL7 v2 messages.

In order to position the unique features of Event Swarm and to make sure readers are aware with the technology, an overview of CEP capabilities and semantics is provided. When such behaviors are detected, an operator or maintenance organization is notified so they may look into the issue and address it as soon as feasible [10, 11]. The technology helps to enhance the safety of HIT components and, eventually, patient safety by enabling health organizations to continually check the quality of data transferred and, therefore, the HIT systems from which the data originates [11, 12]. Outages are also quite expensive because to the high demand and high expense of the equipment utilized to generate laboratory messages. Thus, possessing such real-time analytics and alerts also has a significant financial value. The newly developed syndromic monitoring technique for early HIT system failure detection is the driving force behind this study. They use a static data set to show the viability and effectiveness of such monitoring. We expand on this work by demonstrating how the Event Swarm framework can be used to deploy the surveillance in real time against current data streams. Additionally, our technique expands the breadth of the syndromic surveillance described in [4] by using the HL7 data type.

II. IMPROVING PATIENT SAFETY AND REDUCING COSTS: USE OF IT FOR MONITORING

Health IT and Patient Safety

The growing use of HIT to enhance patient care raises many new issues, according to a recent Institute of Medicine paper. Although the study emphasizes the advantages of HIT, such as the markedly higher quality of healthcare and the decrease in medical mistakes, it also offers some evidence of the unintended negative effects of HIT on patient safety [4, 5] . It issues a warning that HIT has the potential to further complicate the already complex process of providing healthcare if it is developed and used improperly [6]. This might also result in unforeseen negative outcomes, such as incorrect dosage, a failure to identify life-threatening conditions, and treatment delays brought on by subpar human–computer interactions or data loss.

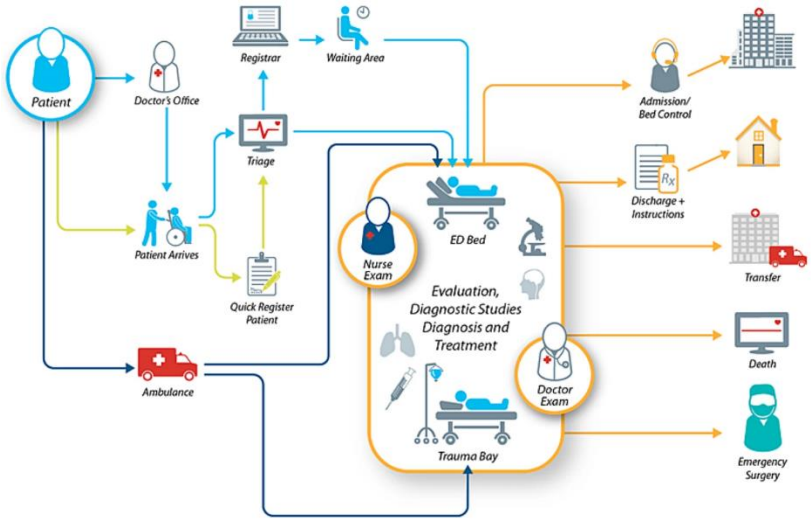

Fig. 4 Using it for Monitoring to Increase Patient Safety and Lower Costs. [5]

Health IT and Cost

By letting physicians to concentrate on providing treatment instead of spending time on tedious paperwork, HIT lowers the cost of healthcare delivery [4, 6] . Automation of healthcare processes also lowers costs.

Syndromic

Laboratory Orders and Results

The viability of using a syndromic monitoring technique to identify health system failures in order to address the early identification of HIT system failures [8, 9]. The scientists sought to see whether a comparable method might identify HIT failures, since syndromic monitoring is often utilized in the early identification of disease outbreaks [8]. They concentrated on identifying HIT failures in a tertiary hospital's laboratory information system (LIS), searching for irregularities in the data's structure (such as missing values) and semantics (such as unexpected values) [7]. The authors conducted research on a one-year dataset from the hospital, simulating four different kinds of HIT failures and used performance statistical analysis of the LIS data to identify those failures.

III. REAL-TIME ANALYTICS AND EVENTSWARM

A programming system to perform sophisticated event processing, or more recently, real-time analytics, is called Event Swarm. Specifically, [7, 8], we provide an overview of CEP and Event Swarm in this section.

A. **Origins:** The analysis of streams of distinct information components, or events, is known as CEP. An event is a record of an observation made by a particular individual or system, signifying an occurrence of importance [8, 9] . The use of correlation, aggregation, and abstraction to raw events streams is highlighted by the complicated adjective, which enables systems to identify intricate behavioural patterns by analysing the streams.

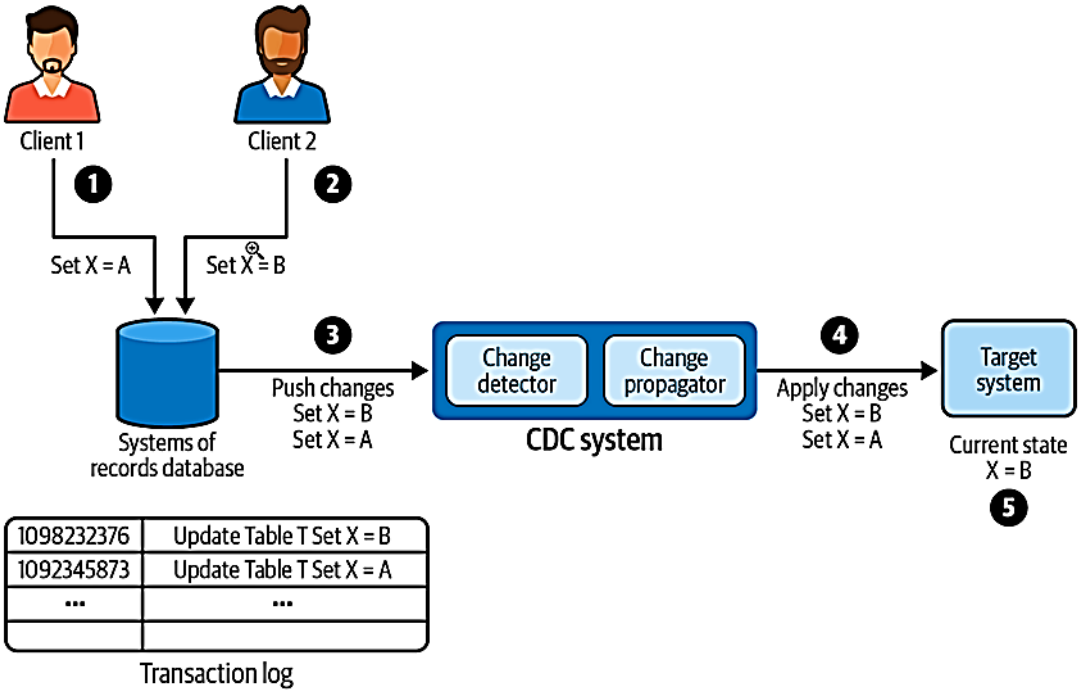

Fig. 5 Theoretical view of a CDC system. [8, 9]

B. **Features:** It is considered that all such systems, as well as many systems that came before them, have the fundamental capacity to match the characteristics of particular occurrences. The capacity to identify higher level patterns via correlation, aggregation, and abstraction is the primary characteristic of CEP systems [8, 11]. Some of these capabilities are described in the next subsections.

C. **Event Expressions:** A CEP system's basic building component is an event expression, which enables the setting of matching criteria for a single event [13]. Event expressions, such as *(e. color = red),* are based on static analysis of individual event properties in the most basic circumstances.

D. **Filtering:** Filtering is a popular technique used to limit the size of analytics in the increasingly large event streams connected with CEP. When an event expression A is applied to a stream or streams, a filter removes events that don't fulfil a from further processing. Depending on the system's event expression capabilities, [12], as covered in the previous paragraph, filtering event expressions may vary in complexity from simple static examination of event properties to comparisons against intricate calculated abstractions [13, 14].

E. **Pattern matching:** One fundamental correlation mechanism is pattern matching, which describes the system's capacity to match event patterns via the use of logical functions and sequence over event gestures, [15], such as:

- AND means that a pattern A AND B matches events e1 and e2 in pairs so that e1 satisfies A and e2 satisfies B [15, 16].
- OR, which is when a pattern A OR B corresponds to one or more occurrences e1 or e2 such that e1 or e2 fulfils A or B [16, 17].
- XOR, or pattern A XOR B, assigns individual occurrences e to either satisfy A or satisfy B, but not both [19, 20].
- repetition, where a pattern A{n} corresponds to n instances of A. Some people believe that repetition is a specialty of AND.

- ・sequence, that is, a pattern A followed by B matching pairs of events e1 and e2 such that e1 in time happens before e2 and e2 fulfils b.
- Using a comparison based on vector clocks, the causal sequence A -> B matches pairs of events e1 e2 such that e1 fulfils A and e2 satisfies B and e1 causally precedes e2.

F. **Complex events:** In a CEP system, [1, 19], a pattern match is frequently referred to as a complicated event. A complicated event is a series of occurrences that point to an interesting behaviour.

G. **Time and ordering:** The sequence of events in time may be controlled in a number of ways, as the preceding section suggested. The fact that many event sources have varying time sources is a major challenge in CEP [9, 13]. Another issue is that events are often provided out of sequence due to the delivery delay.

H. **Sliding Time Windows:** An aggregation strategy known as sliding time windows restricts the range of correlation and abstractions to a window that shifts in proportion to the current time. All events whose timing falls inside a specified window prior to the present time are included in a sliding time window.

3.1 Statistical Analysis

In many CEP applications, statistical computations are crucial. Statistics on event characteristics or, sometimes, [9, 10], other data calculated during processing may be calculated using a CEP framework.

3.2 Language, Expressiveness and Complexity

There are two basic methods for constructing event patterns and abstractions in CEP: methods based on general-purpose programming languages and methods using domain-specific languages, [11], which are often based on SQL (e.g., Esper).

3.3 Distribution and Scalability

Distribution and horizontal scalability are necessary to apply real-time analytics efficiently due to the growing amounts of raw data created by systems nowadays [13, 14]. This vast amount of data is now often referred to as "big data in motion." The distribution and scalability strategies used by CEP systems vary greatly, and as was previously said, [16, 17], the strategy often hinges on the decision between a domain-specific language and a programming framework.

3.4 Near-real-time capability

As events reach the processing point, most CEP systems continually assess expressions, patterns, and other abstractions [15, 16]. They might thus be regarded as having near real-time capabilities. Certain CEP systems depend on periodically evaluating queries and storing events in an in-memory database. In addition to increasing latency, [18], this caching and periodic review usually raises the software's memory and processing capacity needs.

3.5 Event Swarm Architecture and Usage

Event Swarm uses an event-driven, near-real-time method for CEP, as was covered in the sections before this one. In the Java programming language, [17], it offers a variety of predefined abstractions and pattern components.

IV. APPLYING EVENT SWARM TO ORDERS AND RESULTS

This part introduces the technical aspects of HL7 v2 communications and explains how Event Swarm is set up to track the cost and data quality metrics mentioned in section II [16]. A. Messaging in HL7 v2 Laboratory orders, results, and other clinical information are mostly sent using HL7 version 2 (v2) messages in the

healthcare sector in the United States, Australia, and many other nations. For instance, in 2012, [18], more than 90% of pathology reports in Australia were sent electronically as HL7 v2 communications.

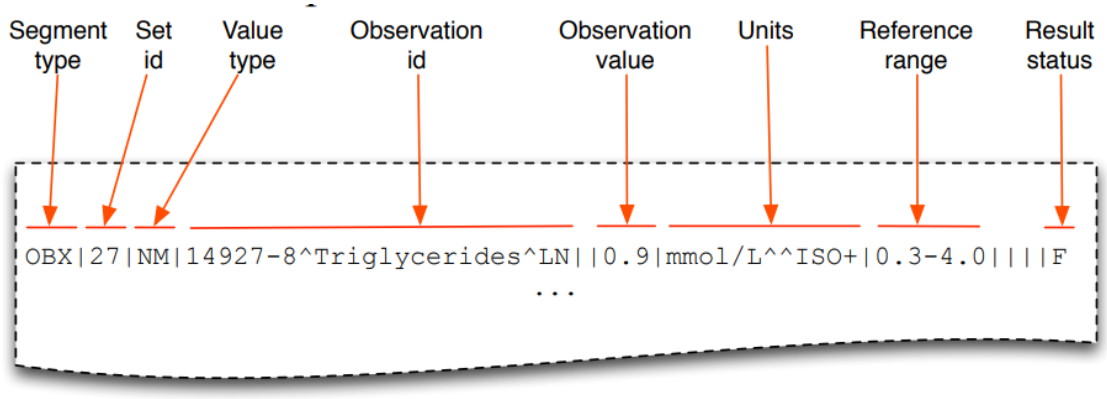

Fig. 6 HL7 v2 OBX Segment. [17]

4.1 Operational Context

This solution's presumed operating environment is a sizable integrated healthcare delivery organization that offers medical services to its members [17, 18]. Although the organization operates regionally, its reach is nationwide. Pathology orders are the implementation's main emphasis. The following metrics describe the organization's size in terms of laboratory orders and outcomes:

(i) One million pathology orders per day; [19],
(ii) Distinct test IDs (i.e., not free text descriptions) are included in around 50% of pathology orders;
(iii) About 2% of pathology orders include more than one test; [19, 20]
(iv) Each laboratory processes and average of 50,000 orders per day;
(v) It now takes an average of one day to uncover problems with the quality of laboratory data.

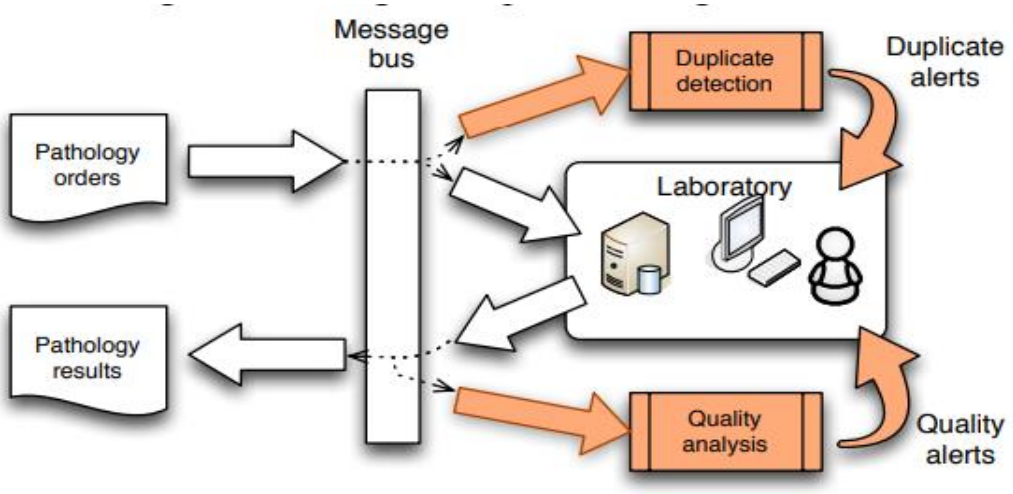

Fig. 7 Functional architecture. [20, 21]

Additionally, localization reduces the variance brought about by variations in content types and coding systems, such as the problems with the code extensions that were previously described. Additionally, before moving on to larger-scale deployments, [21, 23], localized deployment offers a foundation for gradual implementation and benefit assessment. Figure 3 shows this deployment design.

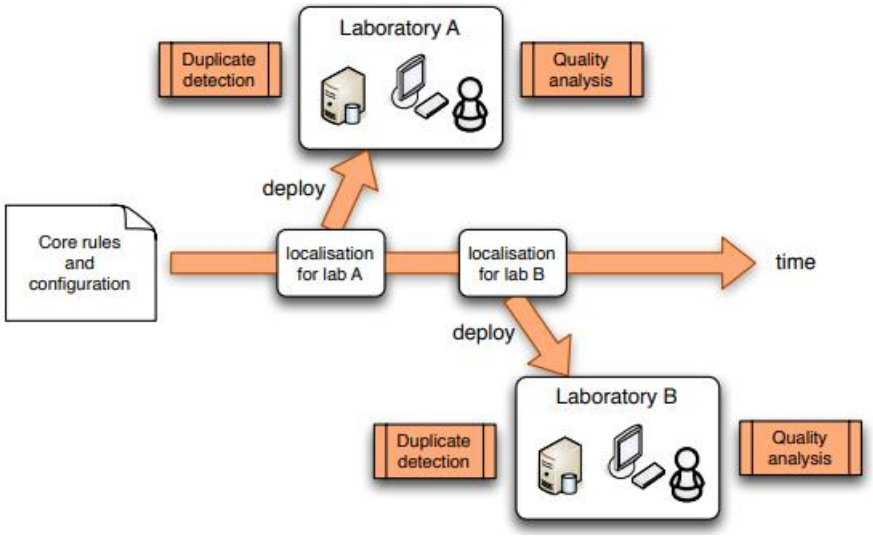

Fig. 8 Deployment architecture. [22]

4.2 Message interception

Message-oriented middleware is used in the operational environment to send and receive messages to and from the laboratory [23, 24]. Event Swarm has the ability to establish a direct connection with the middleware and get copies of every message sent across the company.

4.3 Data Quality Monitoring

The purpose of the data quality monitoring setup is to identify situations in which numerical observations correspond to the guidelines mentioned in section II, [23,24], specifically: At least two of the previous three numerical observations are more than two standard deviations from the mean; at least four of the previous five numerical observations are more than one standard deviation from the mean; and the last six numerical observations are all on the same side of the mean. 1. A numeric observation is more than three standard deviations from the mean [25].

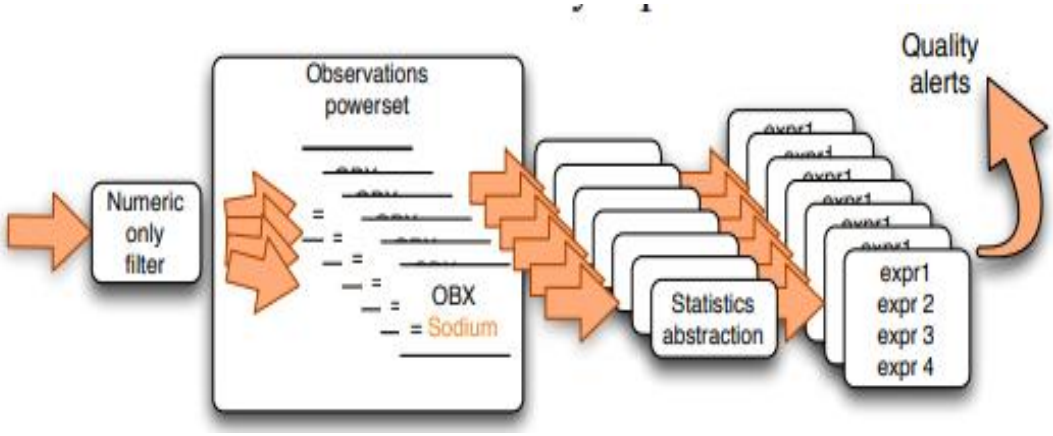

Fig. 9 Data Quality monitoring configuration. [24]

4.4 Duplicate Order Monitoring

To identify duplicate laboratory orders for a patient within a certain time frame, the duplicate order monitoring component is necessary [24, 26] . Since the duplicate detection scale is substantial, as was covered in section IV-B, we have examined the business issue to find solutions that would limit the size without compromising any advantages for the company.

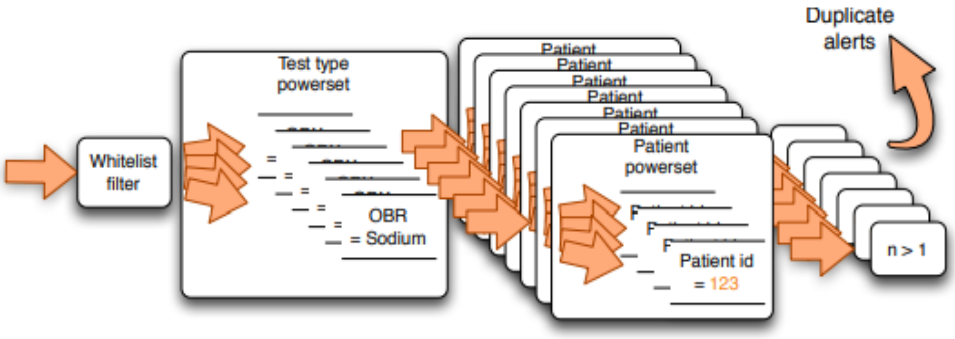

Fig. 10 Duplicate order monitoring configuration. [26, 27]

Most of this processing is done using off-the-shelf components from the Event Swarm framework, just as with the setup for data quality monitoring [28, 29]. Only two new components remain for this monitoring after the new HL7, renderer, and action components discussed in the previous section are reused here:

1. A component that matches the white list [27]. This is a straightforward addition to an already-existing Event Swarm class.
2. A predefined factory component to generate sliding time frames according to a test. The powerset uses this to generate test types for subsets.

The following is an estimate of this configuration's memory use per laboratory:
50000 ∗ 50% ∗ white list percentage ∗ size ∗ mean life. This is the number of daily events multiplied by the mean life in days of the white-listed tests, [22], and then decreased by 50% to exclude unstructured orders and the proportion of orders that match the white list.

A typical life of 10 days, a whitelisted volume of 30%, and an average order size of 1KB provide the following results: (50,000 ∗ 0.5 ∗ 0.3 ∗ 10 ∗ 1KB = 75,000KB or 75MB). As a result, our setup can simply run on a single processing node in memory [22, 23].

V. DISCUSSION AND FUTURE WORK

In this study, we demonstrate the usage of a CEP strategy to enhance HIT safety and efficiency by using real-time analytics on current healthcare messaging [21, 23]. This is accomplished by detecting duplication in pathology orders and by comparing statistical quality indicators with pathology laboratory data. When duplicates or quality problems are found, alerts are utilized to let the appropriate people know [24, 26]. Test data from messages used for conformity testing to Australian HL7 v2 pathological communications standards was utilized to create this solution. A chosen group of stakeholders has seen the solution in action. Our research indicates that the solution's memory needs are well-contained [21, 23]. On a single CPU core, the duplication detection and data quality solutions can handle 2000–3000 HL7 v2 messages per second each, which is more than sufficient for most deployment situations.

In order to transport, connect, and interact with the other modules of the specific deployed system model, we used the Arduino Uno platform as the pulse sensor handling sub-system in this research [23, 25]. The reasons are simple: use the EON and Chart.js frameworks to process, visualize, and analyse real-time sensor data streams and simulate the use patterns of a common resource-constrained embedded IoT device.

VI. CONCLUSION

As was previously said, Java script frameworks that enable the Internet of Things might be crucial in determining the real-time analytics service in an e-Healthcare setting.

Big Data Analytics has changed how healthcare consumers use sophisticated equipment to extract insights from clinical statistics and create informed decisions. The idea of providing patients with efficient data-driven services via predictions has been made feasible with the aid of Hadoop and MongoDB.

Additionally, a number of metrics, including the average data packet delivery and the CPU/memory consumption of the Android app, were used to analyze the performance of the BLE-based sensors. The distance between the smartphone and the BLE-based sensor had an impact on the performance of data packet delivery, while the data transfer to the server had an impact on CPU and memory use.

In order to improve the quality and safety of HIT systems and cut down on needless expenses brought on by the unintentional problem of repeated lab orders, this paper outlines the design and execution of a real-time analytics solution. The solution uses a syndromic surveillance technique to monitor HL7 v2 signals that clinical laboratory systems send and receive. The viability of the syndromic surveillance algorithm for HIT system monitoring is further shown in this research. Our solution is unusual in that it uses HL7 v2 message interception and analysis to create and deploy syndromic monitoring rules in a large and complex healthcare setting with minimum interruption to current HIT systems. This, in our opinion, is the first documented use of CEP technology for quality and safety monitoring in HIT systems, and it should open the door for further advancements in HIT system safety and quality using real-time analytics. A discussion of CEP systems' capabilities and semantics is also included in the study, emphasizing the Event Swarm CEP framework's adaptability, usefulness, and unique characteristics. The capacity to perform near-real-time analytics for big data in motion is influenced by a number of issues, even if implementations of this technology are rather established. The findings in this research show that Event Swarm can provide these kinds of solutions.

VII. REFERENCES

[1] Reddy A R and Kumar S P 2016 Predictive Big Data Analytics in Healthcare Second International Conference on Computational Intelligence & Communication Technology.

[2] Yadav V, Verma M and Kaushik V D 2015 Big Data Analytics for Health Systems Green Computing and Internet of Things.

[3] Asri H, Mousannif H, Moatassime H A and Noel T 2015 Big Data in healthcare: Challenges and Opportunities International Conference on Cloud Technologies and Applications (CloudTech)

[4] Jangade R and Chauhan R 2016 Big Data with Integrated Cloud Computing for Healthcare Analytics International Conference on Computing for Sustainable Global Development

[5] Hussain S and Lee S 2015 Semantic transformation model for clinical documents in big data to support healthcare analytics The Tenth International Conference on Digital Information Management

[6] Vaishali G and Kalivani V 2016 Big Data Analysis for Heart Disease Detection System Using Map reduce Technique International Conference on Computing Technologies and Intelligent Data Engineering

[7] Ni J, Chen Y, Sha J and Zhang M 2015 Hadoop-based Distributed Computing Algorithms for Healthcare and Clinic Data Processing Eighth International Conference on Internet Computing for Science and Engineering

[8] Pramanik MI, Lau RYK, Demirkan H, Azad MAK (2017) Smart health: big data enabled health paradigm within smart cities. Expert Syst Appl 87:370–383

[9] Rathore MM, Paul A, Ahmad A, Anisetti M, Jeon G (2017) Hadoop-based intelligent care system (HICS). ACM Trans Internet Technol 18(1):1–24

[10] Peddi SVB, Kuhad P, Yassine A, Pouladzadeh P, Shirmohammadi S, Shirehjini AAN (2017) An intelligent cloudbased data processing broker for mobile e-health multimedia applications. Future Gener Comput Syst 66:71–86

[11] Ray PP (2017) Internet of things for smart agriculture: technologies, practices and future road map. J Ambient Intell Smart Environ 9:395–420

[12] Ray PP (2017) Understanding the role of internet of things towards providing smart e-Healthcare services. Bio Med Res 28(4):1604–1609

[13] Nambiar R, Bhardwaj R, Sethi A, Vargheese R (2013) A look at challenges and opportunities of Big Data analytics in healthcare. In: Proceedings of the 2013 IEEE International Conference on Big Data, Big Data 2013, USA, pp 17–22

[14] McGlothlin JP, Khan L (2013) Managing evolving code sets and integration of multiple data sources in health care analytics. In: Proceedings of the 2013 international workshop on data management analytics for healthcare, p 14

[15] Chandola V, Sukumar SR, Schryver J (2013) Knowledge discovery from massive healthcare claims data. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, USA, pp 1312–1320

[16] Gravina R, Alinia P, Ghasemzadeh H, Fortino G (2017) Multisensor fusion in body sensor networks: state-of-the-art and research challenges. Inf Fusion 35:1339–1351

[17] Wu P-Y, Cheng C-W, Kaddi CD, Venugopalan J, Hoffman R, Wang MD (2017) Omic and electronic health record big data analytics for precision medicine. IEEE Trans Biomed Eng 64(2):

[18] Ray PP (2014) Internet of Things based Physical Activity Monitoring (PAMIoT): an architectural framework to monitor human physical activity. In: Proceeding of IEEE CALCON, Kolkata, pp 32–34

[19] M. Al-Hubaishi, C. Çeken, and A. Al-Shaikhli, A novel energyaware routing mechanism for SDN-enabled WSAN, Int. J. Commun. Syst. 32 (2018), no. 17, e3724.

[20] F. Aktas, C. Ceken, and Y. E. Erdemli, IoT-based healthcare framework for biomedical applications, J. Med. Biol. Eng. 38 (2018), 966–979.

[21] N. Garg, Apache Kafka, 2014. 31. Apache Spark, Apache Spark™—Unified analytics engine for big data, spark.apache.org, 2018.

[22] A. M. Salem, K. Rekab, and J. A. Whittaker, Prediction of software failures through logistic regression, Inf. Softw. Technol. 46 (2004), no. 12, 781–789,

[23] L. Ganz, 54—Electrocardiography, in Goldman's Cecil Medicine, 24th ed., vol. 1, Saunders, Philadelphia, PA, USA, 2012, pp. 272–278,

[24] M. G. Kaya et al., Coronary ischemia induced Wolf Parkinson White syndrome, Int. J. Cardiol. 129 (2008), no. 1, 3–4.

[25] Mei-Sing Ong, Farah Magrabi and Enrico Coiera, "Syndromic surveillance for health information system failures: a feasibility study, " J Am Med Inform Assoc published online November 26, 2012

[26] Luckham, D., The Power of Events, An intoduction to Complex Event Processing in Distributed Enterprise Systems, Addison Wesley, 2002

[27] Jean Bacon, Ken Moody, John Bates, Richard Hayton, Chaoying Ma, Andrew McNeil, Oliver Seidel, Mark Spiteri, Generic Support for Distributed Applications" IEEE Computer, March 2000, pp 68-76

[28] C. Fidge. Logical time in distributed computing systems. IEEE Computer, pages 28–33, Aug. 1991.

[29] HL7 Version 3 Standard: Decision Support Service (DSS), Release 1, August 2011

[30] Gianpaolo Cugola, Alessandro MargaraProcessing. Flows of Information: From Data Stream to Complex Event Processing.