CLOUD 4ELE

Internet of Things Architecture for Enhanced Living Environments

Stylianos Balampanis, Stelios Sotiriadis, and Euripides G.M. Petrakis
Technical University of Crete

Sensors in enhanced living environments improve quality of life but present efficiency challenges. To address this, the authors propose an Internet of Things architecture based on modular cloud services.

Cloud computing has gained significant attention in recent years, with many companies recognizing its advantages and potential impact on people's lives. Today, the cloud has evolved into an important technology for application developers and users by allowing on-demand and remote resource access. The cloud allows efficient real-time data collection and analysis by offering a comprehensive view of resources, remote data management, easy access, and economic benefits.[1] Over the years, modular services—also referred as future Internet[2] and IBM microservices[3]—have been widely used as a component of large, complex applications to make them easier to configure, monitor, and update. Such services are available through different cloud platform providers, including IBM,[4] Amazon EC2 (https://aws.amazon.com/ec2), and Fiware (https://account.lab.fiware.org). Here, we focused on Fiware, which provides cloud services to build novel future Internet applications that use generic services, known also as generic enablers. Fiware offers open specification for services that could be used across different geographically locations and hosted in various Fiware Lab nodes available over the Internet.[5] These services use a service-oriented architecture (SOA) that allows communication based on Representational State Transfer (REST).[6]

In parallel with these developments, the Internet of Things (IoT) has emerged, with sensors embedded in everyday devices to facilitate automatic monitoring of data produced by humans or their environment.[7] Cloud computing and IoT together offer new opportunities for wide usage of this data, enabling the development of new applications that can impact our daily lives.[1,8,9] The development of applications using cloud resources becomes easier when we use scalable storage, which can increase capacity and performance by dynamically adding new storage nodes; the high bandwidth data transmission speed and real-time analysis makes it even more attractive.

Here, we propose a generic IoT architecture and present a motion-sensing cloud service to monitor patients' movement. The fundamental idea is that, by placing such sensors in enhanced living environments (ELEs), we can offer patients protection from accidents (such as falls) and let caregivers monitor patients remotely. In particular, the caregivers can monitor patients as well as create and monitor predefined movements for patients in rehabilitation. Our work was motivated by an existing motion sensor data collection system,[10,11] which collects data according to an event-based architecture that includes constant updates for patients who might need help.

To implement the service, we use the RESTFul architecture deployed on the open source OpenStack cloud system. Our system's advantages include services' reusability, improved fault tolerance, easy distribution of newer versions, and decoupling of services (and thus easy management). Our expectation is that decoupling the system components from the application logic will offer more flexibility; for example, integrating a new system will not require changes to the service's internal procedures.

Cloud Systems

Cloud computing systems include infrastructure and software that can be delivered in the form of remote services on a pay-as-you-go pricing model; these cloud systems have been defined as the next step of the Internet's evolution. Today, another promising technology is edge computing, which pushes clouds away from their logical network, creating *fog computing*.[12] Fog computing expands cloud functionality, allowing business logic and process management to be executed as near as possible to the actual data source (that is, the laptop or smartphone). This alternative view of clouds extends services to user premises and is utilized directly in users' personal devices. Fog computing could offer cloud technology know-how for remote data storage and management, while local data processing facilitates a self-adaptive environment for data extraction and analysis, such as in mobile devices. In such a solution, traditional legacy systems must be imported to the cloud infrastructure and interoperate in both local and remote clouds. To achieve this, users' software and APIs must communicate successfully and understand the new system constraints.

The SOA offers a paradigm to develop cloud-based software modules to meet these Internet client needs.[13] Using SOA, developers could achieve a high level of system granularity by supporting the exchange of information among services.[14] However, existing services generated from traditional systems are monolithic and difficult to interoperate with because they sometimes use heterogeneous APIs, hypervisors, and communication protocols. It therefore becomes essential to focus on integrating solutions that serve as interoperation strategies for allowing service communication—especially for services that have already been defined in business processes. This effort includes evolving SOA Web services, simplifying heterogeneous services so they can be more easily reused.

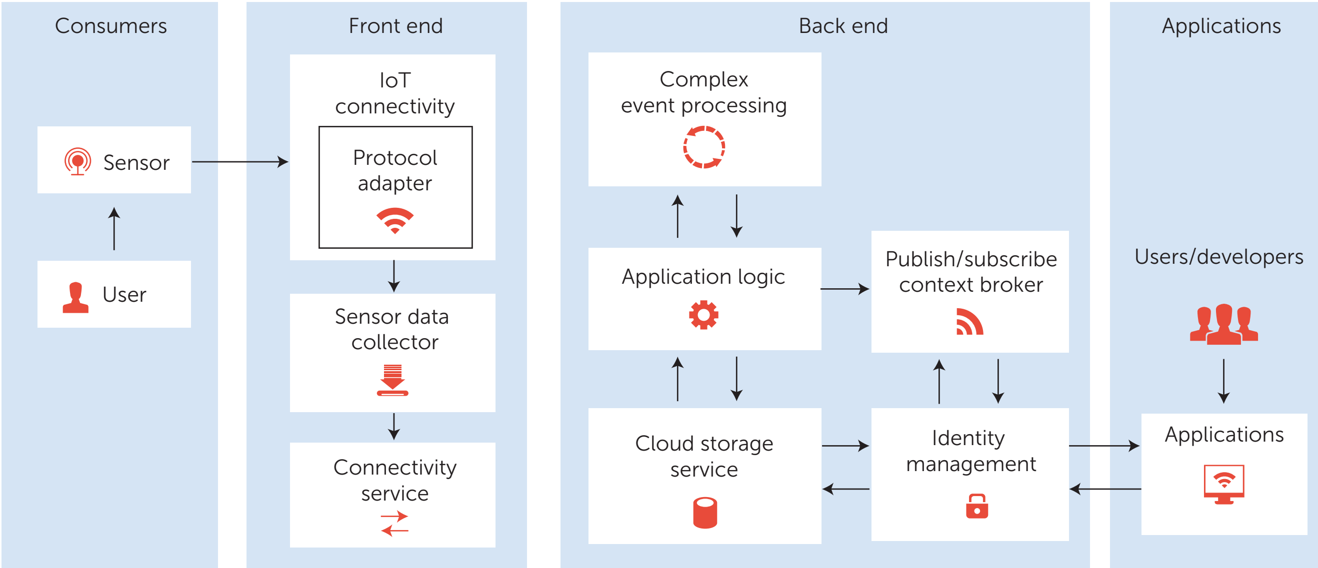

FIGURE 1. The reference architecture for a generic SOA system that includes data collection from IoT devices. The services are divided over four main domains: producers, front-end, back-end, and consumers.

Proposed Solution

In this article, we focus on the remote monitoring of two types of patients: those who are hospitalized and those who are in rehabilitation at home. Our proposed solution intends to facilitate the work of caregiving personnel by allowing remote monitoring, while improving the quality of life and daily life of patients. Continuous monitoring of an ELE—that is, the patient's home or hospital room—will offer significant advantages, such as enhancing patient security and helping staff members perform their tasks more efficiently. Also, it can reduce hospitalization costs as fewer staff members are required.

Our cloud monitoring system uses a motion sensor device (Microsoft Kinect; https://developer.microsoft.com/en-us/windows/kinect) that can be placed in the patient area and interpret patient movements. The solution has several advantages:

- It can increase profits for the ELE (that is, for hospitals or physicians) by minimizing the need to constantly monitor patients, thus serving more patients in an automated way.
- Patients feel safe, as the monitoring is continuous and real-time data are collected and evaluated by the system, which notifies physicians in case of emergency.
- Doctors can receive periodic updates on patients' progress and choose which patients and features require dynamic monitoring.
- It reduces costs for the hospital, personnel, and cloud infrastructure maintenance; it can also offer additional economic benefits, in that patients can be billed for the time periods in which the application is in use.

We expect that the proposed architecture will enhance personalization of care management based on the specific characteristics of patient profiles. It will provide a flexible architecture for analyzing various data from multiple sources and actors, and allow risk stratification for specific patients and their conditions. We further expect that it will facilitate comprehensive and improved therapy treatment coordinated by informal caregivers based in the home environment. Such systems can increase patients' autonomy and confidence in complying with therapy, improve self-management of their condition with the help of informal caregivers, and reduce patients' dependency on therapy. As a result, our system will reduce the need for patients to organize and attend face-to-face appointments with doctors and could reduce the amount of medication and the number of sick days.

Conceptual Model

The SOA-based conceptual model involves different cloud service providers that develop modules, each following its own development principles and tools (such as operating system, programming languages, and natural resources). Figure 1 shows a generic SOA for collecting IoT data and forwarding it to the cloud system.[10] The reference architecture represents a model of groups of services that are divided over four main domains: the producers, front-end, back-end, and consumers.

As the figure shows, the producers are sensor owners that generate data in intervals. The front-end is a gateway that acts as a mediator between the producers and the back-end for data exchange. The back-end system includes general services for user authentication, data context subscription, storage, event and system management, using standards, controls, and conditions to transfer information on individual services and orchestrate the service's business intelligence. Finally, the system consumers are either users or other applications that subscribe to the data. The architecture is based on software modules that operate on the cloud.

The architecture includes eight modules:

- *IoT connectivity/protocol adapter*. The IoT connectivity software module is responsible for connecting the sensor with the future Internet application components. It uses the protocol adapter to adapt the connection to the specific connectivity protocol (such as Bluetooth).[15]
- *Sensor data collector*. This module collects the sensor data and forwards it to the cloud. It also converts data into the desired form (that is, JavaScript Object Notation).
- *Connectivity service*. This module establishes a connection between the front- and back-ends, so data collected by the sensor data collector can be transferred to the application logic module for processing.
- *Complex event processing*. This module analyzes complex conditional events to aid decision making. It processes custom event patterns and then, based on specific user-defined conditions, decides the data's flow.
- *Cloud storage*. This module is responsible for storing and retrieving data from a database. Its main functionalities are offered as a REST API to make storing and retrieving data easy for developers and others stakeholders.
- *Application logic*. This module is application specific and encapsulates the business logic of the future Internet application as it handles and processes sensor data. It uses the complex event processing module for decision making and the cloud storage module for storing and retrieving sensor data; it then sends its results to the publish/subscribe broker.
- *Publish/subscribe context broker*. The publish/subscribe context broker receives the results of the application logic's sensor data processing and publishes them. The context broker's role is to publish context to subscribers.
- *Identity management*. This module handles user authentication and access authorization.

The complex event processing and publish/subscribe context broker modules are based on Fiware services (http://catalogue.fiware.org/enablers). The identity management module uses Fiware's KeyRock authentication service for application users and developers who access services through REST APIs.

Motion Sensors in ELE Using Cloud Computing

The service-centric architecture is based on the idea that any complex problem can be solved optimally and effectively if it's divided into smaller parts. Our architecture comprises a flexible set of design principles and services that communicate with each other and can be used in multiple systems from several business areas. Its advantages include reusable services, faster and more efficient debugging, quicker distribution of new products, and applications and services that aren't bound by the system, but can be modular. As discussed earlier, the proposed system involves information producers, including the sensors that produce data and users who interact with the producers and the user interface (front-end) where data collection occurs.

The system is implemented using Microsoft Kinect, which lets us determine the position and movements of users. Specifically, the data is provided as a set of points that comprise the human skeleton. This lets us record 20 joints of the human body (the wrist, knee, and so on) while the overall frame indicates the user's attitude and position. For each of our points, the coordinates are given in 3D form. In particular, the variable X represents the position or displacement of the user on the horizontal x-axis; Y indicates the user's position on the vertical y-axis; and Z represents the user's distance from the sensor.

System Description

As Figure 1 shows, the system includes three main sections: the user interface (front-end), system management (back end), and the users. The *user interface* includes the Microsoft Kinect sensor and the device that's connected to the Internet for collecting and decoding the sensor data. The interface allows data forwarding to the cloud in real time. A system administrator can insert and remove sensors from the system and save patient information.

The back-send *system management* section consists of general-purpose services for processing and storing data transported from the Kinect sensor to the cloud. More specifically, the services include Orion's Publish/Subscribe Context Broker generic enabler and JSON's Storage generic enabler, which include rules for managing user subscriptions and

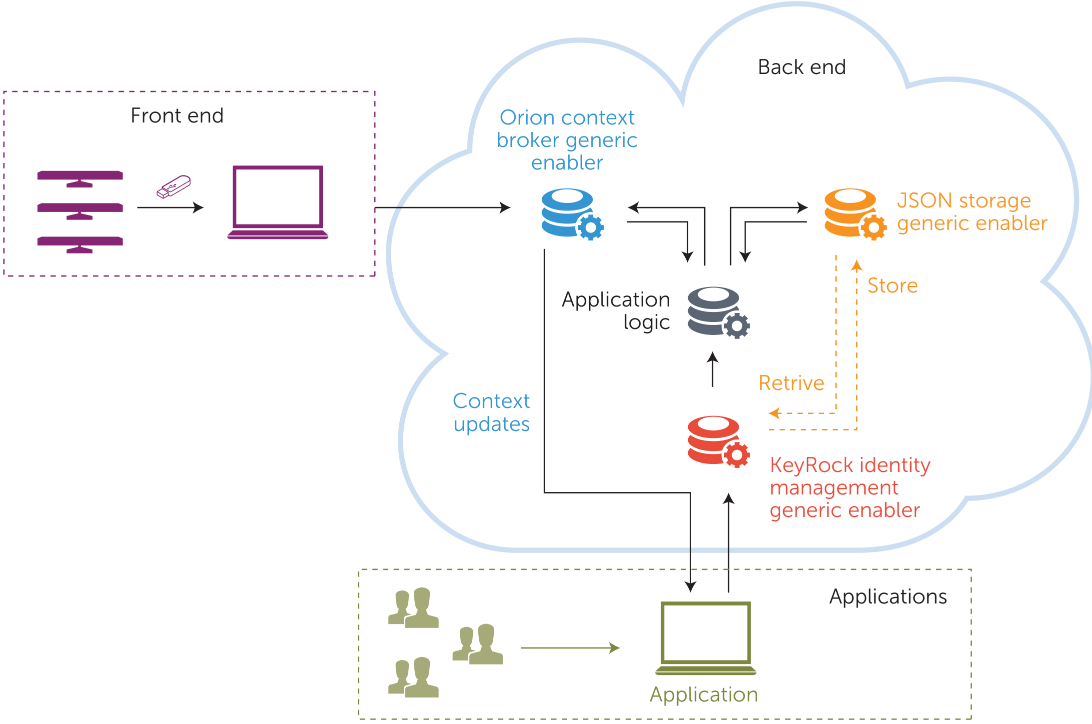

FIGURE 2. The propose system architecture. The service-oriented system uses the Microsoft Kinect sensor to collect data from Internet of Things devices in the enhanced living environment (ELE).

storing information and data, respectively. In addition, this section contains the authentication mechanism for the user entering the application—that is, the KeyRock Identity Management generic enabler. Finally, in the *user section*, medical personnel can use the system logic (application logic) to set conditions and rules of the result produced by the application.

Figure 2 shows the system architecture: the user interface allows sensor installation. System management is responsible for managing and processing data in the cloud, as well as for communication between modules and the application logic; and the users set conditions on the application.

Dataflow Analysis

The data imported from the sensor follow a flow path to the cloud according to a set of rules as follows:

- The Kinect sensor outputs data in two-second intervals. When the user isn't in the motion sensor's area, data isn't recorded.
- The sensor forwards the decoded information to the publish/subscribe context broker generic enabler, which updates with new data from the sensors.
- The JSON storage generic enabler allows storing data that are coming from the Kinect sensor.
- The users have access to the application with their personal details. The KeyRock service identity management generic enabler is responsible for user registration and access.
- The user in the application environment can request assistance from the context broker generic enabler or request his or her patient history and data collected from the JSON storage generic enabler.
- After each request for assistance, the context broker generic enabler service returns a unique identifier (subscribe ID) so the system recognizes the room being monitored.

These actions are orchestrated by the application logic module.

Use Cases

Our system highlights an IoT-based solution for the e-health sector to help medical personal perform their work more easily. It does this by taking advantage of modern technologies that enable remote patient monitoring, focusing on hospital and physiotherapy centers in the ELE area. We present two use cases—monitoring solutions for hospitalized patients and for rehabilitation patients suffering knee injuries.

Hospitalized Patient Scenario

This scenario applies the system motion sensor in a hospital ELE. Initially, medical personnel—that is, members of the nursing staff—have administration rights and place Kinect sensors in specific places in front of patient beds. They configure the patient profile for each sensor with basic information, including the patient's name and room number. The sensors provide continuous information as to whether the patient should be in the bed and whether they need help with basic tasks. Applying the motion sensor solution lets a few nurses and doctors monitor many patients while improving efficiency and the quality of the patient's care experience.

The implementation of this scenario includes installing the sensor and characterizing the body parts that produce the essential information being recorded. For example, we can place the motion sensor in front of the patient; the sensor then starts monitoring the patient's movements and notifies medical personnel accordingly. The Microsoft Kinect can record the frame of the human skeleton and track patient actions by recognizing 20 joints in the human body. The sensor placement point is decided based on the sensor configuration; Microsoft Kinect operates most accurately when it's at a distance greater than 1 meter away from the patient and less than 2 meters away. We therefore decided on the sensor's position in front of the patient's bed.

We monitor the patient's left and right shoulders to identify if he or she wants to get out of the bed. These two values are required for preforecasting the effort to get the patient from either the right or the left side of the bed. We set an upper limit threshold on these values; if the limit is reached, the application notifies the medical personnel to intervene directly. In the second case, where the patient asks for help, the system identifies the position of the wrist. In particular, in cases in which a patient needs a help they raise a hand. To do this, it compares the wrist position to the set limit; once that is exceeded, the alert appears as a request for assistance and a notification is submitted to the medical personnel.

Rehabilitation Scenario

The second scenario focuses on a user rehabilitating from a knee injury at home and while being monitored by a physiotherapy center. In this case, the system administrator sets up the system, activating the sensor system with a unique code and arranging the patient's furniture. The administrator is also responsible for providing the patient with exercises and instructions for doing them (that is, which movements to do and which thresholds are acceptable) and registers the values in the system. The patient can then use Microsoft Kinect to perform the exercises at home.

The doctor can monitor the rehabilitation recovery process remotely based on records of the patient's movement history and the time incurred. The efficient use of the system will make the transactional aspects of healthcare more productive by monitoring patient status, activity, and compliance with therapy. The proposed model is expected to provide improved therapy treatment coordinated by informal caregivers and based in the home environment.

As in the first scenario, we exploit the dynamic recording of the skeleton by tracking information for recognizing movements. We place the sensor at a distance of more than 1 meter away to receive the most accurate results. In this case, the selected values relate to the position of the injured ankle. Initially, the user must have the leg on the ground, and the exercise includes a check of the position translated to the leg height (which has been set by the physician). According to this position, the user is informed as to the maximum exercise height at which to move the injured leg.

Familiarization with the system technology is expected to increase patient autonomy and confidence in complying with the therapy, improve self-management of the target condition with the help of informal caregivers, and reduce the patient's dependency on therapy. We plan to design the system further to consider multichannel information on the specific patient's condition and thereby to encompass a holistic view of the patient's health status for formal and informal caregivers.

Our goal is to facilitate quality healthcare services while simultaneously helping to reduce the costs of healthcare, with patients spending less time in the hospital and yet continuing to generate detailed health data. We expect that this will let caregivers react more quickly to the medical emergencies of elders and let all patients better self-manage their own health and wellness in ELE.

An important part of the system proved to be the use of general-purpose services. As cloud technology advances, adequate space and appropriate tools will mean that more and more applications will be developed. Our proposed system supports future expansion and the addition of functionalities to meet people's daily needs. As future part of its development, we plan to dynamically add new motion sensors, including sensors for measurements such as heart rate and pulse to allow more sophisticated patient monitoring. We also aim to explore different aspects of the system performance related to network delays and the accuracy of sensor data collection with regard to high-bandwidth dataflows.

References

1. A. Castiglione et al., "On Secure Data Management in Health-Care Environment," *Proc. 7th Int'l Conf. Innovative Mobile and Internet Services in Ubiquitous Computing* (IMIS 13), 2013, pp. 666–671.
2. A. Botta et al., "On the Integration of Cloud Computing and Internet of Things," *Proc. Int'l Conf. Future Internet of Things and Cloud* (FiCloud), 2014, pp. 23–30.
3. V. Gucer and S. Narain, *Creating Applications in Bluemix Using the Microservices Approach*, IBM, 2015; www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp5271.html.
4. S. Daya et al., *Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach*, IBM, 2015; www.redbooks.ibm.com/abstracts/sg248275.html?Open.
5. K. Stravoskoufos et al., "IoT-A and FIWARE: Bridging the Barriers between the Cloud and IoT Systems Design and Implementation," *Proc. 6th Int'l Conf. Cloud Computing and Services Science* (CLOSER 2016), 2016, pp. 146–153.
6. M. Massé, *REST API Design Rulebook*, O'Reilly Media, 2012.
7. T. Lynch Koreshoff, T. Robertson, and T. Wah Leong, "Internet of Things: A Review of Literature and Products," *Proc. 25th Australian Computer-Human Interaction Conf.: Augmentation, Application, Innovation, Collaboration* (OzCHI 13), H. Shen et al., eds., pp. 335–344.
8. J. Gubbi et al., "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1645–1660.
9. A. Castiglione et al., "Cloud-Based Adaptive Compression and Secure Management Services for 3D Healthcare Data," *Future Generation Computer Systems*, vol. 43, issue C, Feb. 2015, pp. 120–134.
10. A. Preventis et al., "Interact: Gesture Recognition in the Cloud," *Proc. IEEE/ACM 7th Int'l Conf. Utility and Cloud Computing* (UCC 14), 2014, pp. 501–502.
11. A. Preventis et al., "Personalized Motion Sensor Driven Gesture Recognition in the FIWARE Cloud Platform," *Proc. 14th Int'l Symp. Parallel and Distributed Computing*, 2015, pp. 19–26.
12. *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are*, white paper, Cisco Systems, 2015; www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf.
13. J. Bih, "Service Oriented Architecture (SOA) a New Paradigm to Implement Dynamic E-Business Solutions," *Ubiquity*, Aug. 2006, article 4; http://ubiquity.acm.org/article.cfm?id=1159403.
14. S. Sotiriadis et al., "An Architecture for Designing Future Internet (FI) Applications in Sensitive Somains: Expressing the Software to Data Paradigm by Utilizing Hybrid Cloud Technology," *Proc. 13th IEEE Int'l Conf. BioInformatics and BioEng.* (BIBE 13), 2013; doi:10.1109/BIBE.2013.6701578.
15. *FIWARE Architecture Description IoT Gateway Device Management*, specification, Fraunhofer Institute for Open Communication Systems FOKUS, 2012; https://forge.FIWARE.org/plugins/mediawiki/wiki/FIWARE/index.php/FIWARE.ArchitectureDescription.IoT.Gateway.DeviceManagement.

STYLIANOS BALAMPANIS *is an undergraduate student in the School of Electronic and Computer Engineering at the Technical University of Crete and a member of the Intelligent Systems Laboratory. His research interests include cloud computing and Internet of Things as well as Fiware systems and modeling novel future Internet applications. Contact him at sbalampanis@gmail.com.*

STELIOS SOTIRIADIS *is a research fellow in the Department of Electrical and Computer Engineering at the University of Toronto and a research collaborator at the Technical University of Crete, where he's a member of the Intelligent Systems Laboratory. His research interests include clouds, Internet of Things, future Internet application development, interclouds and cloud federations, high-performance computing systems, and grids. Sotiriadis has a PhD in interclouds from the University of Derby, UK. Contact him at s.sotiriadis@utoronto.ca.*

EURIPIDES G.M. PETRAKIS *is a professor and laboratory director of the Intelligent Systems Laboratory, which is a unit of the School of Electronic and Computer Engineering at the Technical University of Crete. His research interests include clouds, Internet of Things, future Internet, semantic Web, medical information systems, and multimedia and Web information systems. Contact him at petrakis@intelligence.tuc.gr.*