UDC 004.72
DOI: 10.20535/SRIT.2308-8893.2019.1.09

EDGE COMPUTING IN MULTI-SCOPE SERVICE-ORIENTED MOBILE HEALTHCARE SYSTEMS

IHOR PYSMENNYI, ROMAN KYSLYI, ANATOLY PETRENKO

Abstract. IoT based e-Health solutions is an upcoming trend which will revolutionize the healthcare in the near future. IoT has evolved from micro-electro-mechanical systems (MEMS), wireless technologies and Internet which together offer connectivity of systems, microelectronic devices, and medical services and allow data processing at the edge. That at the same time allows to save computational resources and avoid unnecessary point of failure, such as centralized synchronization point. Monitoring of patients' vital signs parameters (measured at home) is achieved by using modern Internet of Things technology which provides networkable connections between portable diagnostic sensors, their cell phones, cloud data storage with patients' Personal Health Records and professional health providers. This paper explores possibilities of using fog computing approach to shift data processing and computations from cloud to the edge and to build a multi-scope infrastructure for mHealth and citizen-observation system, based on SOA approach.

Keywords: wireless sensor networks; personal health systems; cloud services; IoT; fog computing; service-oriented architecture (SOA); mHealth; communication; infrastructure.

INTRODUCTION

Previous developments of medical mobile applications usually were monolithic ones, designed for a fixed HW infrastructure. The whole application must be developed and deployed in one piece and the entire tier must be retested and redeployed when something is changed. Hundreds of applications for smartphones with various operating systems have been created by different providers. We are mostly the first who start to investigate advantages of service-oriented architecture in mobile medicine. This paper is a generalization of the author's publications on the possible usage of the service-oriented computing paradigm (SOC) for building a medical services platform which allows unifying the development of applications for patients, doctors and the central server by orchestrating and composing web services from a common cloud repository. Due to this approach the created applications can be adapted to the particular patient, his disease and the plan of his treatment at home.

Many specialists in the world believe *that patient empowering* can transform medical care. Wireless internet connectivity, cloud computing, mobile devices

(smartphones and tablets), mobile applications and sensors modernized clinical trials, internet connectivity, advanced diagnostics, targeted therapies, and other science enable the individualization of medicine and force overdue radical change in how medicine is delivered, as well as regulated and reimbursed. Let's imagine that every medical sensor (or another data resource) of that ecosystem has its own URI allowing doctors and patients interact with it via the web browser, and at the same time every sensor can have the software interface – a set of web services allowing intelligent software agents to interact with it (analyze the data etc.) on behalf of doctors and patients. Certainly, the integration of that with the classical medical record is vital. Mobile devices are being used to capture data at the point of care and to keep the lines of communication open no matter where the doctor is, and they're being used at home to record and send vital health data back to the health professional and, in turn, to send important healthcare management information back to the patient. The personal data is used to track the ups and downs of patient's conditions as they go about their lives. This approach implies to the ***fog and mist computing paradigms***. Fog and mist layers are abstract conceptual levels at the network edge [1]. While fog connects these edge devices, putting a lot of storage, configuration, computing and analysis tasks away from cloud to the edge, mist resides directly within network fabric with ability for end-user devices to share their available computational and communication capabilities for performing a variety of applications and networking task [2, 3].

Advantages of computations at the edge are listed below:

- Preserving user data privacy by not sending sensitive data to the cloud and processing it in the local environment.
- Maintenance costs by significantly decreasing required bandwidth and computing power on the server.
- Reducing latency.
- Heterogeneous by-design architecture.
- Self-awareness.
- Fault-tolerance by-design.

Possible Computing levels of m-Health system in hand are shown on fig. 1 and includes the following levels:

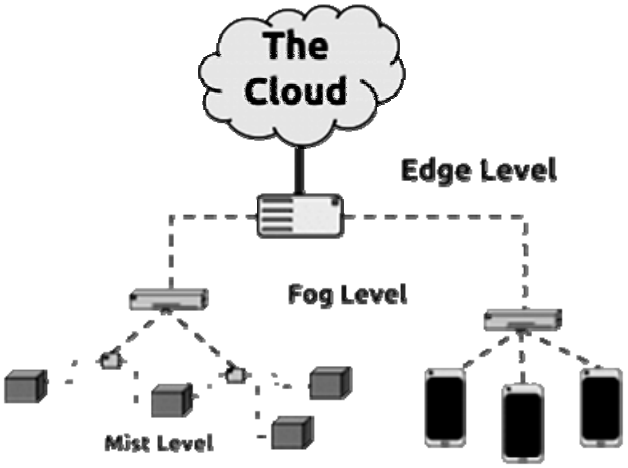

Fig. 1. Computing levels of IoT based system 3 [4]

- Sensor network and mist computing layer which consists of air quality sensors and BSN. Filtering and basic data validation is also performed on this layer.
- Fog-computing layer which is used for gathering and combining data from different sources and performing some analysis on it (described in section IV)
- Fog-gateway layer which provides communication between cloud server and other fog-networks.
- Cloud analytics layer for storing and processing of aggregated data, performing complex analysis on anonymized data for municipalities.
- PHR integration layer which allows to give user recommendations and predictions based on external eHealth information.
- Expert analytics layer which assists expert in making decisions and provides mechanism for feedback.

These levels imply to the security zone approach are proposed in [5]. There are 4 main challenges in development of such architecture:

- Preserving user privacy [6] as collecting and transferring gathered data can reveal both user's identity and sensitive health data with geolocation.
- Efficiency: both in energy consumption and computation power 1.
- Fault tolerance.
- Authentication of system nodes and security against piggybacking attacks, detection of untruthful data.

The rest of the paper is organized as follows - in section 2 we do a deep dive on the Proposed Architectural Framework (and mainly focus on fault tolerance and reducing amount of computational resources), followed by communication issue in section 3 and data storage architecture in section 4.

Novice moving average approach for data preservation during offline phases is also being introduced in section 3. Our final section Y is the summary and conclusions.

The main purpose of such system is to help user monitor and predict development of asthma and other respiratory diseases.

PROPOSED ARCHITECTURAL FRAMEWORK

The explosion of affordable sensors and wearable devices lead to growth of Personal mHealth, self-management of health conditions, and the collection of data, will radically change how health-care is delivered and information is collected. Let us consider these changes on the example of multi-scope infrastructure for mHealth and citizen-observation system (fig. 2). Air pollution in big cities and passive way of life combined with the abundant number of stress-factors caused the growing demand [7] on different health monitoring telecare systems and air-quality and user-breath monitoring systems in particular.

By gathering, combining and analyzing data collected from weather stations, Internet of Things (IoT) sensor networks, Body Sensor Networks (BSN), Personal Health Record (PHR) systems we can find patterns of disease development on early stages on signalize doctor on escalations.

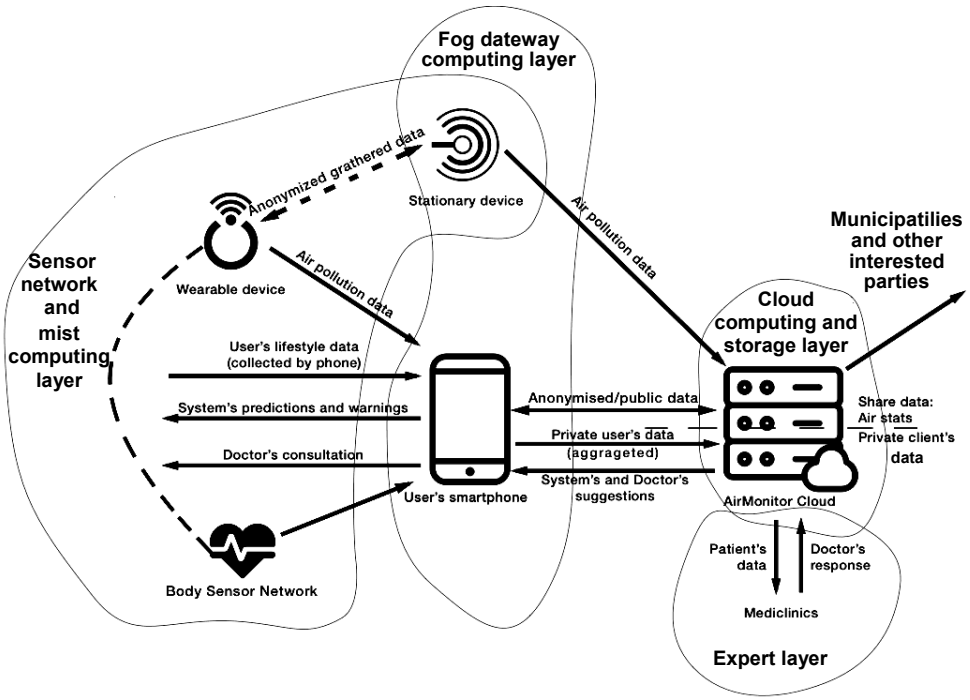

Fig. 2. Infrastructure for mHealth and citizen-observation system

The other scope of this system is citizen observation air-quality monitoring system. In addition to assistance to respiratory-sensitive people we found following usages of constant air-quality measurements for municipalities [8]:

- Commuting analyses based on CO2 levels at different time. This can result in improving city transportation quality and designing better living environments.
- Participation in life-quality indices for different districts, forecasting their development.
- Anomaly detection which can be the subject for further investigations.

As shown on Fig. 2 communication between edge-computing level and cloud is done through ***fog-computing gateway***. The IoT space of terminal endpoints in the discussed infrastructure includes the current smart phones, tablets, and laptops. While each one is quite an advanced technological piece, including sensors and cameras, we can ignore their internal complexity and regard them as simple points, providing connectivity to the person who owns them. Based on how the devices are connected to the patient, the devices can be classified into implantable, wearable, unconnected, or connected on need basis. By the end of 2023 95% of population worldwide will have the access to broadband mobile internet and smartphones [9] (Fig. 3). Respectively, user's mobile phone seems to be the perfect platform acting as gateway. This provides the following benefits:

- Available almost everywhere.
- Can aggregate data from different sources, such as internal motion sensors, GPS systems, provides API for different BSNs (Healthkit for iOS devices and Google Fit for Android correspondingly), can connect over Bluetooth or Bluetooth smart to external sensors. These provides location and situation awareness.

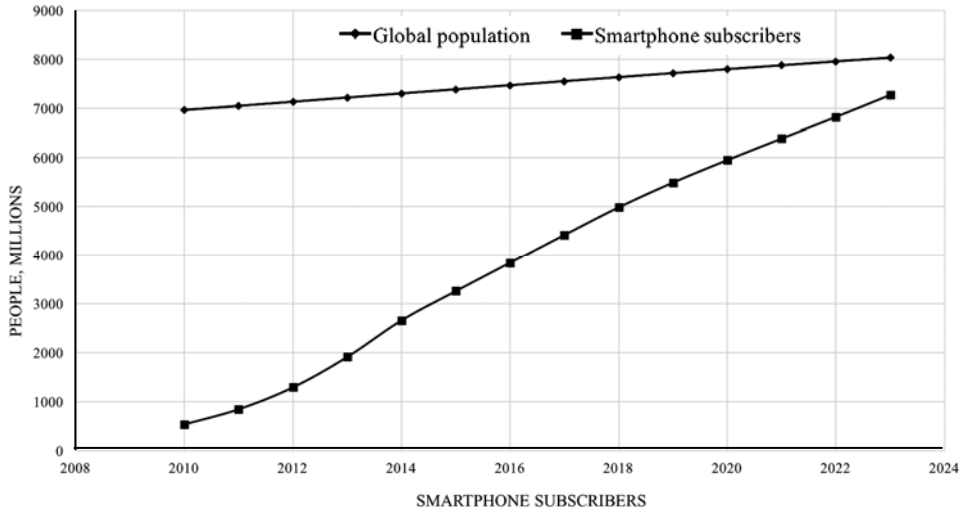

Fig. 3. Growth of smartphone usage compared to the global population [9, 10]

- Has enough computing power for basic data analyses including computing moving average and even running deep learning models.
- Can send data to cloud storage and processing and share analysis results with end-user.
- Provides instrumentation to secure user data available locally. Also, we need to admit, that this responsibility lies on user as he or she is required to lock personal device.

The key issue is the huge amount of data which is gathered by sensors. As smart sensor network (SSN) protocols like Bluetooth Low Energy, ANT and ZigBee have limited bandwidth compared to the volume of collected data it might be unfeasible to process it on fog-gateway level and store in the cloud. Also, there are latency limitations in telecare systems, where the delay in making decision can make it irrelevant and even endanger user. Following solutions are used to solve these tasks:

- Basic signal processing and feature selection are performed on smart sensor itself, significantly reducing amount of transferred data.
- Multi-party computation (MPC) approach is used for utilizing maximum resources and reducing latency (fig. 4).

In addition to mentioned above, by applying edge computing we can prevent sensitive data leakage on cloud level as large part of it is never transmitted to global network, but it is necessary to keep sensitive data anonymized during local transactions. For example, while sending air quality stats between wearable device and citizen-observation module we don't need to reveal user's identity. Another example is PHR system querying where both patient and hospital would like to preserve excessive data from each other and third parties. MPC makes possible sharing data for computation with only analysis result available, without revealing. For example, when exchanging air quality data only coordinates and features are shared without connection to the user identity, and parties receive only resulting prognosis without revealing each other's inputs.

The cloud level is designed as pluggable architecture with microservices approach and consists of:

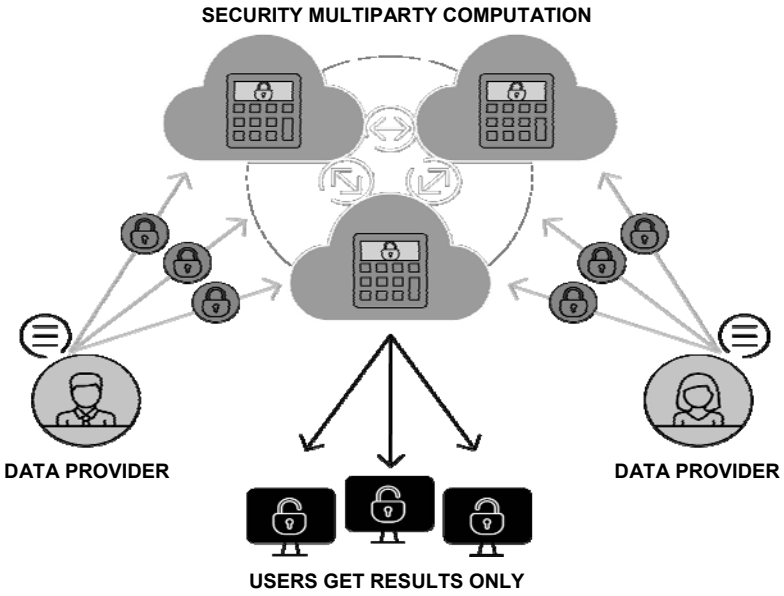

Fig. 4. Multi-party computation (MPC) approach [11]

- Cloud storage layer for storing data in different security layers with encryption solutions used for sensitive information.
- Cloud analytics layer for performing analysis on anonymized data using machine learning and deep learning models.
- PHR and telecare system integration service which is capable for integrating with external application programming interfaces (API) and providing the secure way to exchange sensitive data between physician and his patient.
- Public data service which provides the API for accessing and analyzing public data, such as air quality measurements, generalized medical statistics and analyses results.

Sensitive data in Cloud storage layer is stored in an anonymized way: set of states is bound to particular identifiers. Only state owner knows its identifier, so analysis system is unaware of extraneous data.

COMMUNICATION

As system has the lay`ered structure we should define protocols for communication inside each layer as well as cross-layer.

As we are using smartphone as fog-gateway device the best option on fog-level is to use *Bluetooth Low Energy (BLE)* as it is much more energy-efficient than communication over Wi-Fi and provides enough bandwidth considering data preprocessing on smart sensors. BLE application profiles are based on the generic attribute profile (GATT) [14], so data clients can subscribe only to certain characteristics (features) they are interested in and these features are emitted only when changed further reducing amount of transferred data.

For exchange between client device (fog-gateway) and cloud server it is proposed to apply to Remote Procedure Call approach and *Google's gRPC* framework in particular resulting in the following benefits [15]:

- Reducing data transfer over internet by using binary serialization (by default, Protobuf is used as interface description language).

- Offers client, server and bidirectional data streaming which is useful for constantly uploading air quality sensor data or patient's state from BSN during exacerbation of the disease.
- Deadline and timeout control allowing RPC be aborted and server resources freed when computed results are no longer valid to the client.
- Uses SSL/TLS to authenticate the server, and to encrypt all the data exchanged between the client and the server out-of-the-box.

This approach can also be used for cross-service communication in cloud level.

One of the main issues in designing Wireless Sensor Network (WSN) protocol is preserving data integrity. While data is kept confidential, we need to protect it from corrupting and system from men in the middle attacks [16]. Adversary can use spoofing and provide false data to the system, as his identity is hidden it might be difficult to track compromised party and prevent further intrusions.

As proposed system is designed in pluggable fashion we assume, that there could be possible situations that some of the devices will not be able to connect to the network for some amount of time, we have to be sure that data that was collected would not be lost. Due to insufficient amount of storage place on devices themselves, we propose to use technique of moving average to preserve valuable data. We propose implementing it as follows: part of data stream is stored on the device (depending on devise capacity, but assuming that we collect measurements with frequency 5 measurements per second, optimal will be to keep couple of seconds window on the device), and constantly updating while there is a connection. As the device loses connection, it calculates cumulative moving average [21] of stored window, and next measurements it calculates with the respect of already calculated value:

$$CMA_n = \frac{x_1 + \ldots + x_n}{n}; \quad CMA_{n+1} = \frac{x_{n+1} + nCMA_n}{n+1}. \tag{1}$$

Using these techniques will allow not to lose all the data collected while there would be no connection but obtain mean value of measured data through that time. Also, we have to mention that when connection will appear again, and mean value would be transferred to server (or other intermediate device), we will need to put a flag that indicates that it is a mean value, and also 2 timestamps – beginning and end of the created frame. This frame has variable size, as offline time is in inverse ratio with data detailing, so sensor readings detailing is always maximum available.

Proposed temporal storage algorithm consists of following steps:

- Acquiring list of sensor readings with timestamps.
- As we enclose the storage limits of the edge device, first elements are collided using Cumulative moving average (CMA), so the most recent data is also the most detailed.
- Above repeats after every data append until network connection is acquired again and data is being sent.
- At the same time, this technique should be used only when connection is lost, because as it can be seen from the (1) when calculating moving average, we lose measurement accuracy (while not using this feature we obtain no data from lost period, and overall data accuracy might be lower as a result).

For example, proposed approach was used, while testing distributed IoT network in our experiment with detecting breath patterns using CNN [22]. One part of it was system capable for air quality measurements — CO_2 ppm, dust level, temperature, pressure and humidity taken second with double precision for each value and timestamp, so the resulting size of one document is 44 bytes. Second module required data gathered from 2 smart accelerometer sensors 3 times per second and was used for respiration rate measurement – 3 axis values and timestamp (28 bytes). In the case of the connection loss this makes 158,4+604,8 = =713,2 KB per hour which exceeds memory resources for most wearable SOCs (for example, arduino pro mini used in experiment has only 32Kb RAM available). While recording data from human activity (one of the activities was running), people moved also in areas with poor internet connection. This caused delays and spaces in collecting data. Proposed approach with running average saved consistency of data with few accuracy loses (up to 9% in 5 minutes range) for respiration rate measurement. This proves that proposed approach with moving average works fine, especially with more static data like weather conditions preserving the most valuable measurements for both current context and historical analyses. However, when it comes to highly dynamic data like chest moving, it has main limitation — with time gap increasing, accuracy will decrease.

Researching utilization of different moving average calculation algorithms for different data types is a subject for further research.

DATA STORAGE

Blockchain might help addressing data integrity and trust problems mentioned above [2] [19], but there are a lot of limitations that need to be solved including limited resources that are not enough to properly support mining with keeping low-latency on low-power devices, large traffic overhead, and scalability issues.

It is obvious that blockchain itself can't be kept on the edge devices themselves, due to lack of computational resources and storage place. So, it is a good solution to distribute data storing and split it into saving data itself and metainformation about data, that could be quite sensitive – like PHR history and locations coupled with user id, or other personal information that could expose user's identity.

One of the possible solutions is to use off-chain database [16] together with blockchain storing references to data, but not the data itself. These perfectly combines with MPC as we share parts of secret data between computing nodes with no-party having the full picture. By saving trail of computations in public append-only bulletin-board as proposed in SPDZ protocol we allow auditing party to compare tail-of-proofs and check computing result correctness [18].

So, we suggest splitting data in the following way:

- Personal data – user name, sensor device identificator, cached user history.
- Public anonymized data – data from environmental sensors coupled with geo location, timestamp, and other aggregated collected data.

With this approach we can benefit from exposing data collected from sensors as well as anonymized health information to external services and integrating with them for analysis, research and monetization purposes.

To match sensitive information with particular patient, secret user key is proposed as identificator.

CONCLUSIONS AND FUTURE WORK

In this work, we examined the design of multi-scope service-oriented Mobile Healthcare systems. Proposed multi-layered approach offers pluggable architecture, good scalable and secures user's sensitive data by isolating their transfer on different levels. By using user's smartphone as wearable fog-gateway we achieved integration with wide range of Bluetooth-enabled smart sensors, situation and location-awareness combined with computing power sufficient for running deep-learning models [20]. At the same time we have proposed a novice solution of using moving average for storing data on IoT devices, while there is no connection to the outer world, that enables not to lose that data, but obtain mean values from that period of time with more detailing for more recent data.

The main issues of current implementation are:

- Preserving data integrity [13] and intrusion detection [17].
- Vulnerabilities to man in the middle attacks.
- Running computing consuming deep learning models on the edge level.

Planned future research includes:

- Applying blockchain for confirming transactions with preserving privacy on mist and fog computing levels.
- Designing secure communication protocol on the top of BLE and Bluetooth 5 using recently introduced Bluetooth mesh networking protocol.
- Optimization of deep learning models to be run on edge.
- Usage of smart sensors as intelligent agents in mist computing.
- Researching different moving average approaches to preserve various data types while network connection is lost with maximum result usefulness.
- Combining data from body-worn sensors (like heart rate sensor, accelerometer) for recognition of complex human activities and their outcomes regarding current and prognosed context by building Neural Network Models for it.

Depending on particular requirements the Distributed multi-scope service-oriented Mobile Healthcare system can be scaled from the corporate (national) scale of patients care to the scale of supporting profile patients in a particular region.

REFERENCES

1. *Jones Jake.* Edge Computing - The Cloud, The Fog And The Edge / Jake Jones. — Solidrun, 2017. — Avaliable at: https://www.solid-run.com/edge-computing-cloud-fog-edge/.
2. *Rahmani Amir M.* Fog Computing In The Internet Of Things / M. Rahmani Amir et al. — Cham, Springer, 2017. — 169.
3. *Maier Martin.* Context- And Self-Awareness In Fog And Mist Computing / Maier Martin, Mohammad Hossein Same. — 2017. — Avaliable at: http://www.zeitgeistlab.ca/doc/context_and_self_awareness_in_fog_and_mist_computing.html.

4. *Solidrun.* Fog And Mist Computing Levels. — 2017. — Avaliable at: https://www.solid-run.com/wp-content/uploads/2017/04/edge-computing-600x450.png.
5. *Diogenes Yuri.* Internet Of Things Security Architecture / Yuri Diogenes. — 2017. — Avaliable at: https://docs.microsoft.com/en-us/azure/iot-suite/iot-security-architecture.
6. *Song Tianyi.* A Privacy Preserving Communication Protocol For Iot Applications In Smart Homes / Tianyi Song // IEEE Internet Of Things Journal. — Vol 4, N. 6. — 2017. — P. 1844–1852. doi:10.1109/jiot.2017.2707489.
7. *Global* Structural Health Monitoring Market Demand, Growth & Revenue Opportunity (2016-2023). — 2017. — Avaliable at: https://www.researchnester.com/ reports/structural-health-monitoring-market-global-demand-analysis-opportunity-outlook-2023/173.
8. *CITI-SENSE*: Development Of Sensor-Based Citizens' Observatory Community For Improving Quality Of Life In Cities. — 2017. — Avaliable at: http://www.citi-sense.eu
9. *Ericsson* Mobility Report November 2017 – Ericsson. — 2017. — Avaliable at: https://www.ericsson.com/en/mobility-report/reports/november-2017.
10. *World* Population Prospects: The 2017 Revision, DVD Edition. — United Nations, Department of Economic and Social Affairs, Population Division, 2017. — Avaliable at: https://esa.un.org/unpd/wpp/Download/Standard/Population/
11. *Secure* Multiparty Computation. — 2017. — Avaliable at: https://partisia.com/ secure-simple-efficient/
12. *Secure* Multiparty Computation. — 2017. — Avaliable at: https://partisia.com/ secure-simple-efficient/
13. *Prosanta Gope.* BSN-Care: A Secure IoT-Based Modern Healthcare System Using Body Sensor Network / Prosanta Gope, Tzonelih Hwang // IEEE Sensors Journal. — Vol 16, N 5. — 2016. — P. 1368–1376.
14. *Bluetooth* Topology Options. — 2017. — Avaliable at: https://www.bluetooth.com/ bluetooth-technology/topology-options.
15. *Grpc* / Grpc Concepts. — 2017. — Avaliable at: https://grpc.io/docs/guides/ concepts.html.
16. *Zyskind G.* Enigma: Decentralized Computation Platform with Guaranteed Privacy/ G. Zyskind, O. Nathan, A. Pentland. — arXiv:1506.03471v1 [cs.CR], Jun. 2015.
17. *Stojmenovic Ivan.* The Fog Computing Paradigm: Scenarios And Security Issues / Ivan Stojmenovic, Wen Sheng // Proceedings Of The 2014 Federated Conference On Computer Science And Information Systems, 2014. — P. 1–8. — IEEE, doi:10.15439/2014f503.
18. *Damgrd Ivan.* Practical covertly secure MPC for dishonest majorityor: Breaking the SPDZ limits / Ivan Damgrd et al. // Computer SecurityESORICS 2013. Springer Berlin Heidelberg, 2013. — P. 1–18.
19. *Petrenko A.* Blockchain as a service for medical records / A. Petrenko, R. Kyisliy, I. Pysmennyi // System Research & Information Technologies. — 2017. — N 1. — P. 7–11.
20. *Baidu* Mobile Deep Learning. — 2017. — Avaliable at: https://github.com/ baidu/mobile-deep-learning.
21. *Weisstein Eric W.* Moving Average / Eric W. Weisstein // Wolfram MathWorld. — 2017. — Avaliable at: http://mathworld.wolfram.com/MovingAverage.html
22. *Petrenko A.* Human Respiration Pattern Detection Using Deep Convolutional Neural Networks / A. Petrenko, R. Kyslyi, I. Pysmennyi // Eastern-European Journal of Enterprise Technologies. — 2018. — N 4. — P. 6–13.

Received 22.01.2019

From the Editorial Board: the article corresponds completely to submitted manuscript.