An Open Source Software Architecture and Ready-To-Use Components for Health IoT

Paulo Barbosa, Alex Figueiredo and Sabrina Souto
Center for Strategic Health Technologies – NUTES
Campina Grande, Brazil
{paulo.barbosa, alex.figueiredo, sabrina.souto}@nutes.uepb.edu.br

Eugenio Gaeta
Life Supporting Technologies - LifeSTech
Universidad Politécnica de Madrid
Madrid, Spain
eugenio.gaeta@lst.tfo.upm.es

Eriko Araujo
Atlantico Institute
Fortaleza, Brazil
eriko@atlantico.com.br

Tiago Teixeira
Unparallel Innovation, Lda
Lisbon, Portugal
tiago.teixeira@unparallel.pt

***Abstract*— This paper presents the definition of a software architecture and implementation as open-source components specially designed for the state-of-art of Health IoT technologies. These activities were conducted through a H2020 project that developed a smart IoT solution for childhood obesity and two contracted Health IoT projects for the leading brazilian clinics in wound treatment. The software components and the guidelines presented in this paper are fully available as github repositories and contributors can find immediate application in Health IoT projects. A survey with contributors in four different countries was conducted about the way the realized software architecture addresses quality requirements for Health IoT. The results shows that the integration of components was the main concern of this team due to the distributed nature of the projects and the implementation of patterns such as API Gateway using open source technologies are very likely to be reused in the future. The initiative continues collecting user's experiences for continuous improvements and research goals.**

Keywords—Health IoT, Software Architecture, Open Source

I. INTRODUCTION

The identification and definition of a software architecture is a complex process and involves negotiation to abstract needs and technologies. Reference architectures provide guidelines to standardize such process. In the Health IoT [1] field, some recent initiatives that arose as interesting candidates to provide this support are: WSO2 [2], IIRA [3] and IoT-A [4].

We focus on the following research questions in the Health IoT scenario:

RQ1: *How to efficiently address quality attributes when defining a software architecture for Health IoT?*

RQ2: *Considering the scope of the presented projects in this paper, what are the main expectations and contributions in terms of quality attributes of collaborators when providing open source architectural components for Health IoT?*

Addressing these questions we are contributing to deal with the current gap in the literature of showing how to realize reference architectures to actual development and software components moving forward from ideas and abstractions. Challenges such as integrating medical technologies is a major topic of research nowadays [5] defining microservices, integration of services and security assurance are exemplified by the design guidelines provided here in two scenarios of Health IoT[1] [2]. These scenarios provided open github realizations as the software architecture presented in this paper and opinions from contributors of the two scenarios referring how quality attributes were address are collected in a survey. The first scenario is the result of open source components developed for the H2020 OCARIoT project [3], and, although the solution is not totally available, most of the architectural components that supports the Health IoT infrastructure are available. The last one is the result of private projects for innovations supported by clinical partners in Brazil. Among them, we mention Cicatriza Clinics[4], the Brazilian leader in wound treatment. However, in order to manage the consistency of the definitions and examples shown in the next sections, we focus only on examples of the OCARIoT due to the more open nature of this project.

The chosen reference architecture from literature were [6] and [7] and the chosen paradigm is based on microservices and started from general guidelines on best practices, such as provided by Microsoft[5]. We had previously addressed some challenges of architectural decisions in this field [8] [9] and, in our opinion, several challenges in the area are still open.

The application scenarios for systems that this architecture fits are described in some of the project's reports[6]. We should mention: (i) taking measurements using smart connected health devices; (ii) monitoring physical activity and sleep patterns using wearables; (iii) measuring the impact of environmental variables; (iv) including social robots in IoT environments; (v) gamification and missions involving IoT data; and (vi) image processing. In this sense, we have a wide variety of scenarios and insertions of technology. For example, the experimented scenarios involve: (i) direct BLE connection of health devices and mobile apps; (ii) http connection of mobile apps with proprietary servers; (iii) wifi connection of health devices with proprietary servers; (iv) streaming processing of real-time environment data from sensors in cloud servers; among others.

Considering the experiences with these scenarios, we conducted a survey with thirty practitioners from all the mentioned projects, where we had twenty three respondents, to clearly understand what were the main expectations and best results in terms of quality attributes. In a nutshell, the survey results indicated the vast majority had integration concerns and the best evaluated components improved this

[1] github.com/ocariot
[2] gitbub.com/haniot
[3] ocariot.com
[4] cicatrizacg.com.br
[5] https://docs.microsoft.com/en-us/azure/architecture/microservices/
[6] ocariot.com/public-deliverables

2372-9198/20/$31.00 ©2020 IEEE
DOI 10.1109/CBMS49503.2020.00077

quality attribute in the software architecture following the microservices paradigm.

II. Open Source Software Architecture for Health IoT

Figure 1 introduces the overall idea of the defined software architecture for Health IoT. It is inspired on the best practices of the references mentioned in the introductory section. Taking the benefits of the microservices paradigm, microservices can be implemented in different languages, using different databases. In the case of the OCARIoT platform, we have available components developed in *Nodejs*, *Python* and *Java* for the direct integration of a messaging channel bus for efficient management of microservices. Concerning persistent data in microservices, we also provided libraries for encryption at rest for microservices that use MongoDB and MySQL databases, meeting important requirements from GDPR[7] in Europe. It was also defined a single entry point for microservices using Express Gateway[8], an open source API Gateway built on Express.js. Also we have a efforts toward high-scale and high-availability when adopting RabbitMQ[9], one of the most popular open source message brokers. Finally, the overall privacy and security management of the OCARIoT platform was developed relying in the open source tools Vault[10] and Consul[11].

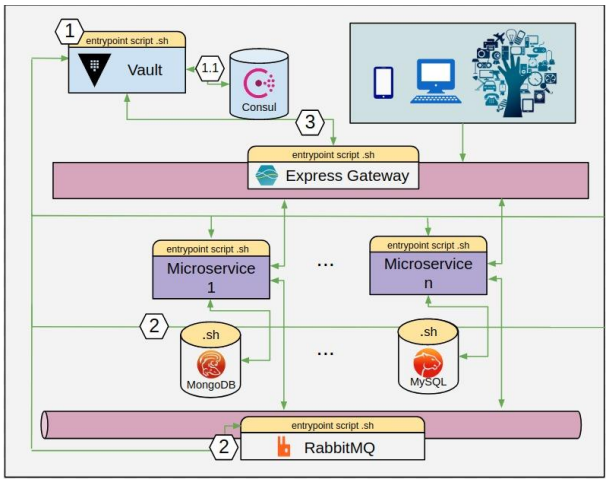

Figure 1. Overall microservices architecture.

The main identified components are better explained in TABLE I. The naming of these confuses with the name of the open-source technology was intentional, emphasizing the main contribution of this paper towards actual instances and ready-to-use components for a software architecture.

TABLE I. Architectural Components

Component	Description
Apps,Dashboards, IoT gateways	Health IoT tools for aggregating data and synchronization services. The data can be synchronized directly or can be requested by a third-party service.
Vault	Offers high level management policies to secure, store and control access ensuring security.
Consul	Extends the Vault functions creating high availability to manage keys, certificates, among others.
Express Gateway	Single entry point for interaction of clients and microservices. Realization of the API Gateway and Aggregator patterns, and used to manage the service call between the external components and the microservices.
RabbitMQ Bus	Implementation of a messaging channel, a mechanism used to enable asynchronous communication between microservices.

The initialization follows the numbers presented in Figure 1 and defines the flow information that will be presented in the next sections. In (1) Vault generates all the encryption keys and the root token. In (1.1) Consul is initialized as the back-end of Vault, providing the storage of encrypted data at rest and other secrets are generated, *e.g.* databases, messaging channel, etc. In (2), all the databases of the microservices and the messaging bus are initialized. Finally, in (3) we have initialization of the microservices. In the next section, we provide deeper insights to the reader through the instantiation in the OCARIoT Project.

III. The Software Architecture of the OCARIOT Project

OCARIoT (Smart childhood Obesity CARing solution using IoT Potential) is a platform that assess the efficacy of an IoT-based personalized coaching solution guiding children (9-12 years) to adopt healthy eating and physical activity behavior. The platform captures and manages data from localization, wearables, IoT sensors, personal health devices, robots and social networking combined with a set of challenge games, with the objective of triggering a long term behavioral change towards healthy habits in the children.

A. Health IoT Scenarios

In order to present the main features of the software architecture, we present here two Health IoT Scenarios of the OCARIoT platform. More scenarios can be found in the OCARIoT public deliverables[12].

Figure 2. IoT scenario monitoring rest activities and sleep patterns.

Consider the scenario described in Figure 2. It is about continuous monitoring through wearables and intervention through the missions in the OCARIoT app. Children are monitored by means their wearable devices that are able to detect physical activity and sleep data. The OCARIoT app

[7] https://gdpr-info.eu/
[8] express-gateway.io
[9] rabbitmq.com
[10] vaultproject.io
[11] consul.io
[12] ocariot.com/public-deliverables

proposes to children a set of missions (provided by a Decision Support System) and personalize in agreement to the identified child profile. On the other hand, health professionals are able to do a follow-up though the dashboard and also to receive tips and recommendations related to missions and the children's evolution.

Consider the scenario described in Figure 3. It is about measuring the weight and the Body Mass Index (BMI). The weight can be obtained using a commercial weight scale. The BMI will be calculated directly by the OCARIoT system using the weight measured and the height and age of the child. Of course, in the case of smart weight scales, other parameters could be integrated (*e.g.* body composition).

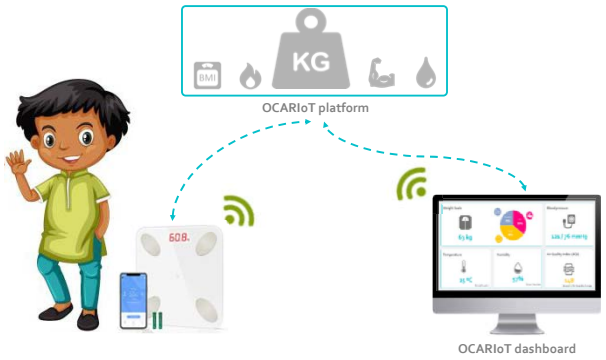

Figure 3. IoT Scenario usage of Personal Health Devices.

B. Instantiation

The software architecture of the OCARIoT platform realizes the proposed microservices architecture of Figure 1. Considering Figure 4, we explore the main architectural decisions and show the ways we took benefits from the open source components using the previous IoT scenarios.

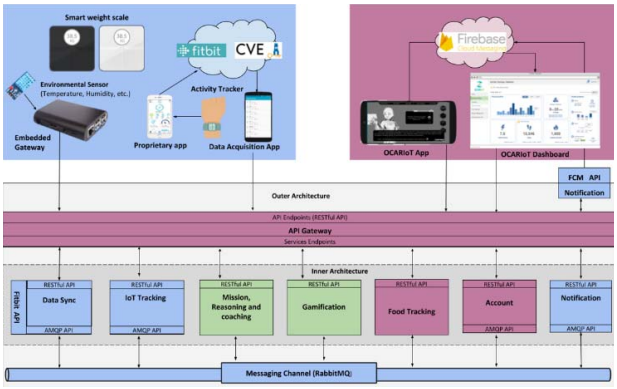

Figure 4. Microservices architecture of the OCARIoT project.

The main components in this instantiation are:

1. Wearables: commercial devices (*e.g.* Fitbit) are used during the pilots and the access to the proprietary server is allowed by vendors.
2. Embedded Gateway: smart sensors collect important environmental information at schools and updates in the platform for educational purposes. The embedded gateway provides the integration of sensors with the platform.
3. Data Acquisition App: mobile phone app manages the collection of data from different devices during the pilots and show key information for the health professionals. Available in github.com/ocariot/da-app.
4. API Gateway: realized using Express Gateway as suggested in the software architecture. Available in github.com/ocariot/api-gateway.
5. Messaging Channel: realized using RabbitMQ as suggested in the software architecture. Available in github.com/ocariot/rabbitmq-client-node.

Seven microservices were defined. Some are not available due to proprietary clauses. Some available are:

1. Account: user management and authentication. Available in github.com/ocariot/account.
2. IoT Tracking: tracking activities, sleep, environmental data and measurements. Available in github.com/ocariot/iot-tracking.
3. Data Sync: data synchronization from wearables platforms, allowing token management, revocation and to publish synchronized data in the messaging channel. Available in https://github.com/ocariot/data-sync-agent.

C. Instantiating the IoT Scenarios

Figure 5 explores the scenario introduced in Figure 2 showing the flow according to what was defined in Figure 4. Based on the collected data the children profiles are updated and custom interventions are triggered through the missions in the OCARIoT App. The architectural components involved are the Data Acquisition App, the API Gateway, the Account microservice, the Data Sync microservice, the IoT Tracking microservice, the Messaging Channel, the Mission, Reasoning and Coaching microservice, the Notification microservice, the OCARIoT App and the OCARIoT Dashboard.

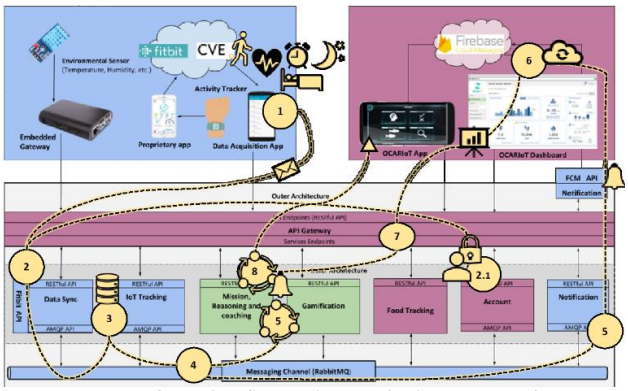

Figure 5. Information flow and events in the IoT scenario monitoring rest activities and sleep patterns.

Data flow is based on the following steps as in Figure 5:

1. The Data Acquisition App obtain the child's permission to collect your data from the Fitbit platform. When Fitbit authorization is granted to the Data Acquisition App, the Fitbit API provides an access token and a refresh token, so the Data Acquisition App performs the first synchronization by taking the child's data from the Fitbit server and

synchronizing with the OCARIoT API. After that, the next synchronizations are done in the background by the Data Sync microservice.

2. Data from the Data Acquisition App comes to the API Gateway that routes the data to the IoT Tracking microservice after the authentication process provided by the Account microservice in the step 2.1 in Figure 5.
3. Data is collected by the IoT Tracking microservice that notifies the Messaging Channel.
4. In this step the Messaging Channel generates an event that a new data set is available.
5. During this step the Reasoning and Coaching microservice elaborates the new data in order to update the missions and models while the Notification microservice sends a notification to Firebase.
6. Firebase then sends a notification to the OCARIoT App in order to inform that new missions are available or its progress is changed.
7. The OCARIoT App connects with the API Gateway for authentication (it follows the same step 2.1 in Figure 5).
8. New missions are provided and/or updated in the OCARIoT App.

Figure 6 explores the scenario introduced in Figure 3 showing the flow according to what was defined in Figure 4. During baseline data collection of the project pilots, smart weight scale and a mobile app were used in order to measure in an easy and seamless way weight and body composition of the children. The measures were automatically updated to the OCARIoT platform and accessible for review and data analysis. The architectural components involved are the Data Acquisition App, the API Gateway, the Account microservice, the IoT Tracking microservice, the Messaging Channel, the Mission, Reasoning and Coaching microservice, the Notification microservice and the OCARIoT Dashboard.

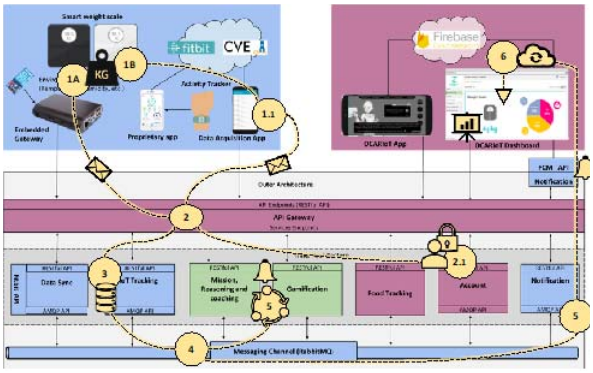

Figure 6. Information flow and events in the IoT scenario monitoring rest activities and sleep patterns.

Data flow is based on the following steps as in Figure 6:

1. The smart weight scale measures the child's weight
   1.1. in the case the weight scale does not connect directly with the OCARIoT infrastructure the data is catch by the Data Acquisition App that sends the data to the API Gateway;
2. Data comes to the API Gateway that routes it to the IoT Tracking microservice after the authentication process provided by the Account microservice in the step (2.1) of Figure 6;
3. Data is collected by the IoT Tracking Microservice that notifies the message channel
4. In this step the Message Channel generates an event informing that a new data is available. In this scenario, the Reasoning, Coaching and Notification microservice is listening for the data;
5. During this step the Mission, Reasoning and Coaching microservice elaborates the new data in order to update the missions and models while the Notification microservice sends a notification to Firebase;
6. Finally, Firebase sends a notification to the dashboard that updates the OCARIoT Dashboard in real-time.

IV. Survey and Analysis of the Software Architecture

In this section we present a survey that aims at understanding the clear role and expectations the software architecture for Health IoT plays in the contributor's point of view.

The conducted research was qualitative, exploratory and descriptive because it is a survey that intends to define respondent characteristics, measure data trends and validate existing conditions concerning the specification and the realization of the software architecture for Health IoT. The research is supported by the ethics and ethical frameworks established in these projects. We had twenty three respondents that were software architects and developers directly involved in the projects mentioned in this paper. These developers come from four different countries (Brazil, Greece, Portugal and Spain) and seven different cities (Fortaleza, Campinas, Campina Grande, Lisbon, Madrid, Derio and Thessaloniki).

We elaborated four very condensed and simplified questions about the software architecture focusing on the individual contribution and the quality attributes expectations, impact and likelihood of adoption in future projects. The idea was to extract only the necessary information, not overloading the technical staff. The questions were applied as follows:

Q1. What was your main contribution for the definition of the software architecture for Health IoT?

Q2. What was your main expectation during the definition of the software architecture for Health IoT?

Q3. According to your view, what is the component with more impact on your quality assessment of the software architecture?

Q4. The software architecture has the proposal of being open-source and support other Health IoT projects. Which component is the most likely to be reused by you in a future project?

A. Respondent Profiles

We contacted thirty individuals who were directly involved developing the projects. We got answers from twenty three of these individuals. In our opinion, these actors were able to evaluate as the main stakeholders of the software architecture because they were directly involved in technical

activities in which the quality attributes of the software architecture has a direct impact. The first question, *Q1*, which results are presented in Figure 7, addressed this point. There we had almost one third involved in the specification/realization activities, almost one third involved in the development of services directly integrated to the software architecture and almost one third developed user/technical requirements and tests/validation. Confirming the qualification of our respondents to address the benefits and drawbacks of the software architecture, no one was involved only in non-technical activities of promoting the benefits.

ANSWER CHOICES	RESPONSES	
I worked on the specification/realization	34.78%	8
I developed a service that is integrated	34.78%	8
I developed tests/validation for components	21.74%	5
I developed user/technical requirements	8.70%	2
I promoted the benefits	0.00%	0
TOTAL		23

Figure 7. Respondent's profiles.

B. Surverying the Quality Assessment

The second question, *Q2*, asked about the main expectations of the technical staff during the definition of the software architecture. According to Figure 8, the vast majority expected improvement in the integration with other services. This is an interesting result, due to the nature of these projects that involves a very distributed development environment. Improvements in the communication also took the second place, a result that we assign the first motivation of the first place. Finally, the second place was also occupied by security, which is perhaps the major concern in Health IoT projects nowadays.

ANSWER CHOICES	RESPONSES	
Improving integration with other services	78.26%	18
Improving security	8.70%	2
Improving communication with stakeholders	8.70%	2
Improving performance	4.35%	1
Documenting the system for future needs	0.00%	0
TOTAL		23

Figure 8. Expectations about the software architecture benefits.

The third question, *Q3*, asked about which component would have more impact on the quality assessment. According to Figure 9, the API Gateway had almost half of the votes. This is the main component that provides the first entry point for the integration between microservices and also improves the security level of the solution. In our analysis this seems to be the main justification. The Messaging Channel, in the second place, is also the complementary component to perform the integration of microservices, although it also reflects performance. No one thought that a specific microservice could have more impact on the quality assessment and three respondents, which were not involved in the implementation of microservices, said they do not have enough technical information to answer.

ANSWER CHOICES	RESPONSES	
API Gateway	47.83%	11
Messaging Channel	26.09%	6
Vault/Consul security solution	13.04%	3
I do not have enough technical information to answer	13.04%	3
A specific microservice you have in mind	0.00%	0
TOTAL		23

Figure 9. Opinion about component with most impact on quality.

Finally, the fourth question, *Q4*, asked to choose the component most likely to be reused. According to Figure 10, differently from the previous answers, in this case we had a very balanced set of answers. API Gateway was still in the first place, but tied with a specific microservice. This means that the likelihood of reuse of microservices is very high in this architecture, although no participant have recognized a specific microservice as causing more impact on the quality assessment of the software architecture. The Messaging Channel was also very close to first places, and, in our opinion, this means that this component has a very important role for new projects involving the communication of microservices. We still had respondents saying they have no technical information to answer, but this number was lower if compared to *Q3* as we assign this reduction to the reuse of microservices.

ANSWER CHOICES	RESPONSES	
API Gateway	30.43%	7
A specific microservice you have in mind	30.43%	7
Messaging Channel	26.09%	6
I do not have enough technical information to answer	8.70%	2
Vault/Consul security solution	4.35%	1
TOTAL		23

Figure 10. Opinion about likelihood of reuse.

C. Summary of the Survey Findings and Issues

In this section we present the main findings and issues in the survey of this paper:

1. Integration of services and components is one of the main concerns when working in Health IoT projects in large distributed environments. It overcomes security that is one main topics in the media and threats reported.
2. The API Gateway, that realizes the Aggregator Design Pattern, really adds value to the perception of quality of the software architecture. The API Gateway leaded the questionnaires about the components and other questions pointed to the integration of services. So we assumed the API Gateway could be seen as a popular solution for this issue.
3. In terms of efforts, the Vault/Consul security solution was the most demanding. However, it was not recognized by the respondents even all the Health IoT scenarios being in highly regulated privacy/security scenarios because of GDPR in Europe and Brazil's security laws. This perception will be discussed with project partners.
4. Most of the non-respondents of the questionnaires (seven in thirty) were not the most directly involved

with the technical activities that drove the definition of the architecture. Probably this was why we had no answers in the respondents profile stating that just promoted the benefits of the software architecture.

5. The main threat to validation was the involvement of respondents with the projects. To mitigate this situation, the first author, who conducted the survey, was self-excluded from the respondents set and managed to avoid any contact of all the other members of the project with this paper and the research questions presented here before the application of the survey and the conclusion of the survey section. In this sense, no other member had any idea about actual purpose of the survey and the possible findings. In our opinion, the answers were as much spontaneous as they could be.

V. CONCLUSIONS AND FUTURE WORKS

This paper presented the initial results of the definition of a software architecture oriented to Health IoT and available for reuse through open source components. We took some initial guidelines of reference architectures and defined our own implementation and quality assessment. We do not dare to call it as a reference architecture because it lacks formalization, but one of the main future works will be to generate efforts towards this realization. Other scenarios are available and the complete information can be consulted in the OCARIoT public deliverables website and in the *github* repository, as suggested in the introductory section.

The chosen open source technologies were the best and mature as we could find. The choices came from results after several meetings in the projects and always involved all the stakeholders. We have good expectations about the long term plans for this software architecture and future works will appear involving more experimental scenarios.

Finally, we understand that this work brought important contributions for answering the research questions presented initially. For *RQ1*, the quality attributes were addressed efficiently through open source technologies and realization of good practices from reference architectures. The high knowledge of respondents not answering that *do not have enough technical information to answer* is another evidence that the integration quality attribute was achieved according to contributor's mind. For *RQ2,* the questionnaires addressed directly its theme, capturing the main expectations of collaborators of quality attributes and what software component best realized their expectations. Quality attributes continue being investigated in this software architecture, and a set of technical experiments are currently being executed addressing verification of items like performance, security, fault tolerance, availability, among others. These experiments are being driven by the SQuaRE [10] family of standards and soon the results will be published for the scientific community.

ACKNOWLEDGMENT

This research has been performed under the OCARIoT project, funded from the European Union`s HORIZON 2020 Programme (2014-2020), ID 777082, and from the Brazilian Ministry of Science, Technology and Innovation through Rede Nacional de Ensino e Pesquisa (RNP), ID 003008. The authors acknowledge the FAPESQ/PRONEX (the State Funding Agency of Paraíba/Brazil) and CNPq-Brazil (Brazilian National Council for Scientific and Technological Development) for financial support. We are also thankful to developers who also contributed to the realization of the architecture, namely: Jefferson Medeiros, Lucas Rocha, Douglas Santos, Aislan Monteiro, Caio Lucena and Lucas Barbosa.

REFERENCES

[1] A. Martins, D. Santos, A. Perkusich e H. Almeida, "IEEE 11073 and connected health: Preparing personal health devices for the Internet," em *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, 2014.

[2] F. Estrada e R. Lazaro, WSO2 Developer's Guide: SOA and Data Services with WSO2 Enterprise Integrator, Packt Publishing, 2017.

[3] Industrial Internet Consortium, "Industrial Internet Reference Architecture V 1.9," 2020. [Online]. Available: https://www.iiconsortium.org/IIRA.htm.

[4] V. I. +. T. GMBH, "IoT-A Project," 2020. [Online]. Available: https://cordis.europa.eu/project/id/257521.

[5] P. Maranhao, G. Bacelar-Silva, D. Gonçalves-Ferreira, C. Calhau, P. Vieira-Marques, M. Alvarenga e R. Cruz-Correia, "OpenEHR Modeling Applied to Eating Disorders in Clinical Practice: OpenEHR-Archetypes in Eating Disorders," em *Proceedings - 31st IEEE International Symposium on Computer-Based Medical Systems, CBMS 2018*, Karlstad, Sweden, 2018.

[6] J. Bogner e A. Zimmermann, "Towards Integrating Microservices with Adaptable Enterprise Architecture," *IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW),* 2016.

[7] G. Olliffe, "Assessing Microservice Architecture for Scalable Application Delivery," Gartner Research, 2018.

[8] P. Barbosa, J. Queiroz, A. Figueiredo, D. Santos, F. Leite e K. Galdino, "RE4CH: Requirements Engineering for Connected Health," em *Proceedings of the 31st IEEE CBMS International Symposium on Computer-Based Medical Systems*, Karlstad, Sweden, 2018.

[9] P. Barbosa, F. Leite, D. Santos, A. Figueiredo e K. Galdino, "Introducing Traceability Information Models in Connected Health Projects," em *Proceedings of the 2018 IEEE 31st International Symposium on Computer-Based Medical Systems*, Karlstad, Sweden, 2018.

[10] "ISO/IEC 25000:2014 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE".