Architectural concepts for managing biomedical sensor data utilised for medical diagnosis and patient remote care.

Mariusz Chmielewski^{1,a}, Damian Frąszczak^{1,b}, Dawid Bugajewski^{1,c}

¹Military University of Technology, Cybernetics Faculty, gen. W. Urbanowicz Street 2, Warsaw, Poland

Abstract. This paper discusses experiences and architectural concepts developed and tested aimed at acquisition and processing of biomedical data in large scale system for elderly (patients) monitoring. Major assumptions for the research included utilisation of wearable and mobile technologies, supporting maximum number of inertial and biomedical data to support decision algorithms. Although medical diagnostics and decision algorithms have not been the main aim of the research, this preliminary phase was crucial to test capabilities of existing off-the-shelf technologies and functional responsibilities of system's logic components. Architecture variants contained several schemes for data processing moving the responsibility for signal feature extraction, data classification and pattern recognition from wearable to mobile up to server facilities. Analysis of transmission and processing delays provided architecture variants pros and cons but most of all knowledge about applicability in medical, military and fitness domains. To evaluate and construct architecture, a set of alternative technology stacks and quantitative measures has been defined. The major architecture characteristics (high availability, scalability, reliability) have been defined imposing asynchronous processing of sensor data, efficient data representation, iterative reporting, event-driven processing, restricting pulling operations. Sensor data processing persist the original data on handhelds but is mainly aimed at extracting chosen set of signal features calculated for specific time windows – varying for analysed signals and the sensor data acquisition rates. Long term monitoring of patients requires also development of mechanisms, which probe the patient and in case of detecting anomalies or drastic characteristic changes tune the data acquisition process. This paper describes experiences connected with design of scalable decision support tool and evaluation techniques for architectural concepts implemented within the mobile and server software.

1 Introduction and motivation

The availability of mobile technologies and wearable class sensors provide new means of medical assistance and health monitoring. In majority cases medical facility based assistance requires specialised, expensive and cumbersome medical equipment. Alternative patient's monitoring approach can be designed using wearables which provide mobility, simplicity of application and usage and sacrificed with lower precision. Such approach is especially valuable during continuous, recurrent data acquisition process which can provide Constructed software platform is a framework for wearable sensor integration and has been designed to host signal processing and medical diagnostics algorithms. As a result configurable and tuneable composition of sensor signals, tailored signal features and analytical algorithms provide base line for development of specialised health related monitoring systems, supporting selected medical conditions. These conditions can be directly or indirectly detected by application of chosen set of biometric signals such as electrocardiography, electromyography, photoplethysmography (multispectral), electrodermal

activity and actigraphy [2],[3]. Combination of several data sources and correlation of biomedical parameters between each other and specific patient's health state, extend predefined ordinarily used event triggering rules. This work provides construction details of wearable-based mobile systems with specialised aimed at health state identification and monitoring. Undertaken construction decisions have been confirmed and justified based on functional and stress tests of system components. The first deployment attempt of designed architecture and its implementation has been aimed at remote, mobile monitoring of elderly people, and preconfigured for crucial health event recognition – fainting, stroke, cardiac arrest, seizures and derivatives of such conditions. The main motivation for this work was to construct and test scalable software architecture with tuning, calibration and extension capabilities mainly aimed at sensor data processing and rich health recognition algorithms (decision support). The platform should deliver these capabilities through asynchronous processing schemes minimising blocking functionality and automating background data reporting and synchronisation.

^a Mariusz Chmielewski: mariusz.chmielewski@wat.edu.pl

^b Damian Frąszczak: damian.fraszczak@wat.edu.pl

^c Dawid Bugajewski: dawid.bugajewski@wat.edu.pl

2 State of the art, technology stacks

Utilisation of biomedical sensor data for diagnosis and remote care requires overcoming many challenges regarding data acquisition, analysis and visualisation [1]. Usually such data is gathered with usage of certified medical sensors. Nevertheless, recent survey shows that non-medical grade wearable sensors provide heart rate measurement accuracy close to medical grade devices while being considerably cheaper [2].

Wearable sensors design and its utilization for monitoring has become a very popular research direction [3]. Some of the system designs include using mobile for data acquisition and its further processing on server [4]. However there are significantly less systems where data could be collected from heterogeneous sources and then processed especially where an enterprise ready architecture would be in scope of research.

Nowadays only two major platforms are available when it comes to smartphones – iOS and Android. Operation system dedicated to medical diagnosis and patient remote care should fulfil following requirements:

- high popularity among users,
- high security,
- allow creating reliable, constantly running background services,
- allow integration of application with system (disallowing closing while monitoring, starting with system boot).

Out of the two mobile operation systems, recommended and proposed is Android, the more popular and contrary to iOS, it delivers more flexible access to low level API, supports background running services, deep integration with system.

Fig. 1. Share of the smartphone market sales by OS. [5]

During wearable devices market scouting over 30 devices have been compared considering number of parameters such as amount of data sources, price, battery runtime, interfaces to integrate with external systems and solution maturity. Three sensors have been selected as most suitable for developed system: Microsoft Band 2, Samsung Simband and Empatica E4. All three of them allow to measure heart rate, galvanic skin response or hand movement. Furthermore Microsoft Band 2 has GPS sensor and delivers pre-processed data regarding activity, pedometer and sleep quality.

The next step was to acquire all selected devices, test them and compare their functionality, accuracy, usability, but most of all integration capabilities and real life performance. Unfortunately it was hard to obtain one

of more functional smartbands: Empatica E4 or Samsung Simband in reasonable time. Because these restrictions and multiple datasources hosted by Microsoft Band 2 (Fig.2), this device was selected as primary wearable multi-sensor.

Fig. 2. Wearable multi-sensor (Microsoft Band 2) device, used for data acquisition and processing providing GPS (5), PPG (2), GSR (6), inertial (1), barometer (5), UV light (3) data sources [6]. MSBAND2 integrates with handheld through dedicated API (in 2016 deprecated)

3 Method and technology requirements

Health monitoring systems have significantly developed to change the way health care is delivered. Nowadays, health monitoring systems are available everywhere including personal mobile phones. Some comprehensive reviews have been presented in literature, mostly focusing on a global overview of the topic [2] [5] [6]. Data security and patient confidentiality are the most significant things to consider during modelling a monitoring system. These systems utilize network connection to continuously and remotely send patient data to datastore. According to [2] [5] medical data security is the main priority when considering telemedicine applications. Application security layer should provide mechanisms providing a prevention of patient data disclosure and only authorised access to patient medical records [5].

Another criteria of patient monitoring systems should concern: reliability, efficiency and context awareness [6]. Some reviews have been reported in literature that utilizing GPRS to transfer sensor data can easily lead to high costs [7]. According to conducted research and prepared system's vision, it should provide compact and highly effective data communication interface with central SOA based server with exposed Restful API. The analytical tools should be provided only in form of web portal supplemented reporting functionalities delivered as external component.

4 Architectural concepts

System has been designed using layered architecture concept applying event-driven approach, which mainly supports efficient event reporting and sensor data

reporting (containing several hours of persisted data). The novelty of constructed approach comes from the composition of sensor data processing and applied technology stack. Main processing component of the system is the mobile application, which manages the configuration channel to integrated wearables (Microsoft Band 2, Microsoft Band 1, MIO Fuse), acquires the sensor data and performs filtering and preliminary processing aimed at recognition of critical conditions. Mobile application acts as a background supervisor, thus the patient is not required to run the application, instead background services communicating with multi-sensors gather the data and recognise patient's health events. Mobile software contains framework for implementing decision algorithms associated with sensor data (inertial, HRM, skin temperature and conductance). Constructed implementation framework deliver means to process signal type of sensor data but also provide means for performing sensor data fusion (complementary, cooperative). To extend the analytical capabilities of the system, a set of integration services has been provided and dedicated web portal for preliminary sensor data analysis. Constructed data sources and stored sensor data can be additionally processed by external modules and analytical tools specialised to perform data-specific analysis – ReportingServices, Matlab or any other tool supporting signal processing algorithms and classification methods. In order to avoid mobile application inefficiency and excessive power consumption some restrictions needed to be introduced concerning health state classification performed by machine learning techniques. In order to support such features, the architecture delivers two stage processing the first performed by decision algorithms on mobile device and the later supported by the analytical services on server. The central component supplements also algorithms, which utilise aggregated data from other patients and the whole monitored population of users which deliver more complex processing schemes. System delivers asynchronous processing, based on the sensor data report queues, on which set of processing threads operate.

5 Technology requirements – mobile

Collection of sensor data in real time requires efficient storage. Most common data storage frameworks for Android have been compared using a customised benchmark based on [8]. Classes used to evaluate frameworks speed have been shown on (Fig. 3).

Fig. 3. Entity classes used for evaluation in constructed testbed framework processing efficiency

Fig. 4. Benchmark tests for read-write and query operations involving both structured and unstructured data.

Fig.4. demonstrates that the most efficient way of data processing has been provided in form of pure SQL queries, which have been further optimized. Unfortunately, this approach requires significant knowledge regarding SQL queries optimization and SQLite database itself. Second solution in terms of write speed was Realm, which delivers rapid development features and requires less implementation experience than the fastest approach. Because of the previous as well as prior experience of developers in using Realm database it has been selected as data storage in developed system.

Data acquisition should be energy efficient therefore Android version should be at least version 4.3 (API 18) due to Bluetooth LE support. According to Google [9] less than 5% Android devices run older versions. Market distribution of OS versions has been shown on (Fig. 5).

To fully utilize Bluetooth LE and allow data collection from heterogeneous sources GATT support was introduced. Generic Attributes (GATT) [10] is a Bluetooth LE standard that allow maintaining interoperability between Bluetooth devices by exposing services within device and characteristics provided by each service. This allows to discover what capabilities certain device offers and subscribe to ones that provide desired data.

Fig. 5. Android API segmentation based on source: [11].

6 Technology requirements and solutions – for services (API) and analytical portal

One of the most important aspect of implementing an integration mechanism (delivered as API) of the central server to allow retention of geolocation, inertial and biomedical data obtained from mobile devices is to provide a quick, highly available and an efficient mechanism for storing large sensory data sets.

Table 1. Performance test result for writing data

Database Type	Number of trials	Average time [s]	Max time [s]	Standard deviation [s]
SQL	1000	1,07	2,80	0,42
NoSQL	1000	0,98	2,02	0,28

Table 2. Performance test result for reading data

Database type	Number of trials	Average time [s]	Max time [s]	Standard deviation [s]
SQL	100	4,25	4,58	0,10
NoSQL	100	0,03	0,03	0,09

Proposed system utilizes NoSQL database Apache Cassandra optimized for huge amounts of data [11]. Casandra is an open-source, column-oriented database with a data model inspired by Google's BigTable and distribution flow based on Amazon's Dynamo [12]. It has been released in 2008 by Facebook as a search engine. In 2009 it has been released as Apache project [12].

We have performed a performance test to compare available SQL and NoSQL databases. We have chosen a recording and reading time of large amounts of data as main metrics. Results of our tests has been shown in above tables. This result has been confirmed thesis available in many researches [13] [14] about a superiority of NoSQL

databases over SQL databases in the reading of large data volumes.

Data model in Cassandra should focus on the query-driven approach not around relationships. Apache Cassandra is optimized for high write performance to create database schema where each table is the equivalent of sophisticated SQL query with JOIN, GROUP BY, OR clauses. In the Cassandra world data is denormalized and data duplication exists [12].

Fig. 6. NoSQL query model for management user sensor data

One of the predefined, sophisticated query is modelled on figure below. It is responsible for storing and querying user sensor data via a primary key represented by session UUID. While modelling data model in Cassandra it is worth to remember that it uses a Java Map structure to retrieve data by primary key.

Implemented system collects biomedical sensor data in real time utilising biomedical sensors such as: MS Band (14 different sources) and FreeStyleLibre (1 source). One day of fully performed system protocol provides about 200 MB raw data of one person. We have proposed a solution able to fast data transfer and persisting on the server in real time which is able to minimize load on the server and the patient's network.

We have proposed a REST API model to transfer zipped data utilising HTTP protocol and utilizing Java Messaging API (JMS) to asynchronously process

uploaded data. The whole process consists of the following steps:

1. The client sends an HTTP POST request with the file parameter of the multipart type. Raw data is serialized to JSON format and zipped before sending to server;
2. The server forwards the request to the logical application controller responsible for sensor data receiving - *SensorDataController*. This controller is invoked by *DispatcherServlet* (implemented in Spring MVC) responsible for forwarding all requests in the application to correct units – *DispatcherServlet* design pattern.

Fig. 7. Sequence diagram for constructed asynchronous sensor data processing – 1 of 2

3. The *SensorDataController* recognizes the user performing a request via a token and forwards the transferred file to the *UserSensorDataUploadService* service responsible for persisting physically file on the server storage.

Fig. 8. Sequence diagram for asynchronously sensor data processing - part 2

4. The *UserSensorDataUploadService* service utilizing the *FileManager* API persists file on the server and then an asynchronous message is sent

via JMS queue to allow a further processing of the data to take place without the need to maintain the server's connection with the user. At this moment, the user is notified of the result of the action.

5. The *UserSensorDataPersistenceManager* service is responsible for the basic data preparation and persisting in the storage. It receives an JMS message, extracts data to temporary directory and saves in NoSQL database utilizing *SensorDataEventRepository*, *SensorDataSessionByDateRepository* and *SensorDataSessionByUuidRepository* repositories.
6. The *UserSensorSimplifiedDataPersistence* service is invoked to execute data analysis, optimization and persisting aggregated data in NoSQL database. It is end of data processing on the central server.

Web portal integration mechanisms utilize several integration, optimization and data fusion approaches to achieve optimal, fully automatic effect of storing and processing patient information. In our solution we have used Douglas-Peucker algorithm to minimize number of both sensor and geolocation points [15] and Microsoft Excel to provide a basic analytical tool. We have prepared automatically created algorithms for data regression and classification based on sensor data via Apache POI API. Automatically prepared Excel file provides complicated and pre-defined views in the form of graphs.

7 Evaluation of architectural concepts

Web portal services have been developed in J2EE environment to provide scalable, reliable and secure application utilizing Spring Boot as a web application framework. Business logic has been implemented according to SOA approach utilizing Spring MVC Controllers, web security has been improved via Spring Security module.

Multilayer divided architecture allocates different responsibilities to separate layers improving flexibility and maintenance factors. Web services are used both by mobile and web client to communicate with server to process sensor data of patient. Each endpoint has been implemented as a Business Delegate object's over HTTP (Hyper-text Transfer Protocol) requests utilizing Spring MVC DispatcherServlet. Web controllers are responsible for process request in particular format and transfer responsibility for logic execution to appropriate service in a Spring container. REST services utilize JAX-RS and Jackson libraries to provide communication base both for JSON (JavaScript Object Notation) and XML (Extensible Markup Language) communication format.

The persistence layer has been implemented according to Repository pattern both for relational and NoSQL database utilizing Spring Data repositories as framework for DAL (Data Access Layer). Spring Data is a library focused on providing a code required to implement the data access layers across different

persistence mechanisms [16]. To provide SQL database communication we have utilized JPA 2.1 (Java Persistence API) standard which defines a specification for persisting, accessing and managing the data between Java objects and the relational database as an ORM (Object-Relational Mapping).

Fig. 9. Biomedical data analysis utilising exported CVS data in Microsoft Excel environment

Hibernate has been chosen as a JPA implementation provider utilizing Spring Data JpaRepository abstraction. To provide NoSQL data access layer we have used Datastax Java Driver and Spring CassandraRepository abstraction. Our solution utilizes a data source pooling component to provide highly efficient DAL of memory allocation and request's speed to the database. The security layer is implemented according to SSO (Single Sign On) recommendations utilizing Spring Security module to simplify distributed and multi-platform authentication. User request authentication and authorization request is verified by HTTP filter depending on existing HTTP headers described server session. To

provide stateless and fine-grained access control we have used a JSON Web Token (JWT) which is an open standard (RFC 7519) for defining a method for securely transmitting information via a JSON object.

Fig. 10. Analytical portal operating on stored biomedical and inertial sensor data, supplemented with structured metadata

One of the most advantage of JWT is its compact size allowing tokens to be easily transmitted via query strings, header attributes and within the body of a POST request [17]. This approach simplifies patient access to system and shorten time needed for both registration and authentication. According to web services REST standard data is transferred over HTTP requests with plain text. The presented system utilizes heterogeneous data sources with implementation of data fusion methods to improve patient state awareness. One of the most advantages of this system is a possibility to collect, store, process and query each type of data. It is achieved via proposed generic services, repositories and NoSQL data model. NoSQL database exceeds relational database in storing and processing unstructured data. Currently this system collects data from devices such as: mobile phone with Android IOS, MS Band 2 smart band and FreeStyle Libre glucose monitoring sensor. The system can be easily extended to gather other biomedical data without database

schema change. Data connected to user via biomedical session is stored via Java Map interface storing gathered data via associations between sensor type as map key via measurement data as a Map value.

Fig. 11. Deployment perspective – system's architecture

The system has defined 54 available sensor types to be gathered on the server. To introduce a new type of sensor it is only needed to extend this list. To closely integrate mobile application with Android OS two Broadcast Receivers were used. One of them was listening to boot message to start application along with the Android. The second one was designed to bring application back alive after it was shut down if system was short on memory. Mobile system architecture has been shown on **Fig. 12** The major functionalities have been divided into components, which simplified the development and deliver flexible extensions for future wearable integration. To closely integrate mobile application with Android OS two Broadcast Receivers were used. One of them was listening to boot message to start application along with the Android. The second one was designed to bring application back alive after it was shut down if system was short on memory. Application restoration process was based on scheduling a custom broadcast to be sent after application was recycled. When the broadcast was send Android is creating a new instance of application allowing to restore monitoring without user intervention. This action is only taken after user enables monitoring. Developed applications allows for users login using system account or Single-Sign-On. Alerts are triggered by abnormal patient states as well as manually

by patient. MD can log into application and analyse measurement results as well as its aggregates. Some of the application screens have been shown on **Fig. 13**.

Fig. 12 Mobile system architecture

8 Conclusion

This paper presents results of implementing a scalable system architecture focused on mobile biomedical signal monitoring. The product of this work provides an architecture containing a requirements model including a logical and physical perspectives of the developed system components such as: an integrated biomedical band, glycaemic monitoring system, Android mobile application, server integration services with the portal able to manage obtained personal, geolocation and biomedical data.

System implementation required the development of a dedicated domain system model that includes medical profile data, patient scheduling, geolocation and both inertial and biomedical sensor data sets. Efficient processing of large data sets has been achieved via the NoSQL database responsible for storing geolocation and sensor data. We have also used a relational database to

support storing the medical profile, configuration and event data. The mobile application has been designed as a tool strongly integrated with the smartphone's operating system providing the reliability of communication with the band, monitoring and event data reporting to the central server.

Fig. 13 Mobile application views presenting simplified sign-in/sign-up, dashboard data, daily health state estimates, automatic event-driven emergency mechanisms.

The proposed mobile application utilizes heterogeneous data sources includes: Microsoft BS Band 2 integrated with Microsoft Health API, FreeStyleLibre Sensor via NFC protocol and mobile phone sensors with Android OS. It is providing user authorization and authentication functions, sensors configurations, patient profile management and biomedical data monitoring mechanisms with reliable wireless transmission to server services. One of the most important things is fact that great emphasis was placed on the development of technologies responsible for obtaining, storing, processing and biomedical data visualisation for monitoring people at an advanced age. The proposed portal allows the presentation and analysis of stored biomedical data on online charts and in Excel spreadsheet providing automatically generated regression and

classification mechanisms and geolocation data visualization on the maps.

We have prepared and performed a set of test cases to verify the correctness of the developed technologies. Finally, we have performed load tests of server services to perform validation of architecture and preparation of guidelines for scaling the solution in further stages of development works. Currently the systems is being adopted for integration with other biomedical sensors such as: MYO and glucometers.

Acknowledgements: This work was partially supported by grants RMN-815, RMN-817 and EU research project PUB-01-423 entitled: Development of innovative technology for the acquisition and integration of geolocation and biomedical data of monitored patients.

References

- [1] A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci i S. Andreescu, „Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges,” *Services Computing (SCC), 2015 IEEE International Conference on*, pp. 285-292, 2015.
- [2] F. El-Amrawy i M. I. Nounou, „Are currently available wearable devices for activity tracking and heart rate monitoring accurate, precise, and medically beneficial?,” *Healthcare informatics research*, pp. 315-320, 2015.
- [3] A. Pantelopoulous i N. G. Bourbakis, „A survey on wearable sensor-based systems for health monitoring and prognosis,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*40, tom 40, nr 1, pp. 1-12, 2010.
- [4] P. Kakria, N. Tripathi i P. Kitipawang, „A real-time health monitoring system for remote cardiac patients using smartphone and wearable sensors,” *International journal of telemedicine and applications*, 2015.
- [5] M. M. Baig i H. Gholamhosseini, „Smart health monitoring systems: an overview of design and modeling,” *Journal of Medical Systems*, Auckland, 2013.
- [6] M. Chan, D. Estéve, J. Fourniols, C. Escriba i E. Campo, „Smart wearable systems: Current status and future challenges,” *Artif. Intell. Med*, 2012.
- [7] V. Chan, P. Ray i N. Parameswaran, „Mobile e-Health monitoring: an agent-based approach,” *IET Commun*, 2008.
- [8] „Android ORM benchmark GitHub repository,” [Online]. Available: <https://github.com/touchlab/android-orm-benchmark-updated>.
- [9] „Android Developers - Distribution Dashboard,” [Online]. Available: <https://developer.android.com/about/dashboards/>. [access: 10 02 2018].
- [10] „GATT Specifications,” [Online]. Available: <https://www.bluetooth.com/specifications/gatt>. [access: 25 02 2018].
- [11] „Apache Cassandra,” [Online]. Available: <http://cassandra.apache.org/>. [access: 16 06 2018].
- [12] E. Hewitt i J. Carpenter, „Cassandra: The Definitive Guide, Distributed Data at Web Scale,” O'Reilly Media, 2016.
- [13] H. C., „Rdbms vs nosql: Performance and scaling comparison,” University of Edinburgh, Edinburgh, 2010.
- [14] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira i J. Bernardino, „Choosing the right NoSQL database for the job: a quality attribute evaluation,” *Journal of Big Data*, 2015.
- [15] D. H. Douglas i T. K. Peucker, „Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature,” *The Canadian Cartographer*, 1973.
- [16] „Spring Data Access,” [Online]. Available: <https://docs.spring.io/spring/docs/5.0.7.RELEASE/spring-framework-reference/data-access.html#spring-data-tier>.
- [17] „JWT Handbook,” [Online]. Available: <https://auth0.com/resources/ebooks/jwt-handbook>.
- [18] A. G. Mariusz Chmielewski, „Semantic Battlespace Data Mapping Using Tactical Symbology,” *Advances in Intelligent Information and Database Systems*, nr DOI: 10.1007/978-3-642-12090-9_14, pp. 157-168, 2010.
- [19] „Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 1st quarter 2018,” Statista, 2018.
- [20] „Microsoft Store - Microsoft Band 2,” [Online]. Available: http://microsoftstore.com/store/msusa/en_US/cat/All-Microsoft-Band.
- [21] R. Yonglin, R. W. N. Pazzi i A. Boukerche, „Monitoring patients via a secure and mobile healthcare system,” *IEEE Wireless Communications* 17(1).
- [22] P. S. Pandian, K. Mohanavelu, K. P. Safeer i V. C. PadakiV., „Smart Vest: Wearable multi-parameter remote physiological monitoring system,” *Med. Eng. Phys.*, 2008.