

Designing a Cloud Tier for the IoT Platform EMULSION

IVAN GANCHEV*

Department of Computer Systems,
University of Plovdiv “Paisii Hilendarski”,
24 Tsar Asen St., Plovdiv 4000,
BULGARIA.

&

Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences,
Akad. G. Bonchev St., block 8, Sofia 1113, Bulgaria.
Ivan.Ganchev@ul.ie <https://orcid.org/0000-0003-0535-7087>

ZHANLIN JI

College of Information Engineering,
North China University of Science and Technology,
Caofeidian, Tangshan City,
CHINA.

Zhanlin.Ji@ieee.org <https://orcid.org/0000-0003-3527-3773>

MÁIRTÍN O'DROMA

*Telecommunications Research Centre (TRC),
University of Limerick (UL),
Plassey, National Technological Park, Co. Limerick,
IRELAND.

Mairtin.ODroma@ul.ie <https://orcid.org/0000-0002-1672-1668>

Abstract: - This paper presents some design aspects of the cloud tier of a generic multi-service cloud-based IoT operational platform EMULSION, which is developed as a non-expensive IoT platform primarily to serve the needs of small and medium business enterprises (SMEs). The EMULSION is a representative of the new horizontal type, next-generation, IoT platforms that come as a replacement of the existing vertical type platforms. The architectural design and main characteristics of the platform are presented and its multi-tiered structure is explained with particular attention paid to the cloud tier. This proposed cloud tier, with a Data Management Platform (DMP) based on a three-layer Lambda architecture, achieves improved high throughput and low latency. This is done through the inclusion of two distributed ‘publish-subscribe’ Kafka-based modules which are designed for data processing, and for data subscribing and message storage, respectively. Initial trials have begun with two pilot platform-demonstration IoT systems, utilizing this EMULSION platform. These are shortly to be presented in separate research papers.

Key-Words: - IoT, horizontal platform, service-oriented, heterogeneous, multi-tiered, cloud tier

1 Introduction

Emerging technologies such as the fifth-generation of cellular wireless communications (5G), Internet of Things (IoT), Big Data, and Cloud Computing have been receiving a lot of attention from designers, application developers, and researchers both in academia and industry. The introduction of 5G has changed the traditional service (including IoT) delivery model by integrating

applications within the programmable networks, as part of the Software-Defined Networking (SDN) and/or Network Functions Virtualization (NFV) architectures. In order to reap the full benefits of these different technologies when seamlessly integrated together, many networking-, protocol-, architectural-, service-, and interoperability challenges need to be addressed. This opens up good prospects for research work with respect to the development of novel, scalable, cost-effective, and

robust models, techniques, frameworks, platforms, and architectures that can efficiently and seamlessly support future diverse set of IoT services and applications with heterogeneous user-, device-, and network requirements.

In the light of this, the design aspects of a generic multi-service cloud-based IoT operational platform EMULSION are presented here. The EMULSION is a representative of the new *horizontal* type, next-generation, IoT platforms that come as a replacement of the existing *vertical* type platforms. The latter are dedicated platforms adapted to target customer domains and/or specialized for delivery of services within one particular IoT domain. With this former dedicated platform type, each IoT provider delivers entire vertical, involving separate applications/services, separate client connections, separate constrained networks, separate things/objects (Fig. 1). In fact, this approach led to the creation of an 'Internet of Silos' in the IoT domain, generating (too) many IoT solution platforms (450 by 2017 [1]), causing fragmentation and closed infrastructural boundaries thus hampering interoperability, integration of (new) components, increasing operating costs, and leading to problematic scaling, and constrained openness to new services. According to [1], the leading sector for IoT vertical-type platforms in 2017 was the Industrial/Manufacturing (32%), followed by Smart City (22%) and Smart Home verticals (21%).

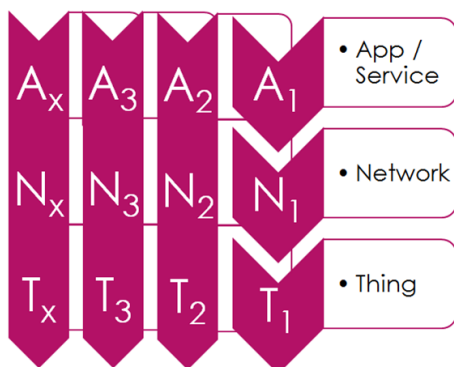


Fig. 1. Schematic of the *vertical* approach towards building IoT platforms.

The new generation of *horizontal* IoT platforms will counter this silo attribute, enable the satisfying of a number of important engineering requirements yielding attractive benefits, as summarized in the following. With them, each provider delivers horizontal slice (Fig. 2), meeting the requirements for flexibility, scalability, efficiency, and multi-purpose use, with easy adjustment to new scenarios and use

cases, and with efficient management of the entire ecosystem throughout the lifetime of the IoT services/applications. By using the integration and interoperability principles, this new approach is able to simplify the environment, remove duplicate solutions, enable inter-technology operation, and generate new services and new business opportunities.

The commercial IoT platforms currently available on the market are primarily of the connectivity management platform (CMP) type, such as Cisco Jasper, used to maintain IoT services primarily through the cellular operators' networks [2]. A step forward towards the next-generation IoT platforms is the HPE universal IoT platform [3], which is focused more on the use of infrastructure and hybrid cloud management solutions. This, however, could be prohibitive for smaller IoT service providers. In this sense, our aim with the EMULSION is to come up with a non-expensive, generic IoT platform, mainly for the needs of small and medium business enterprises (SMEs).

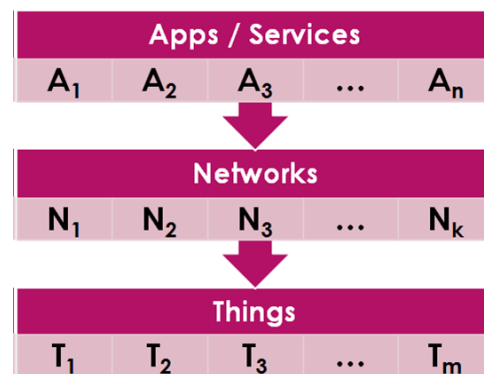


Fig. 2. The new *horizontal* approach for building IoT platforms.

The rest of the paper is structured as follows. Section 2 describes the main characteristics of the IoT platform EMULSION. Section 3 explains the different tiers forming the EMULSION structure. Section 4 presents some of the design aspects of the EMULSION's cloud tier. The description of the pilot IoT systems that are being designed and tested, based on the EMULSION, is provided in Section 5. Finally, Section 6 concludes the paper.

2 EMULSION Characteristics

The EMULSION exhibits the following important characteristics:

1) *Heterogeneity*: This feature allows integration of different types of IoT (including constrained) devices with heterogeneous processing components

and various communication capabilities and protocols, utilizing different access networking technologies, e.g. Long Range Wide Area Networks (LoRaWAN), wireless local area networks (WLANs) based on the IEEE 802.11 standard (Wi-Fi), wireless personal area networks (WPANs) based on the Bluetooth 5.1 standard, Low Power Wide Area Networks (LPWAN) based on the 3GPP standards LTE-M (Cat-M1) or Narrow-Band Internet of Things (NB-IoT). The aim is to allow 50,000 heterogeneous IoT devices to communicate simultaneously online with a single server. The platform is also able to support communication and data exchange with various IoT applications.

2) *Multi-tiered structure*: This is described in detail in Section 3.

3) *Service-oriented architecture*: In the future, this will also support programmability of composite services.

4) *Multi-service provision*.

5) *Flexible and easily expandable configuration*.

6) *Cloud-based architecture*: This is based on the ZooKeeper-based Redis framework and as such it has a Redis service governance architecture.

7) *Openness*: This is achieved at all tiers with possibility of building comprehensive ecosystems for different IoT application domains.

The ultimate goal is to build an IoT platform which is able to serve any IoT application domain, provide any type of service (including composite services), work with any device and any type of cloud.

Along with building the platform, novel models and techniques for the effective provision of IoT services are being elaborated, primarily targeting the 'smart environment protection' and 'smart health' IoT domains but also with an ability to be used in other IoT domains as well. Novel algorithms, data mining techniques, software solutions and mobile applications are being developed to provide the best quality of experience (QoE) to mobile users when utilizing different IoT services, accessible through any kind of mobile devices (e.g. smartphones, tablets, laptops, etc.) via heterogeneous access wireless networks, *anytime-anywhere-anyhow*. The platform will be able to provide each user with highly contextualized and personalized services by taking into account the current network- and service context, and the user preferences, and by utilizing 'big data' processing techniques. Taking into account the 'big data' aspect of information about (and gathered from) users, networks, and services, a cloud-based version

of the platform has been envisaged as more capable for facilitating the delivery of increasingly contextualized services to support the user-choice optimization process. Distributed data management techniques are being elaborated to provide an efficient computing environment for real-time, high-throughput 'big data' management and 'data mining' processing in the cloud tier. With a such distributed nature, the platform is able to turn real-world raw sensing data and mobile service activities (behavior) of users into actionable analytic datasets and proactively recommend the best service instances (e.g. the best driving/walking routes to follow in order to avoid polluted areas posing health risk) applicable to (and tailored to the preferences of) each individual user under the '*always best connected and best served*' (ABC&S) communications paradigm.

3 Multi-tiered Structure of EMULSION

3.1 Sensor Tier

This tier includes various types of sensor sets and stations for monitoring and reporting changes in the physical world. Fig. 3 depicts the developed hardware of an air quality index (AQI) monitoring point for use with the EMULSION.



Fig. 3. The developed AQI monitoring point's hardware.

3.2 Communication Tier

This tier encompasses various types of Data Transfer Units (DTUs) [4], collecting data from the sensor tier and sending these towards the cloud tier via communication gateways (each gateway may contain also a fog node to increase efficiency of the cloud). Communication in the opposite direction is used to send configuration information and/or control

commands to the sensor tier for the purposes of management and control.

For wireless communication, DTUs utilize different standards, e.g. LoRa, Bluetooth, Wi-Fi, 5G/4G/3G/2G, etc.

3.3 AmbiNet Tier

This tier has a network structure consisting of building blocks ('ambients') that model virtual-, physical-, and social objects of interest to a particular IoT service, along with their attributes, spatial, temporal, and event characteristics [5]. The AmbiNet model can be parameterized to the dynamically changing cyber-physical-social (CPS) world based on the information received from the sensor tier.

3.4 Client Tier

This tier contains different client applications and intelligent Personal Assistants (PAs) [6], functioning on the user mobile devices for facilitating the use of different services by taking into account the user's needs, personal preferences, and desires. The PAs support both the daily (operational) and the long-term management, execution, and control of different tasks, e.g. related to trip planning, reservations, health, payments, etc. Combining the PA concept with the IoT paradigm creates new opportunities for offering context-aware services to users. Being context-aware, PAs receive information about the CPS world through the AmbiNet tier. The PAs could be easily adapted to different IoT application areas, scenarios and use cases, and appropriately configured for various applications and mobile user devices as to efficiently enrich the use of IoT services by the users. The intelligent PAs can utilize the social media and other resources available online for self-learning and self-enhancement. Techniques from the field of artificial intelligence (AI) are employed to build intelligent PAs that independently perform tasks on behalf of users. PAs are implemented as software agents with a Belief-Desire-Intention (BDI) structure by means of the Java Agent Development Framework (JADE) and the rational agent layer Jadex (JADE extension), sitting on top of JADE, which allows easy development of rational agents.

3.5 Service Tier

This tier contains the regular electronic services for different groups of users (e.g. patients with different health status and requirements) along with additional recommendation services for generating and suggesting recommendations to users about the best

available instances of regular services. Examples of regular services include:

- Monitoring for (and predicting) changes in the environment (e.g. the AQI) and generating warnings to users about potential hazards;
- Planning and navigating on (dynamically changing) travel routes, e.g. to avoid areas presenting a risk to the health of the particular user-patient;
- Monitoring vital signs of patients and sending automatic alarms to hospital/medical center/doctor/physician/relative(s), etc., for urgent/immediate medical care/assistance;

The intention in this tier is to apply the new paradigm of the Internet of Services (IoS) [7] to enable the integration of different types of basic services (even supplied by different service providers) into one composite service (at the moment of initiating the user's request), while at the same time personalizing and customizing it to the particular user, taking into account all aspects of the current context for service contextualization.

3.6 Middleware Tier

Operating as a medium between other tiers (Fig. 4), this tier helps tackling the heterogeneity and integration of different technologies employed in order to achieve interoperability [8]. The middleware is concerned also with device- and data management, autonomous integration, programmability, and scalability issues [9]. It is the central manager of all IoT devices, registered to the platform, and as such it periodically checks their status. The middleware is also used to map IoT devices to services (and vice versa) and to schedule applications to the IoT devices under given resource constraints and performance requirements, as in [9].

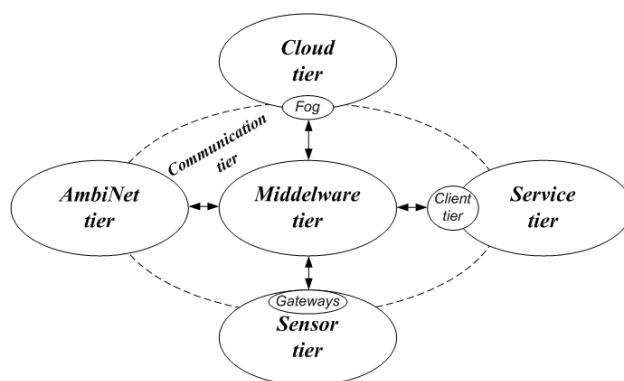


Fig. 4. The multi-tiered structure of the EMULSION platform.

The middleware tier acts also as a bridge between the IoT devices and IoT applications. The bridge

functionality is realized as a transparency server (in Java). A very robust *HashedWheelTimer* module is developed to enable each server to hold 500,000 IoT devices with a 2 Kernel / 8G virtual machine. Fig. 5 shows a high-level view of the developed bridge application.

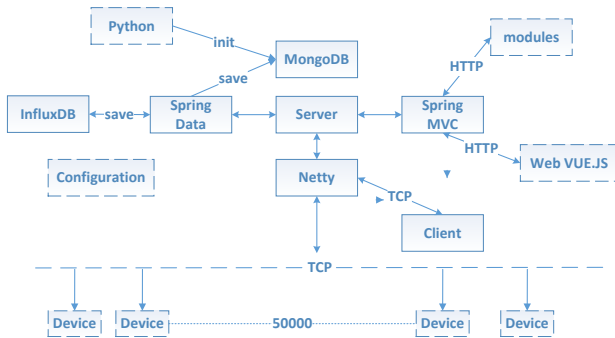


Fig. 5. A high-level view of the developed bridge application.

Beside communication with the IoT devices, the transparency server also communicates with user's mobile applications and service providers' web applications.

3.7 Cloud Tier

This tier has a cloud configuration used to mine the collected 'big data' in order to analyze and generate appropriate service recommendations for users using machine learning techniques. To this end, the cloud supports off-line computing and real-time applications (written in Python) for 'big data' analytics.

The research here is focused on the processing of large-scale and diverse in form (structured and unstructured) data, and semantic modeling of this data (primarily AQI data) with intelligent statistical methods and computer algorithms in order to uncover trends and dependencies of concentrations of major air pollutants, particularly harmful to human health, depending on the weather, meteorological conditions, etc., over different periods of time.

Additional fog nodes, located between the IoT devices and the cloud, are used to reduce the exchanged data volume, and to perform initial data analytics and knowledge generation close to the data source. In addition, the fog could be used to achieve transient and secure connection between IoT devices and the cloud, as suggested in [8].

The cloud tier is presented in more detail in the next section.

4 Cloud Tier's Design

The cloud tier provides 'big data' storage and computing resources for the IoT services. The distributed open-source Apache Hadoop framework (hadoop.apache.org) is utilized for the design of this tier as it provides an efficient Hadoop Distributed File System (HDFS) for storing 'big data' on different computers, and Map/Reduce programming models for 'big data' computing. Hadoop is highly reliable and scalable for parallel processing of 'big data' sets. The most recent data are stored in the Hadoop's Hbase [15] database to support real-time queries, whereas the old data are serialized to Hive [16] (the data warehouse in Hadoop) for subsequent data summarization, query, and analysis. For computing, a number of Map/Reduce algorithms [17] are used, such as a recommendation algorithm for suggesting the 'best' travel route avoiding the polluted areas, an algorithm for user profiles updating, etc. To build an efficient and scalable system, a rule engine Drools [18] is used to make decisions, based on facts, quickly and reliably.

Some of the functionalities of the IoT services are provided offline and depend on the Map/Reduce algorithms, e.g., the user profiles updating algorithms, etc., whereas others are provided in real time, e.g. travel routes' navigation. For the latter, the distributed dataset allows efficient data processing, whereby recommendations for the 'best' travel route could be delivered to different users from different servers. To meet this requirement, the cloud tier is developed with three clusters—Kafka [19], Storm [20], and HDFS [21]—as depicted in Fig. 6. Kafka is a high-throughput distributed messaging platform, used as a load-balancing cluster for parallel data loading into Hadoop. The Message Queues produce topics via multiple Brokers; topics are then consumed by the Storm Spouts. The Storm Supervisors maintain the topology, i.e., when a Spout receives topics from Kafka, the Bolts start processing the data (i.e., filtering, clustering, mining, etc.) in real time. Then the useful dataset is serialized to HBase in the Hadoop cluster. With the column-based HBase database, the IoT applications can access the database with *put*, *scan*, *add*, and *get* operations in real time. Hive [16] and Cloudera Impala [22] are utilized for data mining purposes.

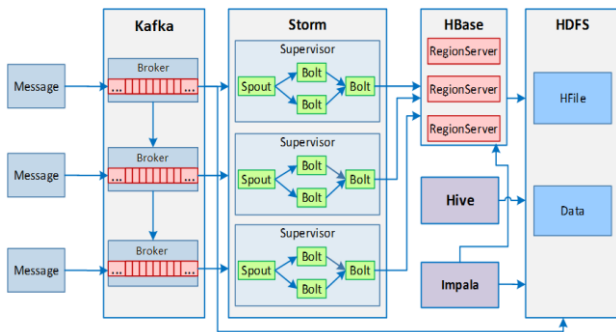


Fig. 6. The cloud tier's architecture

The Data Management Platform (DMP) is one of the key components of the cloud tier. As the IoT datasets are extremely large, the Lambda architecture with the following three layers is utilized for building a reasonable and effective DMP:

- 1) *Batch layer*: This supports the batch processing of the growing datasets in a distributed 'big data' file system by utilizing Map/Reduce operations and Hadoop [14]. A number of batch shell scripts are being designed for driving the Hive queries in the DMP.
- 2) *Serving layer*: This indexes and exposes the batch views in order to query the cloud in an ad-hoc way and with low latency. The Apache HBase technology is utilized as it offers random and real-time read/write access to the DMP database.
- 3) *Speed layer*: This provides real-time operations on recent IoT data and frequent updates to the serving layer by means of stream processing. The open-source distributed real-time computation Storm technology is utilized at this layer as it is capable of fast data processing at the rate of million tuples per second per node.

As the raw logs in the designed DMP are stored in a distributed fashion, to achieve high throughput, a Kafka-based distributed publish-subscribe messaging module is designed for use in the data stream part for data processing. The Apache Kafka publish-subscribe messaging platform allows a single Kafka module (broker) to handle hundreds of megabytes of reads/writes per second from thousands of clients simultaneously. In the query part, another Kafka-based module is designed for data subscribing and saving messages on a disk with ETL-batched consumption operations. It uses Camus (a simple Map/Reduce job developed by LinkedIn) to fetch available topics and store them on HDFS (the portable file system written in Java for the Hadoop framework).

A number of logs need to be recorded on the DMP, such as *Service Type*; *Service Session's Start Time, End Time, and Duration*; *Acct-Input-Packet*; *Acct-Output-Packet*; etc. During this process, supplementary services are loaded on the DMP for profiling of users and targeting of audiences. To support data serialization and remote procedure call (RPC) operations within a Hadoop environment, the Apache Avro technology is employed to provide the scripting languages.

The logs are first fetched and published by the Kafka Spout in a high-throughput distributed manner. Then the Storm Bolt consumes the log topics. After real-time tag analyzing and non-real-time statistics calculation, the output dataset of Storm is published to Kafka, including a number of topics. Fig. 7 depicts the corresponding data flow sequence diagram.

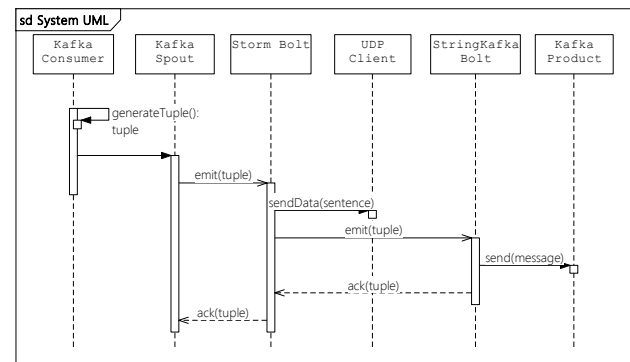


Fig. 7. The Kafka-Storm-Kafka data processing diagram.

- 1) *I/O Operation*: An Avro-based object is designed for data fetching. During the output operation, the messages in the Storm are first published into corresponding topics, which are then serialized as a byte stream to the HDFS and the Redis server.
- 2) *Storm*: The Storm part is focused on data subscribing and processing. Before obtaining data from Kafka, the defined Kafka Spout threads are first started in parallel and then linked with the defined topic's partitions for ingesting data.
- 3) *ETL*: The Camus acts as a pipeline between Kafka and HDFS in a way allowing every run of the Map/Reduce job to pick up where the previous run left off. An ETL job consumes topics from Kafka, formats the byte stream with a corresponding Avro schema, and serializes the Kafka message with a corresponding topic to HDFS.

In order to increase the performance of the cloud tier, a set of components for the provision of efficient

data services and a highly efficient interface, e.g. for the AQI publishing service, are being implemented.

To provide high-performance and on-demand IoT services to users, a key-value based NoSQL database—Redis [26]—is used in the cloud tier to provide scalable and distributed job queues. A distributed Redis cluster is developed consisting of two layers:

- 1) *Database layer*: A ZooKeeper-based cluster is used for distributed operations. To simplify the access to the Redis cluster, a functional *RedisExecutor* interface is designed with a Concurrent-HashMap algorithm for database's write/read operations;
- 2) *Service layer*: In order to provide the Redis service itself and a monitoring service for the upper layer, a service governance scheme is elaborated for exploring the services' interfaces and providing a distributed remote procedure call (RPC) mechanism. For the monitoring service, an Apache Curator framework is utilized to monitor the Redis server's connection status. When a Monitor is instantiated with an Extensible Markup Language (XML)-based configuration metadata, a Curator namespace is instantiated by the *CuratorFrameworkFactory.builder* function. Fig. 8 depicts the monitoring service' diagram.

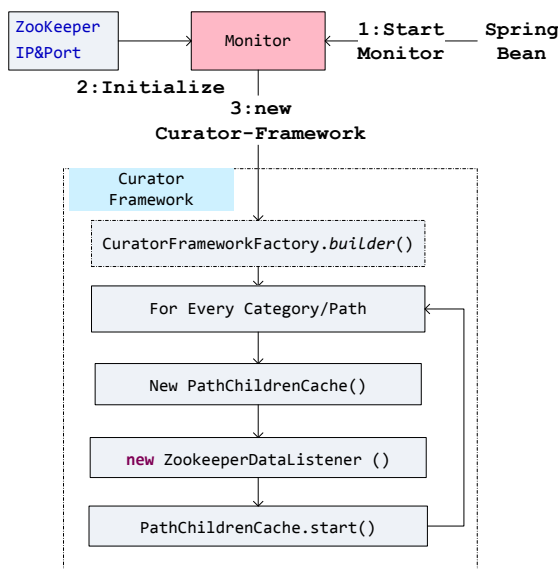


Fig. 8. The monitoring service' diagram.

5 Pilot IoT systems

Two pilot IoT systems are currently being designed to work on top of the EMULSION platform.

5.1 Smart AQI Sensor Control System

The main goal of this pilot system is to provide live environmental data, supplemented by AQI forecasting information, as an input to the other pilot system, along with a corresponding mobile application serving patients with different health problems and assisting them in advanced (proactive) planning of routes, optimized in accordance with the current user context, for travelling with a minimum health risk. The routes are pre-planned, based on short-time forecasting techniques [10, 11], and could be dynamically changed 'on the fly' depending on the current weather/AQI conditions.

5.2 Smart uHealth System

The main goal of this pilot system is to provide various ubiquitous healthcare (uHealth) services, optimized for delivery to different categories of patients in accordance with their current context. The covered uHealth services range from basic daily activity and community services to more complex ones, e.g. involving fall detection and life threatening situations, and requiring special care and human intervention, with supplementary Global Positioning System (GPS) tracking of patient in case of emergency. For tracing changes in the health status of the patients, generalized net models of processes could be utilized as proposed in [12, 13].

6 Conclusion

This paper has presented a proposal with details for a new and different structural approach towards IoT platform design and implementation. It moves away from the present-day autonomous vertical structure platform type to a horizontal one which we call EMULSION. The engineering platform design requirements it satisfies include flexibility, multidimensional scalability, interoperability, and multi-purpose use, with easy adjustment to new scenarios and use cases, and with efficient management of the entire ecosystem throughout the lifetime of the IoT services and applications. Some immediate benefits include simplification of the IoT environment, removal/elimination of duplicate infrastructures (which are characteristic of vertical IoT platform solutions), enablement of inter-technology interoperability, openness to new services with minimal additional infrastructure change, and thus openness to new business opportunities.

In the cloud tier, a Data Management Platform (DMP), based on a three-layer Lambda architecture, has been proposed. To achieve high throughput and low latency, two distributed 'publish-subscribe'

Kafka-based modules are designed for data processing, and for data subscribing and message storage, respectively. The Apache Avro technology is utilized to support data serialization and RPC operations within a Hadoop environment. The Storm bolt technology is used to consume log topics. The performance of the Kafka-Storm-Kafka part of the DMP depends strongly on the composition of the Kafka Brokers and Storm Supervisors employed in the cloud. This will be a research topic for future studies.

Acknowledgment

This publication has emanated from a research conducted with the financial support of the Bulgarian National Science Fund (BNSF) under the Grant No. КП-06-ИП-КИТАЙ/1.

References:

- [1] W. Zaña Diaz, IoT platforms company list 2017 update, IoT Analytics. 2018. <https://iot-analytics.com/iot-platforms-company-list-2017-update/>
- [2] T. Glushkova, S. Stoyanov, A. Stoyanova-Doycheva, V. Ivanova, and L. Doukovska, AmbiNet – an environment for ambient-oriented modeling, *International Journal of Computing*, Vol.18, No.3, 2019, pp. 331-340.
- [3] W. Razouk, D. Sgandurra, and K. Sakurai, A new security middleware architecture based on fog computing and cloud to support IoT constrained devices, 1st International Conference on Internet of Things and Machine Learning (IML). ACM, 2017, pp. x1-x8.
- [4] H. Lee, et al., A novel service-oriented platform for the Internet of Things, 7th International Conference on the Internet of Things (ICITEE). ACM, 2017, pp. x1-x8.
- [5] I. Ganchev, Z. Ji, and M. O'Droma, Designing a low-cost data transfer unit for use in IoT applications, 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2016), 2016, pp. 85-88.
- [6] V. Valkanov, J. Todorov, B. Daskalov, N. Stancheva, and S. Stoyanov, Personal assistant with ontology based knowledge, International Conference "Big Data, Knowledge and Control Systems Engineering", 2017.
- [7] I. Ganchev, R. D. van der Mei, and H. van den Berg, (Eds.) *Autonomous Control for a Reliable Internet of Services: Methods, Models, Approaches, Techniques, Algorithms, and Tools*, Lecture Notes in Computer Science (LNCS 10768), Springer, Cham, 2018, pp. 401.
- [8] A. Ali, IoT forecast: operator spend on connectivity management platforms 2017–2026, KNOWLEDGE CENTRE. 2018. <http://www.analysismason.com/Research/Content/Regional-forecasts-/IoT-CMP-forecast-RMA17/>
- [9] HP Enterprise, Delivering on the IoT customer experience: The HPE Universal IoT Platform 1.4, <https://h20195.www2.hp.com/v2/GetPDF.aspx/4AA6-5353ENW.pdf>
- [10] A. V. Ivanov and S. G. Gocheva-Ilieva, Short-time particulate matter PM10 forecasts using predictive modeling techniques, 5th Conference of the Euro-American Consortium for Promoting the Application of Mathematics in Technical and Natural Sciences (AMiTaNS'2013), Vol.1561, 2013, pp. 209-218.
- [11] S. G. Gocheva-Ilieva, A. V. Ivanov, and I. P. Iliev, Modeling of air pollutants and ozone concentration by using multivariate analysis: Case study of Dimitrovgrad, Bulgaria, *British Journal of Applied Science & Technology*, Vol.14, No.3, 2016, pp. 1-8.
- [12] V. Andonov, M. Stefanova-Pavlova, T. Stojanov, M. Angelova, G. Cook, B. Klein, K. Atanassov, and P. Vassilev, Generalized net model for telehealth services, 6th IEEE International Conference "Intelligent Systems", 2012, pp. 221-224.
- [13] M. Stefanova-Pavlova, V. Andonov, T. Stojanov, M. Angelova, G. Cook, B. Klein, P. Vassilev, and E. Stefanova, Modeling telehealth services with generalized nets, In: Sgurev V., Yager R., Kacprzyk J., Atanassov K. (eds) *Recent Contributions in Intelligent Systems. Studies in Computational Intelligence*, Vol.657, Springer, Cham, 2017, pp. 279-290.
- [14] M. Pastorelli, D. Carra, M. Dell'Amico, and P. Michiardi, HFSP: Bringing size-based scheduling to Hadoop, *IEEE Transactions on Cloud Computing*, Vol.5, No.1, 2017, pp. 43-56.
- [15] L. George, *Hbase: The Definitive Guide*. O'Reilly Media Inc.: Sebastopol, CA, USA, 2011.
- [16] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, R. Murthy, Hive – A petabyte scale data warehouse using Hadoop. <http://infolab.stanford.edu/~ragho/hive-icde2010.pdf>

- [17] J. Dean, S. Ghemawat, MapReduce: A flexible data processing tool. *ACM Commun.* Vol.53, No.1, 2010, pp. 72-77.
- [18] B. Michal, *Drools JBoss Rules 5.0 Developer's Guide*; Packt Publishing Ltd.: Birmingham, UK, 2009.
- [19] J. Kreps, N. Narkhede, J. Rao, In Kafka: A distributed messaging system for log processing. 6th International Workshop on “Networking Meets Databases”, 2011, pp. x.1-x.7.
- [20] N. Marz, Storm – Distributed and fault-tolerant realtime computation.
<http://cloud.berkeley.edu/data/storm-berkeley.pdf>
- [21] D. Borthakur, *The Hadoop Distributed File System: Architecture and Design*, The Apache Software Foundation: Forest Hill, MD, USA, 2007.
- [22] A. Chauhan, *Learning Cloudera Impala*; Packt Publishing Ltd., Birmingham, UK, 2013.