

## ASSESSING THE HUMAN CONDITION IN MEDICAL CYBER-PHYSICAL SYSTEM BASED ON MICROSERVICE ARCHITECTURE

Bohdan Havano<sup>1</sup>, Mykola Morozov<sup>2</sup><sup>1</sup> Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine.<sup>2</sup> Technical University of Munich, Boltzmannstr. 3, Garching b. Munich, 85748, Germany

Authors' e-mail: bohdan.i.havano@lpnu.ua; mcmikecreations@gmail.com

<https://doi.org/10.23939/acps2021.02.112>

Submitted on 10.10.2021

© Havano B., Morozov M., 2021

**Abstract:** The goal of the work is to propose architectural and information model for assessing the human condition on the basis of microservice architecture in medical cyber-physical system, which, in contrast to the known models for assessing the human condition, can simultaneously provide scaling, fault tolerance and increase the speed of human condition assessment. The theoretical substantiation and the new decision of an actual scientific problem of development and research means of an estimation of a human condition in medical cyber-physical system have been considered. These means involve the parallel processing of data on vital signs of the human condition, organizing the means of information processing into separate independent logical elements – microservices, in comparison with other existing medical cyber-physical systems. An architectural model based on microservice architecture has been proposed.

**Index Terms:** medical cyber-physical system, microservices, assessing the human condition, indicators, asynchronous data processing.

## I. INTRODUCTION

Today, the world is facing an increasing number of diseases and patients. In addition, wars, environmental pollution, foodborne illnesses, and human-animal relationships are leading to the emergence and spread of new diseases and viruses. Eventually, with the advent of the unknown disease, governments face two main challenges: firstly, medical centers need more and more qualified staff to periodically monitor each patient and act quickly in the event of an emergency. Secondly, the correct diagnosis of the patient's condition and the progress of his situation are critical for the medical staff. In addition, the integration of new technologies into health care has become an important task in order to make hospitals more efficient in terms of monitoring and analyzing patient data, thus improving medical processes.

The recent advent of sensor devices, especially Wireless Sensor Network and the Internet of Things, as well as big data technologies such as the Hadoop ecosystem, is leading to a new revolution in healthcare. Basically, the medical cyber-physical system consists of a set of biomedical sensors that allow constant monitoring of vital signs (e.g., heart rate, respiratory rate, blood oxygen level, temperature, blood pressure, etc.) of the patient and periodically transmit the collected data to the responsible

person for further verification [1]. Smart technologies have opened up a whole world of applications in the diagnosis and treatment of diseases such as cancer, glucose monitoring, depression, Parkinson's disease, connected contact lenses and more. In addition, it has become a major motivation for healthcare organizations to invest heavily in big data analysis.

To date, there is no detailed review of medical CPS software architectures, which makes it difficult to assess the maturity of current solutions and identify trends, gaps in research or future measurements. Both European and American initiatives – such as the European Commission's Horizon 2020 Program – emphasize the need to develop a research agenda for medical CPS, for example on ICT solutions for active and healthy aging and e-health innovation to empower patient.

As the number of vital signs to be monitored increases, these signs become more complex and a complex physical process (such as disease and treatment) needs to be controlled, and new solutions emerge. Some initial studies describe their solutions in terms of cyber-physical systems, the Internet of Things, and others in terms of the Internet of the future. So, [2] includes monitoring of heart rate, blood oxygenation level and respiration rate. The proposed solution in [3] is based on the Spot Vital Signs reading station, which allows to read vital signs from a limited set of compatible sensors. In [4], the authors apply programming abstractions to human monitoring programs to monitor posture and activity and count steps. The [5] supports the management and storage of large amounts of data streams coming from the body's sensory networks, which monitor the health of patients (e.g., heart attacks, diabetes and asthma). [6] suggests monitoring the patient's movements, as counting the number of steps and determining postures during the day. In [7], device manufacturers are not considered, because the approach is based on the use of sensors built into smartphones, and therefore, the drivers are already built-in. The chronic patient monitoring system described in [8] supports early diagnosis with a smart mobile device capable of processing data from Bluetooth sensors such as heart rate monitors, accelerometers and body temperature sensors. The EcoHealth platform [9] provides physicians with control over all information about their patients, as well as data attached to them by body sensors, medical records, and

reports of problems, symptoms, or abnormalities measured by sensors. [10] introduces a system called CardioNet, which includes a monitoring system that facilitates remote monitoring of patients with cardiovascular disease using certain portable devices. [11] offers telemonitoring of health-related parameters – speed and direction of movement, pulse and blood oxygen saturation. [12], unlike others, implements “Intelligent Construction” to process large amounts of data received from sensors connected to the body (e.g., wrist, ankle, heart and chest). PRIME [13] offers a two-tier architecture consisting of a communication layer that uses overlays using devices in a superimposed network built on top of low-level wireless technologies, and an API programming layer that provides abstractions and operations to implement P-RESTful (Pervasive) programs. The above medical systems are not flexible. Adding new sensors or supporting other vital signs is difficult or impossible. Some of them use the quantum K-means algorithm described in [14].

## II. PROBLEM STATEMENT

Medical CPS continue to cause difficulties in the scientific community due to:

Collection and storage of big data. Biosensors constantly record the vital signs of patients, usually every second, and then they send their records to the data center. Patients' vital signs have three characteristics: mass data collection; high-speed generation; and heterogeneous nature [1]. Unfortunately, relational database management systems and traditional storage technologies cannot handle this type and amount of data.

Rapid detection of emergencies. A normal range of health is determined for each vital indicator. Records outside this range lead to an abnormal situation that should be quickly identified and reported to medical personnel in order to take appropriate action. Therefore, rapid detection and response to emergencies is crucial because it can endanger the safety or even the life of the patient.

Diagnosis and classification of patients. In the health sector, several diseases can cause similar symptoms. Thus, the diagnosis of the disease is susceptible to human error, which can provoke disability or death. Therefore, the development of new methods and algorithms to study, on the one hand, the correlation between symptoms, and on the other hand, the classification of patients according to their situation can help make the right decision. One of the goals of this work is to provide acceleration on classification of patients.

To overcome the above problems, it is proposed to develop an effective and reliable medical cyber-physical system for assessing the human condition, which is based on the proposed model of processing human vital signs.

The software platform and middleware of the medical cyber-physical system must meet the architectural requirements:

- Heterogeneity and dynamic connection of the sensor. The medical cyber-physical system requires the ability to use a large number of body sensors using platforms that support high-level interfaces for transparent

access to sensors, hiding specifications and protocols for users and applications that use them.

- Scalability. The vast amount of data that is expected to be generated by medical CPS sensors requires scalable storage and processing capabilities. Real-time data streams from body sensor networks can trigger contextual and situational events. Managing and processing these events (so-called big data analytics) can be useful for decision making and can, in turn, trigger services in response to these events.

- Compliance with standards. Data transfer to cloud storage and processing services often requires improved data quality and involves converting data to standard to address interoperability, data cleanup, redundancy, and aggregation.

- Security and privacy. Personal data as well as medical data often become the object of attack by intruders, so the task is to ensure their protection.

## III. PURPOSE OF THE WORK

Also, it is worth highlighting the functional requirements for medical cyber-physical systems. Therefore, the medical CPS must perform such functions within specified conditions and properties which are better than existing medical CPS have:

- Event notification function. Existing medical CPS usually have only 1 notification channel or don't have it at all. So, the purpose of the work is to develop at least 2 different notification channels. Also, event notification should be performed less than 5 seconds (before new measurement starts).

- Identify critical values of vital signs of the human condition. Some existing medical CPS do not automatically identify critical values of vital signs and just gather information to show it to doctor or any other responsible person. Purpose of the work is to develop automated identification of such critical values. It also should be performed less than 5 seconds (before new measurement starts).

- Analyze users and their vital signs, generate recommendations and identify similarities. Existing medical CPS put the responsibility of making diagnosis on doctors, so it is a manual process. The purpose of the work in this direction is to automate diagnosis and classification of patients and to increase speed of classification algorithm.

- Process and aggregate data for storage and subsequent visualization in a standardized format for semantic compatibility of medical data in case of need to send them to external services. Existing medical CPS are using their own data format which is not usually compatible with known health data format. So, proposed system should be flexible and compatible with different aggregated data structures. It also should be performed less than 30 seconds (this is default client request timeout).

- Process, store and aggregate historical (archival) medical data of the user. The purpose of the work is also to find solution how to store archival data separately from latest data to make the aggregation process in parallel and avoid overcrowding of database.

#### IV. INFORMATION PROCESSING MEANS OF MEDICAL CYBER-PHYSICAL SYSTEM FOR THE HUMAN CONDITION ASSESSING

The result of performing its functions by means of information processing in medical cyber-physical systems is an assessment of the human condition. Such means should process raw vital signs data obtained from measuring devices in order to provide these data with value in the process of monitoring and assessing the human condition.

Information processing means must solve the previously established problems for medical cyber-physical systems and solve the set tasks. Each mean must perform its work independently of other means, which allows to parallel the process of information processing. This is an important aspect in medical CPS, as some drugs should work as quickly as possible, which may directly affect the state of human health. In the case when a certain mean depends on the result of another, you need to think about the priorities of these means, thereby determining the order of data processing by these means.

One of the most important and priority means of information processing in the medical CPS is a means of identifying critical indicators. When receiving data on human vital signs, you first need to determine how critical these indicators are. In order for this mean to be able to perform indicators of human condition indicators, it is necessary to use a certain method of assessing human vital signs, which will be used to decide on the detection of critical events. It should be noted that the means of detecting critical indicators should perform its functions as soon as possible. An equally important function of this mean is to respond immediately to this event, as it can increase a person's chances of timely regulation of these indicators without harm to health. Thus, the means of identifying critical human vital signs is one of the keys in terms of assessing the human condition in medical CPS. As responding to events is a key mean in identifying critical indicators of a person's condition, there is a need for a means of reporting events.

The event notification means should perform the function of informing the responsible person or group of people monitoring the human condition. Events can be of different origins and levels of severity. For example, there may be an event triggered at the request of the patient himself, the purpose of which is to notify the responsible person of the event. Events can also be created by other means of information processing in the medical CPS, such as a means of identifying critical indicators, etc. That is, these are events that are created as a result of certain data processing automatically. Keep in mind that these events can be critical or informative, so the event notification means should take into account the severity of these events and use appropriate notification methods. Therefore, the event notification means can have several notification methods, as well as different endpoints of these notifications. Namely, certain event notifications must be immediate and addressed to the responsible person by means of developed methods of immediate notifications, and the purpose of such methods is to achieve the notification of the endpoint in the most

efficient and convenient way. If the event is not critical, the notification will be sent in another form. Thus, responding to events of varying severity should also be prioritized. The end point of the event notification does not have to be the responsible person or group of people, it can also be another means of processing human data, which should be performed only in certain events. In general, the event notification means is partly related to other information processing means in the medical CPS, as it must respond to events that may occur in these means. Therefore, the main task of the means of notification of events is to inform other means or responsible people about some event. Also, in addition to information about the conditions of the event, these notifications may contain an additional message for the endpoint according to certain methods of event notifications used for different types of events.

An equally important means of information processing in the medical CPS is a means of classifying patients and diagnosing diseases. Compared to the means of detecting critical indicators, it is not so high priority, and has no restrictions on the duration of its functions. The main tasks of this means are the classification of people according to certain similar parameters defined in the methods of classification of this means, as well as the diagnosis of diseases based on methods of diagnosing symptoms and diseases. This means should simplify the task of the doctor as a responsible person in a correct diagnosis. Also, this means can perform the function of forecasting indicators of the human condition, as well as offer advice based on decision-making methods based on this forecasting. Therefore, this means is also one of the keys in terms of assessing the human condition.

An integral part of medical CPS is the visualization of these vital signs of the human condition. Proper and convenient visual presentation of this data can significantly improve and speed up the work of the responsible person who monitors and assesses the human condition. To ensure a quick response to the client's request from the responsible person, as well as faster visual display of this data on the client, it is proposed to develop a data processing means for storage and subsequent visualization. This mean has 2 main tasks: processing raw data in terms of their structuring; providing a convenient structured response to customer requests. As data on the state of human vital signs are received by the medical CPS in the form of raw data, there is a need to process these data and structure them. Saving the data in the correct unified format will ensure compatibility with different types of customers, as well as a quick response to customer requests.

As the history of indicators of a human condition can contain the useful information for the responsible person, there is a task in archiving of the vital indicators data. This task is performed by means of storing and accessing archival data. Archiving patient data is a key operation in the human treatment process. On the one hand, it allows the responsible person to notice the progress of the patient's condition over time, and on the other hand, it helps professionals to better understand the disease and improve the quality of care. It should be borne in mind that archival data can be considered as big data, so these means should be designed to work with such data in the processing,

aggregation and storage of archival data. It is worth noting that archival data are not very relevant data, and the question of a quick response to a request for archival data is not relevant.

All these means of information processing in the medical CPS are certainly important for assessing the human condition. However, it should be noted that they may have different priorities in terms of the urgency of their functions. The question arises of dividing these means into separate independent logical elements. To address this issue, you need to build the right model of information processing in medical CPS, which can integrate the means described above. All means must work in parallel to respond in a timely manner to changes in the human condition, which is another requirement for the model of information processing medical CPS.

## V. MODEL OF INFORMATION PROCESSING IN MEDICAL CYBER-PHYSICAL SYSTEM FOR THE HUMAN CONDITION ASSESSING

The model of information processing in medical cyber-physical system must meet the requirements for the integration and implementation of means of processing vital signs of the human condition.

Since the above means perform simultaneous and independent tasks, it is advisable to perform them in parallel. To differentiate the means of information processing into separate independent logical elements, it is advisable to use a model of information processing based on microservice architecture. It is proposed to apply a microservice architecture, where each means will be in the form of a microservice.

Microservices are programs that are organized or structured as a set of freely related services. Microservices are independently scalable. As the number of users and data in the system grow, it is easier to scale with the help of microservices. You can increase resources for essential microservices instead of scaling the entire system. It also means that scaling is faster and often more economical. New microservices can be added to the medical CPS without requiring system downtime. Microservices reduce system downtime by isolating faults. If a particular microservice fails, you can isolate this failure in this single service and prevent cascading failures that could cause the system to fail. This fault isolation means that your critical system can stay in working order even if one of its modules fails.

The proposed model is as follows (see Fig. 1) and consists of the following key elements: orchestration service; operational database; database to organize the queue of processes; N-number of microservices with own database, if necessary.

The data source for the proposed model is an operational database. As it has been mentioned earlier, the model accepts raw, not aggregated and unprocessed data. This means that there is a lot of this data, and storing it in a single storage location leads to a number of problems. To avoid these problems, an operational database is proposed, the essence of which is the temporary storage of data. The data is stored in it exactly until all the time of information processing – microservices – perform their task, the overall

process of processing raw data is over. It is no longer relevant to store data in this form, so they are deleted. This allows not to overcrowd the operational database, which preserves the speed of reading and writing data. It also solves the question of waiting for a response to the request to record data, because the mechanism of data transfer from biosensors to the system should not wait for information processing, but receives an immediate response to the acceptance of information for asynchronous processing.

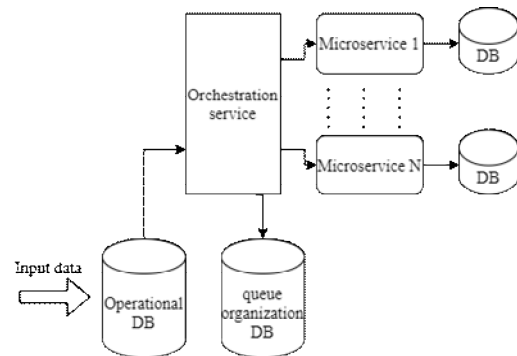


Fig. 1 Information processing model

The control of their tasks by means of information processing takes place in one of the most important elements of the proposed model – the orchestration service. It is the service that determines when data should be deleted from the operational database. The orchestration service is a microservice that forms a queue of tasks of various means of information processing, sets the priorities of these means, monitors their implementation. Also, if necessary, it allows microservices to communicate with each other. Another task of this service is the aggregation of data to respond to the customer's request, i.e., for different requests, it communicates with different microservices that are responsible for certain means of information processing.

To manage and form a queue of tasks of various means of information processing, it is proposed to use a special database. The orchestration service creates priority queues that are stored in this database. As soon as the microservice performs its task, the record is deleted from the queue. That is, this database is a list of tasks that are planned to be performed by information processing microservices. It is on the basis of this list and the decision is made to delete raw data from the operational database, provided that all services have completed their task for this set of vital signs of the human condition. With the help of this database, different processes of information processing by microservices can be performed asynchronously and in parallel.

In the proposed model, each of the above proposed means of data processing in medical CPS is a microservice that solves the problem clearly set before it, which is the basic concept of microservice architecture. Depending on the number of means that are proposed to be used in medical CPS, the number of microservices may be different. This allows to build a flexible architecture of the medical cyber-physical system, in which you can add or remove certain information processing means even after the design stage of the system.

## VI. THE PRINCIPLE OF WORK OF THE ARCHITECTURAL-INFORMATION MODEL

Human vital signs come to the API gateway. After successful authentication and authorization of this request, the data are recorded in the operational database through the data recording service, and the orchestration service is informed about the receipt of new raw data of vital signs of a person. Thus, the data source receives the answer that the data is accepted for processing. After that, the orchestration forms a queue of processes that must perform information processing services. This queue is created in a special database for the process queue.

According to the above queue, microservices perform their function of processing raw data to assess the human condition: identification of critical indicators; classification of patients and diagnosis of diseases; data processing for storage; preservation of archival data.

When all these processes are performed, the raw data will be deleted from the operational database.

The process of obtaining data for visualization is as follows. The client executes a request to a specific API gateway endpoint where the request and the user go through the authentication and authorization process. Provided that the user is authenticated and has the right to receive the requested information, the API gateway accesses the required microservice, which is responsible for the requested data, through the orchestration service. The data is received in response to a customer request, where further visualization takes place.

## VII. FUNCTIONAL ELEMENTS OF THE SYSTEM

Data recording service and operational database. This service records the received raw data without prior processing in the operational database. The purpose of creating this service is to reduce the waiting time of the client who sent the data of human vital signs. The operational database, in turn, serves as a centralized storage of raw data, which is used by microservices to process information asynchronously to the request to record this data. The data in this database is stored temporarily, which avoids problems with database overload, and preserves its performance.

The organization of storage of raw data in the operational database is implemented using MongoDB. This is a document-oriented NoSQL database that does not require a description of the table schema. This database supports the storage of documents in JSON-like format, which meets the needs of fast recording of raw unstructured data.

```
{
  identifier: {
    user_id: user_id,
    indicators_raw_data: raw_data
  }
}
```

Fig. 2 Data structure in operation database

Recording raw data in the operational database is shown in Fig. 2. Moreover, raw\_data can have any structure. Implementation of an operational database as MongoDB solves the problem of heterogeneity of measuring devices, the structure of raw data which is heterogeneous. This means that regardless of the type of measuring device, the logic of storing raw data in the operational database is the same.

The organization of storage of background tasks that must be performed to process raw data is implemented using Redis DB. Redis is an open storage structure in memory that is used as a database. Redis is a resident NoSQL class database that works with key-value data structures.

Authentication and access control service. In this service the processes of authentication and authorization of requests and users take place. With the implementation of authentication and authorization in microservices, the process becomes much more complex than in traditional monolithic architecture. Authentication and authorization of programs in microservice architecture is usually implemented in a centralized service, which is responsible for this. Authentication checks must stop inside the API gateway. Authorization can be implemented either in the API gateway or in microservices. To be able to perform large-scale authorization checks for specific programs, authorization must be processed in certain microservices.

One way to solve the system authorization problem is to implement it in the API gateway. This makes it possible to restrict requests to certain endpoints. The disadvantage of implementing authorization in the API gateway is that it can only be role-based access to endpoints. Implementing additional access checks on specific domain objects requires the creation of specific domain logic within the API gateway, and therefore will lead to domain logic leakage. In addition, with the implementation of multiple backends for API interfaces / gateways, authorization management becomes increasingly difficult and complex.

The best solution would be to put microservices in charge of authorization processing. Thus, a certain program needs to be implemented only in one place. The disadvantage of this is that authorization will be more common in different services. If you have a lot of roles that change very often, it becomes harder to manage.

This means that when human vital data is received from the sensors, the API communicates with the authentication and access control service, where the request is authenticated, and this data will belong to a specific user. It will also be checked whether the authenticated user has the right to write this data to the system. In the case of a request to read the data, the authenticated user will also be checked for access to the requested data. Therefore, this service ensures compliance with the confidentiality and privacy of personal and technical data of users, as well as protection of data from unauthorized access.

A service for collecting personal data and building connections between users and user databases. The main function of this microservice is the processing and storage of personal data of users. He is primarily responsible for registering new users in the system, creating an entry in the user database, including the required fields of personal data

required for further work in the system and interaction with other users.

An additional function is to interact with external resources to obtain additional personal and technical data. Since, in addition to data on vital signs of the human condition, there are other data that could be useful for the responsible person in the process of assessing the human condition in the medical cyber-physical system, obtaining such data from external resources can facilitate the process of assessing the human condition and provide additional information.

An equally important function of this service is to build relationships between users, which is the basis for further control of access to other user data according to the business logic of the system.

Therefore, this service is a centralized repository of personal data of users and the links between them, to further control access to write / read data according to business logic.

Orchestration service and databases for process queues. To organize data processing, it is advisable to create a separate microservice that could control the process and priority of performing the functions of other microservices that are responsible for information processing.

Upon receipt of raw data, the orchestration service informs other services about the need to process this data, according to the priority queue created in a special database. It also monitors the execution of these processes, and is responsible for deleting raw data after it has been fully processed.

Also, this microservice is a certain type of orchestrator, which allows other microservices to communicate with each other. For example, if a certain event was detected and created during the process of certain data processing, then through it the information about this event will be transferred to the service responsible for the events.

If we talk about the request to read data, then this service is aggregating data to respond to the request. This means that, depending on the request, the orchestration communicates with the necessary services responsible for a particular type of requested data and aggregates them into a single data set.

It should be noted that this microservice is the center of the architecture of the medical cyber-physical system. This is a problem because the failure of this microservice endangers the operation of the entire system. To solve this problem, the system also has the ability of microservices to communicate with each other, in case of unavailability of the orchestration service. The same approach is with the aggregation of the requested data. If this microservice fails, this task is undertaken by the API gateway, which also has the ability to communicate with each microservice directly.

This microservice is implemented using the Ruby on Rails (ROR) framework. It has a built-in framework for declaring deferred tasks (Active Job) and their execution in different queues. Sidekiq is used as a queue adapter for the implemented system. Sidekiq uses threads to work simultaneously with many tasks in one process. It provides functions for asynchronous execution of any tasks or for planning their further execution by placing them in the

queue of background tasks. It also offers clear retry strategies in case of failure. Therefore, it perfectly corresponds to the principle of realization of processes of information processing by various microservices. The Redis database is used to store all operational data.

Service for storing and notifying about events and event databases. As it has been previously reported, events should be divided into automatic and on-demand events. And also, on critical and informational ones. If the reaction to critical events should be immediate, then the situation with information events is somewhat different. Response to information events is not immediate, so it should be possible to learn about this event with a certain period of time after its creation. The question arises in saving events and data about them for further receipt and visualization on the client. That is why the service of storing and notifying about events has its own database, where all events and their accompanying information are stored. Information events are notified on the client side via a graphical interface.

If it is enough to store information events in the database, and wait until the responsible person makes a request to receive this type of event, the principle of notification of critical events is completely different. As it has been mentioned above, the response to critical events should be immediate, as it can have a direct impact on a person's condition and health. To ensure an immediate response to this type of event, the question arises in the creation of a means of notification, which is also a function of this microservice. This means that with the help of the developed mechanisms for sending notifications, when creating this type of event, the service is responsible for immediately informing the responsible person or group of people.

If we talk about events created on demand, they can also have different levels of severity, according to the business logic of the developed medical cyber-physical system. However, creating this type of event works on the same principle as an automatic one. Records of their creation and related information are made from the event database, and critical events are notified immediately.

When implementing the system of notifications of critical events, the following goals must be met: the ability to inform the responsible person; opportunity to inform a group of people; the ability to inform using different communication channels.

Therefore, the function of this service is to store and inform about events related to the functional state of the human body. To send e-mails, the SendGrid service was used. The implementation of the use of this service involves the generation of an API key and its use in subsequent system requests to the service to send an email to the e-mail address of the responsible person or group of people. To implement sending Push notifications to the mobile application of the responsible person or group of people, the Rpush library, designed for services based on Ruby OnRails, is used. It supports such push notification platforms as Google, Apple, Microsoft and Amazon. Critical event notification testing was performed. According to the testing, after detecting a critical event for its creation, notification of it takes an average of 2.225ms (2.2 seconds).

The critical indicator detection service is one of the most important services in terms of assessing the human condition in the medical cyber-physical system. Identifying critical indicators and responding to this event in a timely manner is very responsible, as it can directly affect human health and even life.

The process of performing the function of this service has the highest priority, so the detection of critical vital signs of the human condition occurs in the first place. According to the above-described method and elements of the business logic of the system, the critical detection service processes the information received from the sensors. In case of detection of critical indicators, a critical event is created, about which the responsible person is immediately informed through the storage service and event notifications.

When critical indicators are detected, the latest data, the values of the indicators of the last measurement, are taken from the raw data. That is, regardless of the size of the input data, the time required to identify relevant critical indicators is the same. However, critical metrics are detected for all raw data sets, starting with the most recent, but a critical event is created only for the last measurement, as it is the latest data that requires immediate detection of the criticality level. Critical events of older measurements are created in the database as information events. Accordingly, they are not notified. Testing of critical indicators was carried out. According to the conducted testing, on average 1236.69 ms (1.24 seconds), are required to detect critical indicators of the last measurement and send a request to create a critical event.

Patient classification and disease diagnosis and database service for clustered users. Another important step in assessing the human condition in the medical cyber-physical system is the process of analyzing the data, to understand the disease and try to minimize their future consequences. The first step in data analysis is the classification of patients into clusters, where patients in one cluster have common characteristics (symptoms and situations). The second step is to study the correlation between the vital signs of patients in one cluster. This can help the responsible person to understand the causes of the disease and the behavior, and thus avoid misdiagnosis of the disease and find appropriate treatment. The purpose of this microservice is to implement two proposed data analysis algorithms, one – for patient classification and the other – for disease diagnosis.

You need to understand that the data being analyzed is big data. Big data is a term that refers to data that is beyond the storage, and processing capabilities of a classic computer and getting some information from a large amount of data refer to a very big problem. That is why quantum computing comes to the rescue, offering many promises in information processing systems, especially in big data analytics.

Quantum computers are known to perform certain complex algorithms exponentially faster than classical computers. The quantum version of the K-means algorithm provides exponential acceleration for very high dimensional input vectors. One version of the quantum K-means

algorithm is described in [14]. Classification of patients and diagnosis of diseases is based on a similar principle, using an adapted algorithm K-means.

To implement an adapted quantum K-means algorithm, you need to use three different quantum routines: SwapTest, DistCalc, and Optimization Grovers.

The SwapTest subroutine measures the overlap of two quantum states  $\langle \alpha | \beta \rangle$  based on the probability of measuring the control qubit in the state  $|0\rangle$ . Overlap is a measure of similarity between two states. If the probability of the control qubit in the state  $|0\rangle$  is equal to 0.5, it means that the states  $|\alpha\rangle$  and  $|\beta\rangle$  are orthogonal, whereas if the probability is equal to 1, then the states are identical. The states  $|\alpha\rangle$  and  $|\beta\rangle$  may be unknown before the procedure is started, and only measurements on the control qubit are required. The probability of the controlled qubit being in the state  $|0\rangle$  is given in (1).

$$P(|0\rangle) = \frac{1}{2} + \frac{1}{2} |\langle \alpha | \beta \rangle|^2 \quad (1)$$

The SwapTest scheme is shown in Fig. 3.

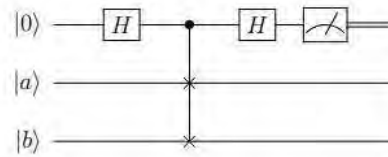


Fig. 3 SwapTest scheme.

The states  $|\alpha\rangle$  and  $|\beta\rangle$  consist of  $n$  qubits each. They are either prepared using amplitude encoding or can be downloaded directly from Quantum Random Access Memory (QRAM). Then the overlap is calculated by measuring on the control qubit.

The overlap from the SwapTest routine is used to calculate the distance in the DistCalc routine. The Euclidean distance  $|\alpha - \beta|^2$  between the two vectors  $|\alpha\rangle$  and  $|\beta\rangle$  is calculated by performing the following three steps.

First step is preparation of states: Prepare two quantum states as in (2),

$$\begin{aligned} |\vartheta\rangle &= \frac{1}{\sqrt{2}} (|0, \alpha\rangle + |1, \beta\rangle) \\ |\varphi\rangle &= \frac{1}{\sqrt{Z}} (|\alpha \parallel 0\rangle + |\beta \parallel 1\rangle) \end{aligned} \quad (2)$$

where  $Z = |\alpha|^2 + |\beta|^2$ .

Second step is finding the overlap: calculate the overlap  $\langle \vartheta | \varphi \rangle$  using SwapTest.

Third step is calculation of distance: Using (3) to calculate the Euclidean distance.

$$D = 2Z \left| \langle g | \phi \rangle \right|^2 \quad (3)$$

Using this method, the K-distances are calculated to each centroid of the cluster. To find the nearest cluster centroid, the Grovers Optimization subroutine is called, which is based on the Grover algorithm. The input point is then assigned to the cluster of this nearest centroid. This is done for all input points. After assigning clusters to each data point, the centroids are counted by calculating the average of all data points in the respective clusters. The whole process is repeated until the purpose of the clusters stops changing.

Due to the efficient method of calculating distances using quantum parallelism, the adapted quantum algorithm K-means achieves exponential acceleration. Lloyd's K-means algorithm has time complexity  $O(NMK)$ , while the quantum version has time complexity  $O(\log(N)MK)$ . This can be explained by the fact that N-dimensional classical information is encoded using  $\log(N)$  qubits.

Data processing service for storage and storage of processed current data. Proper and convenient presentation of data to the client is the key to rapid and effective assessment of the human condition by the responsible person. Processes of additional processing of the received data on the client will cause superfluous expenses of time for visualization of vital signs and a condition of a human body as a whole. About the processing of raw data on the server at the time of the client request, it will also cause a delay. Therefore, it was decided to process the raw data immediately upon receipt, without waiting for the client's request to read this data, which will reduce in the future the time of receipt and visualization of the requested data.

This process takes place in parallel with other raw data processing processes. In a data processing service for storage, raw data is processed and formed into a certain structure (defined by the business logic of the system), which will be compatible with different types of customers. Structured and processed data are stored in a database of processed current data. This database stores only current data (for a short period of time, determined by the business logic of the system), which are ready to be sent to the client for further visualization, or in other medical information systems. According to the business logic of the system, in a certain period of time the data becomes irrelevant, which is the need to remove them from the database of processed current data.

To store only current data, the process of deleting irrelevant data is implemented. The UNIX Cron task scheduling service is used. The list and time of execution of the planned tasks is created in the special file crontab. This list includes Ruby's built-in mechanism for performing external tasks, called Rake Task.

The input data consists of a set of values measured for each indicator for the entire period of time, starting from 00:00 of the current day. When processing data, the measurement time of the last stored indicators in the system is firstly checked, so the data of indicators that have already been processed in the stored in the system are not further processed. Thus, if the data to be processed are received regularly, only the data of the last measurements are processed.

However, under certain conditions (for example, in the absence of an Internet connection), the data of each measurement was not sent and accumulated on the client. In this case, the data is processed not just by one measurement, but by many, depending on the duration of the connection. The maximum amount of data that can be sent to the server for processing may include the measurement of indicators for the whole day (the request for data processing came for the first time at 23:59). In this case, the input data contains information about the indicators for 1439 measurements, which consist of 1439 data sets with the value of each indicator. According to the tests performed, the average processing time of input data was determined depending on their volume. Namely, when sending the data of each measurement regularly, and when sending the maximum amount of data – a set of measurements for the whole day (see Table 1).

Table 1

Data processing execution dependence

The amount of input data	Data processing execution time
Data set 1 measurement	149ms (0.15 seconds)
Data set of 1439 measurements	4200ms (4.2 seconds)

Data aggregation service for further visualization. To parallelize the processes of recording and reading relevant data on vital signs of the human condition, it was decided to divide these processes into separate microservices. When a client requests data on vital signs of the human body, the data aggregation service for further visualization accesses the database of processed current data and retrieves the necessary data.

Table 2

Data aggregation execution dependence

Query parameters	Average execution time of data aggregation
Obtaining running values of all indicators	19ms (0.02 seconds)
Getting the average values of all indicators for the day	15ms (0.015 seconds)
Getting the values of a specific indicator for the day	89ms (0.9 seconds)
Obtaining the values of a specific indicator for the week	284ms (0.29 seconds)
Obtaining the values of a specific indicator for the month	678ms (0.68 seconds)

The requested data is aggregated according to the customer's request and the business logic of the system. For example, the service may return running actual data or aggregated data for a certain period of actual time (determined by the business logic of the system). The service can also aggregate data of all vital indicators of a person's condition, several, or only one of them, depending on the client's request.

The data aggregated by the service is ready for visualization on the client and does not require additional processing. Also, testing of data aggregation execution time



was performed, depending on the data request parameters. Testing was performed according to the requests generated by the client program. The average execution time of data aggregation is shown in Table 2.

Service of storage and access of archival data and archival storage. As follows from the data processing service for storage, it is proposed to divide the data into current and archival. The difference between these data is their relevance in the process of monitoring and evaluating human condition, as well as the frequency of requests for these data.

Archival data are processed and stored in parallel and on different principles than current data. Therefore, it is obvious to create a separate microservice for such needs.

The service of storage and access of archival data performs the functions of processing and storage of processed information in the archival storage. Also, it is responsible for aggregating this data according to the client's request to receive this data. Data aggregation also occurs according to the business logic of the system.

### VIII. CONCLUSION

To solve the main problems of modern medical CPS, namely: collection and storage of big data; rapid detection of emergencies; diagnosis and classification of patients, it was necessary to develop new architectural solutions. An architectural model based on microservice architecture was proposed. This model solved the established problems via designing each mean in separate independent item – microservice.

The main functional features of proposed medical CPS were: event notification function – it took 2.22 seconds to perform this function, which was less than 5 seconds, also it had 2 different notification channels (email and push-notification to client device); identification of critical values of vital signs of the human condition was automated and it took 1.24 seconds to perform this function, which was less than 5 seconds; analysis of users and their vital signs, to generate recommendations and identify similarities – the quantum version of the K-means algorithm was used for users clustering and it provided exponential acceleration; processing (it took from 0.15 seconds to 4.2 seconds to perform this function, which was less than 30 seconds) and aggregation (it took from 0.015 seconds to 0.68 seconds to perform this function, which was less than 30 seconds) of data for storage and subsequent visualization in a standardized format for semantic compatibility of medical data in case of need to send them to external services using API gateway to aggregate data for different requests; processing, storage and aggregation of historical (archival)

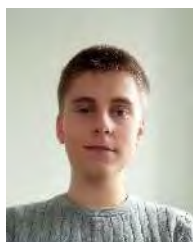
medical data of the user were separated from processing latest data and used as a separate service with its own storage.

### REFERENCES

- [1] Melnyk, A. *et al.* (2020) “Investigation of wireless pulse oximeters for smartphone-based remote monitoring of Lung Health”, *Advances in Cyber-Physical Systems*, 5(2), pp. 70–76. doi: 10.23939/acps2020.02.070.
- [2] Melnyk, A. *et al.* (2021) “HealthSupervisor: Mobile Application for Round-the-Clock Remote Monitoring of the Human Functional State”, *CEUR-WS*, 2853, pp. 24–37. Available at: <http://ceur-ws.org/Vol-2853/keynote3.pdf> (Accessed: 9 October 2021).
- [3] Mendez, E. O. and Ren, S. (2012) “Design of cyber-physical interface for Automated Vital Signs Reading in electronic medical records systems”, *2012 IEEE International Conference on Electro/Information Technology*, pp. 1–10. doi: 10.1109/eit.2012.6220696.
- [4] Gravina, R. *et al.* (2017) “Cloud-based activity-AASERVICE cyber-physical framework for human activity monitoring in mobility”, *Future Generation Computer Systems*, 75, pp. 158–171. doi: 10.1016/j.future.2016.09.006.
- [5] Zhang, Y. *et al.* (2017) “Health-CPS: Healthcare Cyber-Physical System assisted by Cloud and Big Data”, *IEEE Systems Journal*, 11(1), pp. 88–95. doi: 10.1109/jsyst.2015.2460747.
- [6] Bazzani, M. *et al.* (2012) “Enabling the IOT paradigm in E-health solutions through the virtus middleware”, *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1954–1959. doi: 10.1109/trustcom.2012.144.
- [7] Corredor, I. *et al.* (2014) “A lightweight web of things open platform to facilitate context data management and personalized healthcare services creation”, *International Journal of Environmental Research and Public Health*, 11(5), pp. 4676–4713. doi: 10.3390/ijerph110504676.
- [8] Páez, D. G. *et al.* (2014) “Big Data and IOT for chronic patients monitoring”, *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, 8867, pp. 416–423. doi: 10.1007/978-3-319-13102-3\_68.
- [9] Maia, P. *et al.* (2014) “A web platform for interconnecting body sensors and improving health care”, *Procedia Computer Science*, 40, pp. 135–142. doi: 10.1016/j.procs.2014.10.041.
- [10] Sebestyen, G. *et al.* (2014) “EHealth solutions in the context of internet of things”, *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1–6. doi: 10.1109/aqtr.2014.6857876.
- [11] Suci, G. *et al.* (2015) “Big Data, internet of things and cloud convergence – an architecture for secure E-Health Applications”, *Journal of Medical Systems*, 39(11). doi: 10.1007/s10916-015-0327-y.
- [12] Rathore, M. M. *et al.* (2016) “Real-time medical emergency response system: Exploiting IOT and Big Data for Public Health”, *Journal of Medical Systems*, 40(12). doi: 10.1007/s10916-016-0647-6.
- [13] Caporuscio, M. and Ghezzi, C. (2015) “Engineering future internet applications: The prime approach”, *Journal of Systems and Software*, 106, pp. 9–27. doi: 10.1016/j.jss.2015.03.102.
- [14] Kopczyk, D. (2018) *Quantum Machine Learning for Data scientists*, *arXiv.org*. Available at: <https://arxiv.org/abs/1804.10068> (Accessed: 9 October 2021).



**Bohdan Havano** received the B.S. degree in computer engineering at Lviv Polytechnic National University in 2015 and M.S degree in system programming at Lviv Polytechnic National University in 2016. He has been doing scientific and research work since 2017. His research interests include architecture and data protection in cyber-physical systems.



**Mykola Morozov** received the B.S. degree in software department at Lviv Polytechnic National University in 2021. Currently, he is a Postgraduate student of Informatics Department at Technical University of Munich. His research interests include pathfinding algorithms.