Computers and Electronics in Agriculture 140 (2017) 255–265

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: A case study

Tomo Popović [a,*], Nedeljko Latinović [b], Ana Pešić [c], Žarko Zečević [d], Božo Krstajić [d], Slobodan Djukanović [d]

[a] Faculty of Information Systems and Technologies, University of Donja Gorica, Oktoih 1, 81000 Podgorica, Montenegro
[b] Biotechnical Faculty, University of Montenegro, Mihaila Lalića 1, 81000 Podgorica, Montenegro
[c] Institute of Marine Biology, University of Montenegro, P. Fah 69, 85330 Kotor, Montenegro
[d] Faculty of Electrical Engineering, University of Montenegro, Džordža Vašingtona bb, 81000 Podgorica, Montenegro

A R T I C L E I N F O

Article history:
Received 8 September 2016
Received in revised form 29 April 2017
Accepted 9 June 2017

Keywords:
Cloud computing
Ecological monitoring
Internet of things
Precision agriculture
Remote sensing
Software architecture

A B S T R A C T

This paper discusses a case study of designing a private Internet of Things (IoT) enabled platform for the research in precision agriculture and ecological monitoring domains. The system architecture is gradually derived using an approach of multiple, concurrent views. Each view represents an architectural perspective describing the solution from the viewpoint of different stakeholders, such as end-users, researchers, developers, and project managers. The end-user requirements have been identified using a set of high-level scenarios, which capture the context and illustrate the motivation for building the platform. The requirements and architecture of the proposed platform have been derived so that the users of the platform, researchers, and developers on the project, can utilize it for prototyping solutions for these high level use cases. The paper further describes the implementation of the platform and its evaluation using various sensor nodes deployed at the research and end-user facilities. The solution is open to further development with respect to supporting additional IoT protocols, data types, and interfacing to various analytics tools. The proposed architecture can also be implemented using different server platforms and cloud technologies.

1. Introduction

In the time of increasing demand for food, precision agriculture provides higher yields with a lower input cost and leads to a reduction in environmental pollution and labor (Shirish and Bhalerao, 2013). Modern day food production and precision agriculture are expected to dramatically increase the usage of the latest computer and electronic technologies (Cho, 2012; Zaks and Kucharik, 2011). In accordance to this, decision support systems have been developed in the last decades in order to provide expert knowledge needed for farmers in their agricultural management. In Rossi et al. (2014), the authors present a good example of such solution, which is designed to help grapevine farmers make the right decision on the proper time for pesticide treatment based on sustainable agriculture. It consists of a system for monitoring basic parameters in the vineyard such as air, soil, plants, pests, and diseases and software tools that analyze these data providing alerts and decision support information. In 2009, for example, the European Parliament and the Council established Directive 2009/128/EC to achieve the sustainable use of pesticides (Directive 2009/128/EC, 2009). According to this directive, integrated pest management is obligatory in all the European Union (EU) Member States by 2014. In order to achieve integrated pest management, modern information technology and decision support systems have an important role to decrease use of plant protection products and improve the quality and yield of crops (Vujović et al., 2016). For the purpose of public health and mariculture protection, monitoring water quality and biomonitoring in fishing sea are conducted regularly (Law on Marine Fisheries and Mariculture, 2009). This monitoring is at this moment partially aligned with EU regulations Directive 2006/113/EC, Regulations (EC) No 853/2004 and (EC) No 854/2004 and in the next period will be fully harmonized (Directive 2006/113/EC, 2006; Regulation (EC) No 853/2004, 2004; Regulation (EC) No 854/2004, 2004).

During past decade, monitoring marine environment has become one of the most important issues of marine investigations due to high pressure of human activities through industrial, tourist, and urban development in coastal areas. Physical and chemical seawater parameters represent the base for every investigation of marine environment, hydrographical and biological. Parameters such as temperature and salinity influence horizontal and vertical

* Corresponding author.
E-mail address: tomo.popovic@udg.edu.me (T. Popović).

http://dx.doi.org/10.1016/j.compag.2017.06.008
0168-1699/

distribution of water masses, density of water, and distribution of dissolved particles. Together with pH and oxygen, these parameters influence and determine the distribution and biology of all living beings in the seas and oceans, which has an important impact on human health as well. With its access to the coastal line, Montenegro is using its aquaculture capacities, including fish and mussel farms. Both aquaculture farming and expanding tourism require advanced marine environment monitoring and water quality assessment (Cater, 2008; Xu et al., 2014). In addition to weather data, which is important for both the agriculture and aquaculture, the water quality assessment employs smart buoys (Helmi et al., 2014; Nam, 2005), as well as systems for the use in inland waters (Papoutsa and Hadjimitsis, 2013). As a candidate country for the EU, Montenegro is evaluating and adopting the use of latest technology in the domains of precision agriculture and ecological monitoring.

In recent years, the development of information and communication technology (ICT) resulted in the emergence of two important concepts that affect the world around us: Internet-of-Things (IoT) and Cloud computing (Evans, 2011; Mell and Grance, 2011). Both concepts are expected to be put to use in agriculture on a much larger scale in the near future (Vermesan and Friess, 2013; Kaloxylos, 2012). The IoT is a network of physical objects (i.e. devices, vehicles, buildings) instrumented with embedded electronics, sensors, software, and networking connectivity enabling these objects to collect and exchange data (Ojha et al., 2015). The IoT equips objects of interest to be sensed and controlled remotely over existing and future network infrastructure, which creates various opportunities to integrate physical objects with computer-based systems. The main goals of IoT include improved efficiency, accuracy, economic gains, and better quality of life (Holler, 2014). Cloud computing is based on the utilization of computer resources (processors, memory, storage, network), which can be located and managed remotely. Cloud computing service models include infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). Clouds can be deployed as public, private, or hybrid (Mell and Grance, 2011). Our goal is to implement a private IoT cloud platform that can be used as a foundation for research and development in the domains of precision agriculture and ecological monitoring. The mission of the project is to create a research and development platform in the areas of sustainable agriculture, monitoring of the crops, forest and water ecosystems, development of techniques for controlling and reducing pollution, analysis and standardization of food products, control of land quality, and improvement of the public health (Fig. 1). The project is focusing on the utilization of IoT and Cloud to support the adoption of these novel technologies and innovations in the areas of precision agriculture and ecological monitoring.

Designing and implementing computer and electronics systems, especially IoT, in the domains of precision agriculture and ecological monitoring can be challenging, therefore a systematic approach is needed (Krčo et al., 2014; Kruize, 2016). Several IoT cloud platforms are available in the form of public cloud services (Azure IoT Suite, Amazon AWS IoT, DeviceHive and others), but main requirements in this project were to focus on private deployment and the use of open source software. Few open source IoT platforms have been evaluated (Kaa, FiWare, ThingSpeak). Among those, the ThingSpeak platform has been identified as the closest fit for the project needs (ThingSpeak IoT Platform). However, the open source variant of that solution has not been as active recently since it has been commercialized and offered as a public cloud service.

This paper describes a case study of architecting an IoT-enabled platform aimed at the use for research and development. The paper is organized as follows: after the Introduction section, Section 2 provides context and motivation by illustrating the end-user requirements. Section 3 describes derivation of the platform architecture and its implementation. Section 4 discusses the evaluation and practical usage of the platform. Finally, the conclusions and references are provided at the end.

2. Context and motivation: high-level use scenarios

The IoT platform should provide support for researchers and developers on the project working on prototype solutions for various scenarios from the research domains of precision agriculture, mariculture, and ecological monitoring. The platform should enable rapid creation of testbeds and prototypes of new analytic, modeling, and predictive functions. The following sections discuss the context and motivation using examples of high level use scenarios.

2.1. Precision agriculture

The list of precision agriculture use cases currently being developed is given in Table 1. These end-user applications include smart

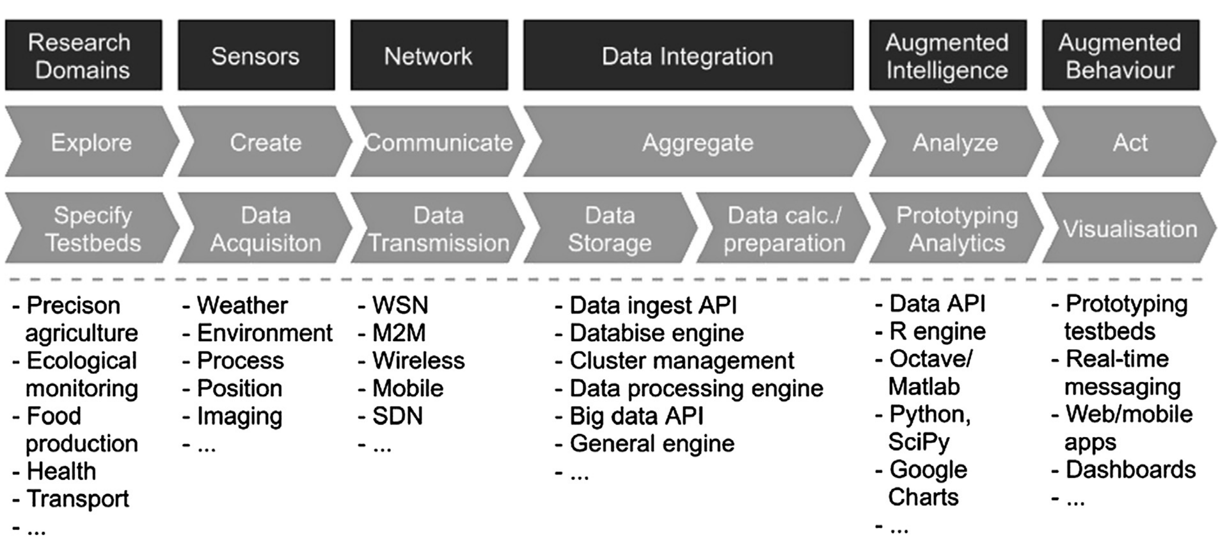

Fig. 1. Concept: an IoT-enabled research and development platform.

Table 1
Precision agriculture: high-level use case examples.

UC	Actor/Role	Use Case
1-1	Farmer	Smart irrigation
1-2	Farmer	Smart soil fertilization
1-3	Farmer	Smart spraying
1-4	Farmer	Diseases forecasting and detection

irrigation, smart soil fertilization, smart pest control (spraying), and plant disease forecasting and detection (Sekulić et al., 2016; Jhuria et al., 2013). For example, a smart spraying expert system for forecasting of grapevine downy mildew (*Plasmopara viticola*) is critical as the disease causes significant damages in vineyards in Montenegro each year (Fig. 2). The control of grapevine downy mildew represents the most expensive part in the total costs provided to protect vines from harmful organisms. Because of the consequences of the disease that in some years reflect in a total yield loss, as well as the rationalization of the costs, the expert system is being designed to help grapevine growers to determine the appropriate time of treatment with adequate fungicides.

Fig. 2. Severly damaged grapevine cluster due to the infection by *Plasmopara Viticola*.

For these use scenarios, the parameters of interest include temperature, pressure, relative humidity, soil humidity, leaf wetness, UV radiation, etc. The periodicity for the most use cases is identified to be at one hour, but for some, the data needs to be collected every 15 min. For research purposes, the periodicity may as low as 1 min. In order to prototype such applications, the platform shall provide means for quick configuration and setup of continuous data collection and interfacing to analytics tools. Some applications may need inclusion of digital media such as digital images from camera nodes and/or satellites (Montalvo, 2013).

In this project, the precision agriculture focus is on employing remote sensing and IoT, wireless communications, cloud computing, intelligent systems, and expertise in agriculture together in order to enable implementation of various intelligent applications. Fig. 3 illustrates a context view of the IoT deployment in an agricultural environment. Several IoT sensor nodes are installed throughout the field to collect, preprocess, and transmit the measurements of interest. The IoT nodes are communicating to the servers in Cloud directly or via gateway. Digital images may be captured by the camera nodes using filed cameras, drones, or even by accessing satellite imaging (Kale, 2015). The servers in the Cloud are used to host data integration, analytics, remote visualizations, smart applications development, and prototype deployment. Remote users can access the data and smart applications using their workstations and/or mobile devices. Field users can access the Cloud using their mobile devices.

2.2. Mariculture and ecological monitoring

Data on seawater temperature, salinity, pH and oxygen are collected in Montenegro through various national and international projects and national monitoring programs. For example, water quality and fishing sea biomonitoring is regularly conducted for the purpose of public health and mariculture protection at every shells and fish farm (Law on fishery and mariculture, Official Gazzete of Montenegro 56/09, 40/2001, 47/2015). Every 15 days in the period from April to October, as it is prescribed in a Law on water

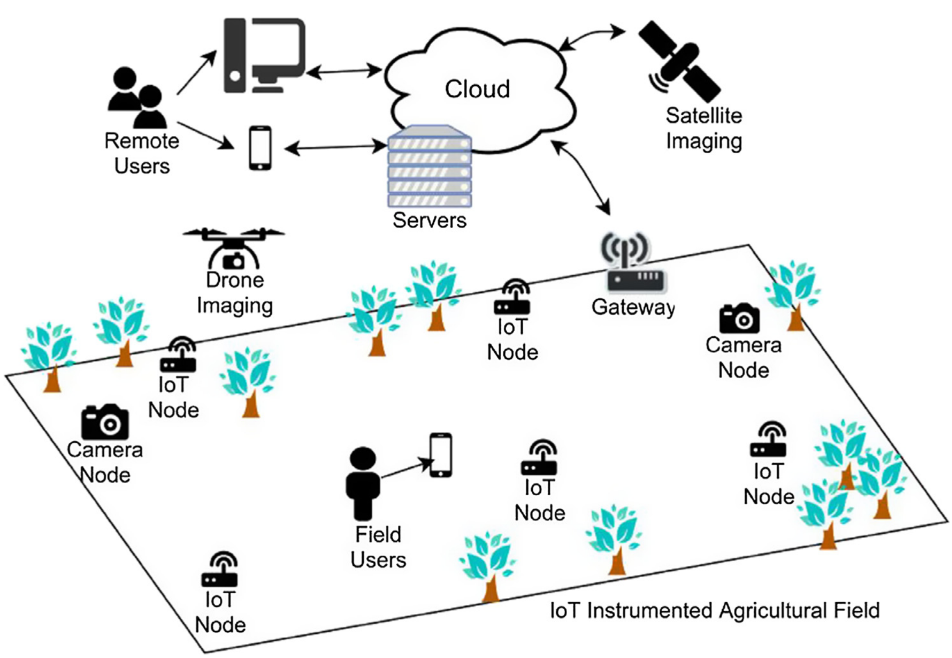

Fig. 3. Context view: precision agriculture.

Table 2
Mariculture and ecological monitoring: high-level use cases examples.

UC	Actor/Role	Use Case
2-1	Relevant ministry	Assessment of the state of marine environment
2-2	Farmer	Fish/ Mussel Farm Monitoring
2-3	Port clerk	Port Water Monitoring
2-4	Beach owners, Swimmer	Beach Water Conditions Assessment

(Official Gazette of Montenegro 27/07) and a Law on environmental protection (Official Gazette of Montenegro 48/08, 40/10, 40/11), continuous monitoring of public beaches in Montenegro is conducted (more than 100 beaches) in order to establish sanitary quality of water and to prevent any negative impact on human health.

High level use cases relevant to the ecological monitoring of the sea water within this project are listed in Table 2: assessment of the state of marine environment, fish/mussel farm monitoring, port water monitoring, beach water conditions assessment. The measurements of interest include air temperature, air pressure, air humidity, water salinity, water pH, dissolved oxygen, etc. The periodicity of these measurements is expected to be in the range of 1–30 min.

In this study, when speaking about ecological monitoring, an emphasis is put on the collection of weather data and water quality data of the sea water (Fig. 4). The weather data is expected to be taken at various locations and the sensors include air temperature, air pressure, humidity, wind speed, and others (Xu et al., 2014; Cater, 2008; Papoutsa and Hadjimitsis, 2013). The periodicity is expected to be at 15 min, but for some research purposes it may be as low as 15 s. Some use scenarios may require utilization of digital images (i.e. on-site cameras, drones, satellite).

3. IoT platform: deriving the architecture

In this case study, we have gradually derived the software architecture of an IoT-enabled platform for the use in precision agriculture and ecological monitoring research domains. The architecture is defined by using the approach of multiple, concurrent views. This approach has been widely adopted by the software development community and aligned with the latest international architecture standard ISO 42011 (derived from IEEE Std 1471) (Rozanski and Woods, 2011; ISO/IEC/IEEE, 2011). Each architectural view is used to describe the solution from the viewpoint of different stakeholders, such as end-users, researchers, developers, and project managers.

3.1. IoT platform-specific functional requirements

The proposed IoT platform allows the researchers on the project to implement prototypes of analytic, modeling, and predictive functions needed to create applications for the high level use scenarios described in the previous section. Researchers and developers want to use the proposed platform to quickly include and configure IoT nodes, establish sensor data collection, and implement prototypes of smart applications Fig. 5. These main functionalities of the platform can be seen as support use cases in which researchers utilize the IoT technology to achieve the goals of target applications (high-level use scenarios discussed in Section 2).

The platform shall provide the user with tools for adding, describing, and modifying various sensor nodes and their quick integration into the working system. Each sensor node is identified with its name, location, accessibility, position, and the description of measurements obtained by its sensors. The platform shall also provide for the calibration of the raw data when necessary. The data collection is a service-type functionality. The platform continuously waits for sensor data, and when new data arrives, utilizing the application programming interface (API) for writing, the sensor data is persisted into the data storage. Finally, the platform provides researchers and developers with tools to quickly setup and configure testbeds for their research experiments, as well as to prototype and implement end-user applications. The platform provides tools for visualization and real-time messaging, and the smart analytics functions can be implemented using the read API and API to third party analytics tools.

3.2. Conceptualizing the platform's architecture

The conceptual architecture of the implemented solution and the interaction of the system blocks is shown in Fig. 6. The data from the IoT nodes are transferred to the platform using standard communication protocols and API for data import. IoT nodes communicate to the platform directly or indirectly through a gateway. The database is used for persistence of sensor and node configuration meta data. The platform provides an API for the external data

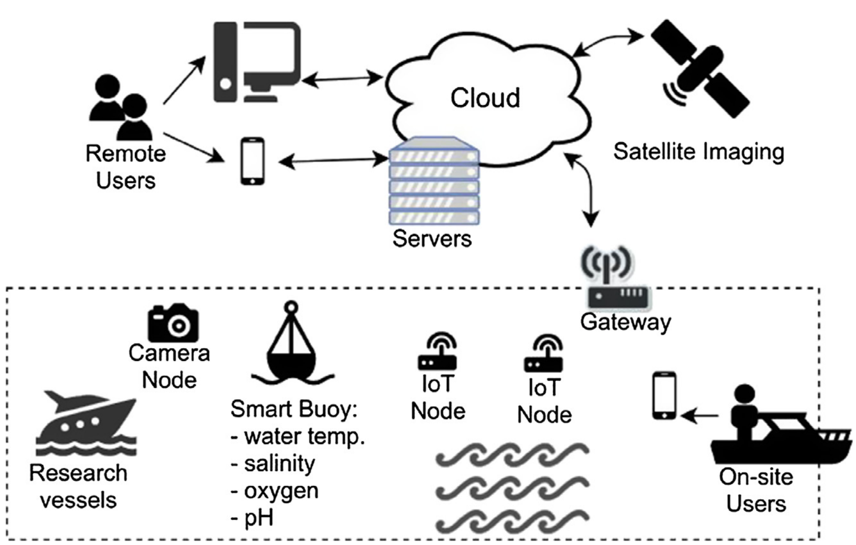

Fig. 4. Context view: mariculture and ecological monitoring.

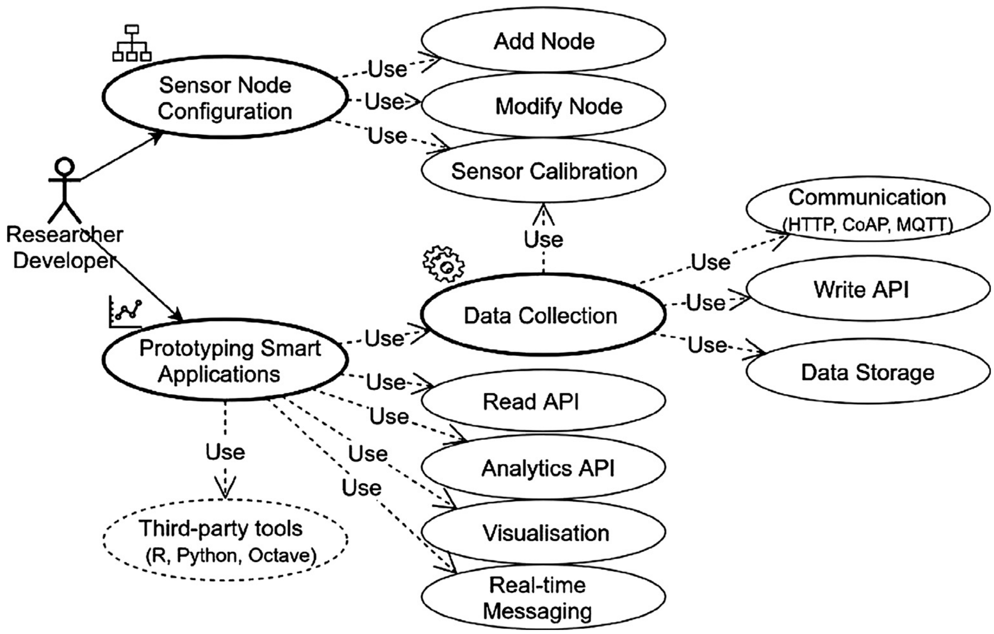

Fig. 5. Deriving the IoT platform-specific use cases.

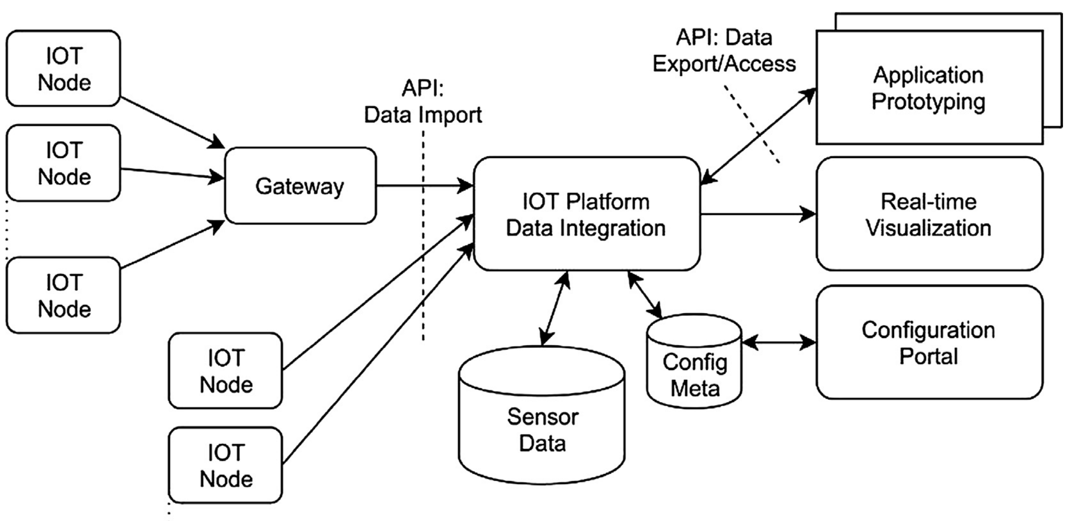

Fig. 6. IoT platform: interaction of the system blocks.

export/access (current and historical sensor data), thus enabling the rapid prototyping of research testbeds and end-user applications. In addition, the platform shall provide data visualization using real-time trend charts, as well as the configuration portal. It is anticipated that the future implementations will provide portal access to the analytics functions based on the integrated data.

The concept of utilization of the proposed platform for prototyping the end-user applications is illustrated in Fig. 7. In addition to the scalar sensor data, depending on the needs, the platform may be equipped with Cloud-based file management solution aimed at handling digital media images from on-site cameras, aerial imaging (drones), and satellite data. It provides both the user interface and APIs with the ability to upload, store, and access files, in this case digital photos, which may be used by prototype applications developed for the end users in near future. The digital media file management can be implemented using an open source solution called ownCloud (Moscicki and Lamanna, 2014).

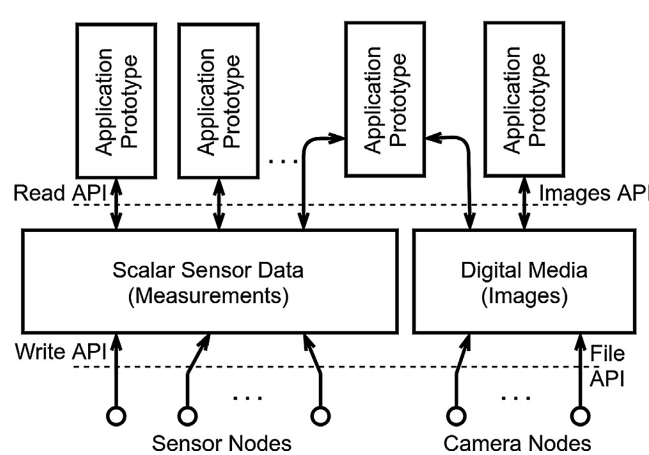

Fig. 7. Prototyping end-user applications using the IoT platform.

3.3. Implementation view

The implementation view of the proposed system is illustrated with component diagram shown in Fig. 8. Sensor nodes communicate to the data collection module using HTTP(S) requests that conform to the write API. When received, the data is persisted into the database, which is also used by other components (dashed arrows). For the user, the platform is implemented as a dynamic web portal using PHP and Laravel framework, which implements the Model-View-Controller or MVC architectural pattern (Olanrewaju, 2015). Using the portal, the user can access pages to subscribe, login, and manage the accounts. The web interface provides functions to add and configure sensor nodes, define calibration parameters, list nodes, and visualize data in real-time. The data can be exported and imported manually by the user for the purpose of analysis and backup. Finally, the data can be accessed using the read API for the use in newly developed research testbeds and applications. The read API shall provide for an easy implementation using HTTP(S) requests to obtain the current or historical data in JavaScript Object Notation (JSON) or comma-separated values (CSV) formats. Such an API allows simple interfacing to third-party analytics tools (Octave, Python/SciPy, R).

3.4. Dynamic aspects of the solution

The dynamic aspects of a solution based on the platform's use is shown in the UML sequence diagram in Fig. 9. The interaction between the IoT nodes and the platform is reflected in the continuous loop. It is anticipated that the sensor nodes will periodically update the sensor data, which is accepted and stored in the platform's database. This process takes place at all times and can be concurrent to the remaining processes. The sensor data can also be obtained on request.

As for the new applications, it is anticipated that the user will interactively initiate the analytics, then the application will request data from the platform, perform the analysis, and the results will be sent back to the user. These results represent the new augmented intelligence about the problem and support user's decision making process (as shown earlier in Fig. 1). The user can then request the action from the system, i.e. turn the irrigation on or off, and obtain the confirmation about these actions through the platform or directly from the analytics application. In the initial implementation phase, the focus is on providing the platform for developing the applications to obtain the augmented knowledge about the objects of interests. The applications can also be designed to run continuously (i.e. periodically) and then, depending on the results, to notify users using visualization or real-time messaging when the actions are needed.

3.5. Platform deployment: PaaS in a private cloud

The IoT platform has been deployed as PaaS in a private cloud in a university setting. The platform has been be installed on a server with Linux operating system, Apache webserver, MySQL database engine, and PHP support, also known as a LAMP stack (Lee and Ware, 2002). Depending on the needs, the file management platform, i.e. ownCloud can be installed on the same or on a separate VM. The target platform for this project has been defined as a virtual machine (VM), with following characteristics: 8–12 GB RAM, 4-core CPU, 500 GB hard drive, 2×1 GBit Ethernet connection, Ubuntu LTS Linux operating system, LAMP stack, and OpenSSH for remote access and management. As a cloud model, the platform follows the PaaS service model and private cloud deployment. The Cloud infrastructure for the current solution consists of 3 physical servers, 3PAR storage, double power supplies, fiber-optic Ethernet, and uninterrupted power supplies. The resources are managed with a bare-metal VMware hypervisor and organized into a cluster, which currently contains 60 CPU cores, 520 GB RAM, 10 TB storage, fiber communication, and fail over VM management. In the case that one server fails, the VM running the platform will be moved

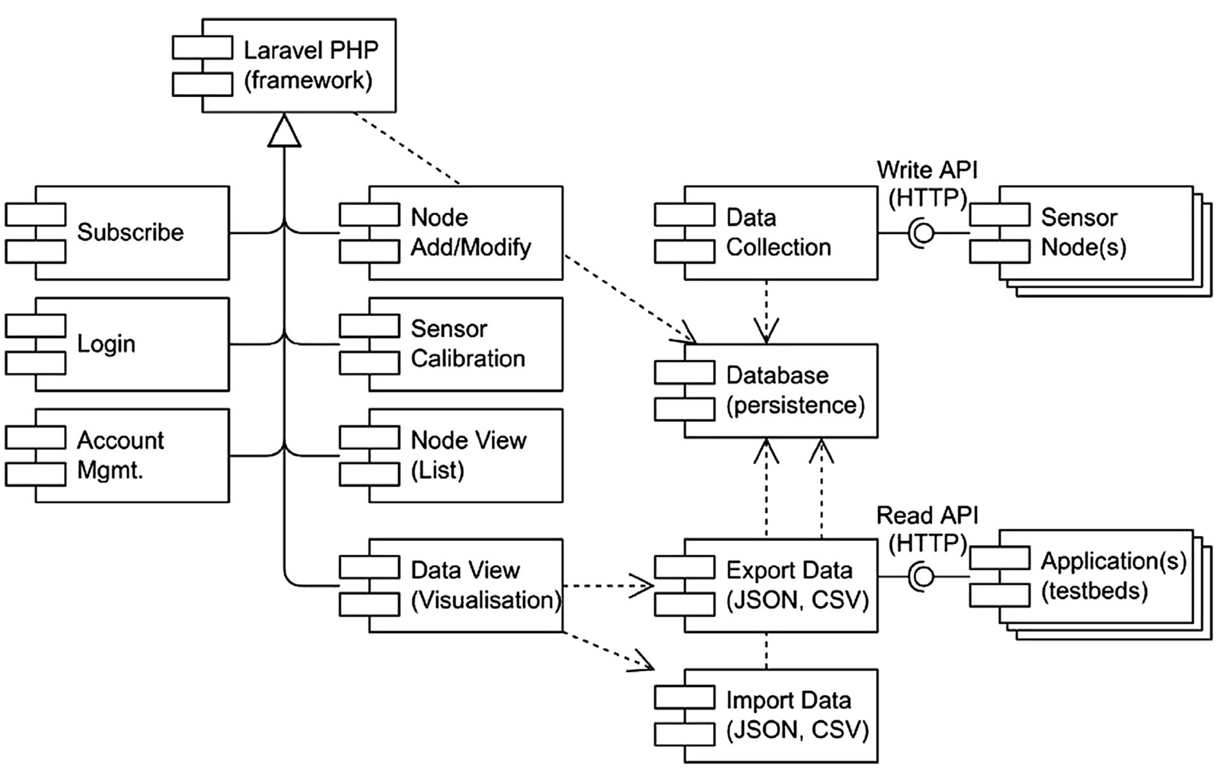

Fig. 8. System components from implementation perspective.

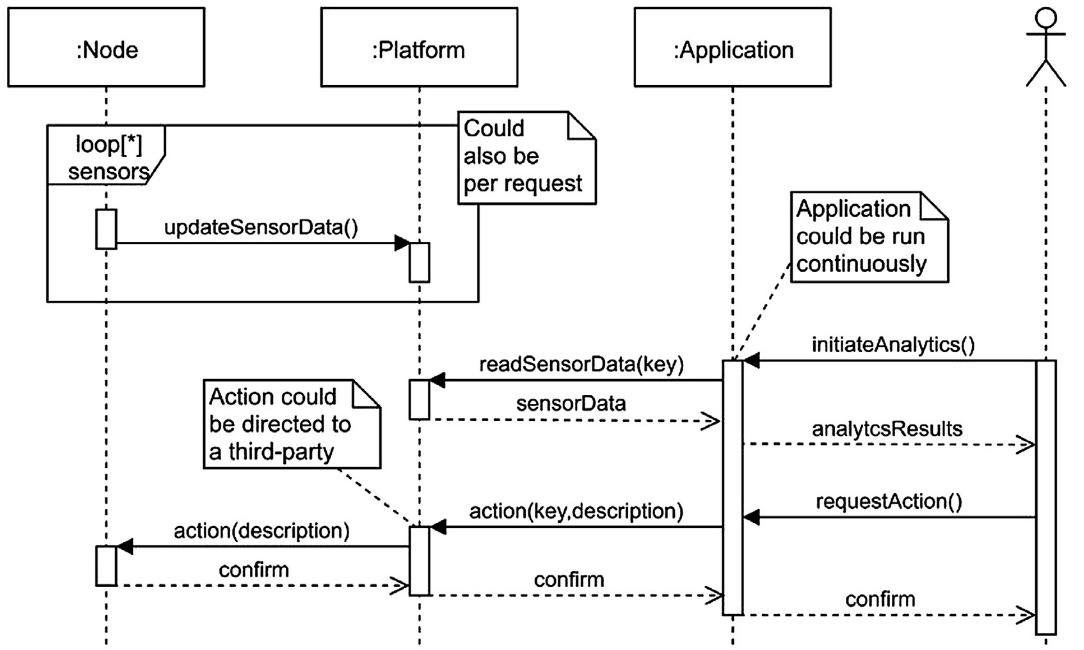

Fig. 9. End-user interaction with the prototype analytics application.

to another server with little or no down-time. The equipment is installed in a server room with backup power supplies and a controlled environment.

3.6. Data view: describing the information flow

The data view provides an overall perspective on the data flow, data types, formats, volume, and periodicity (Table 3). It can be seen that the majority of measurements come in the form of scalar data collected from various sensors. Conceptually, the camera nodes can be considered just another type of an IoT node with a camera as a sensor. From the practical implementation point, multimedia acquisition and communication by current IoT computing techniques is not easily doable, because digital media requires more processing power and higher data rates.

Table 3
Data view: type, format, and periodicity.

Group	Type	Source	Format	Periodicity
Weather	Temperature	Sensor	Scalar	1 m/15 m
	Temperature	Sensor	Scalar	1 m/15 m
	Pressure	Sensor	Scalar	1 m/15 m
	Humidity	Sensor	Scalar	1 m/15 m
	Wind speed	Sensor	Scalar	1 m/15 m
	...	...	...	...
Agro	Soil temp.	Sensor	Scalar	1 m/30 m
	Soil humidity	Sensor	Scalar	1 m/30 m
	Leaf wetness	Sensor	Scalar	1 m/30 m
	Solar radiation	Sensor	Scalar	1 m/30 m
	...	...	...	...
Water	Temperature	Sensor	Scalar	15 s/15 m
	Salinity	Sensor	Scalar	15 s/15 m
	Oxygen	Sensor	Scalar	15 s/15 m
	pH	Sensor	Scalar	15 s/15 m
	...	...	...	...
Misc.	Photo	Camera	Images	1 d/event
	Photo	Drone	Images	1 d/event
	Photo	Satellite	Images	1–14 d

The sources of information are sensors connected to various sensor nodes in the field. Sensor nodes convert the measurements into scalar data, integers or floating point numbers, and send them to the platform. The periodicity of scalar measurements is expected to range from 15 s to 30 m, and in some instances to be even more for slower transfer. Whenever possible, the measurements should be time stamped using Coordinated Universal Time (UTC). The data can be time stamped by nodes (raw data), but also on the server if the data are collected remotely.

It has been recognized that for some IoT scalar data, additional calibration is needed. The platform shall provide means for converting the raw data measurements into calibrated values. Non-linear calibration shall be supported by defining the approximation of the relationship between the raw data and calibrated values by specifying points that define linear segments as shown in Fig. 10. The figure illustrates how the non-linear calibration curve is approximated by three linear segments defined by four points. Three raw data values r_1, r_2, and r_3 are translated into calibrated values c_1, c_2, and c_3, respectively.

As for digital images, in this project their periodicity is expected to be once per day or slower. However, some future applications

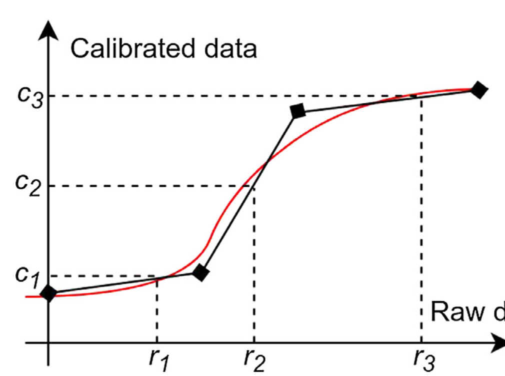

Fig. 10. Cloud-side calibration of IoT scalar data.

may require event-triggered image capturing by the camera nodes and may need more frequent transfer of digital images. Satellite images are accessed using API from the satellite data owner(s). The camera nodes need faster communication channels, more processing power, and support for digital media file transfer API.

Generally, the faster periodicity is expected for the research related applications and test pilots. The platform should be able to accept the data via standard communication channels and its API, utilize format unification, and be able to keep the data for at least five years (or longer if possible). The API shall be simple enough and agnostic to the data source. It should be able to accept the data from sensor nodes, computers, and third-party applications.

3.7. Interface view: write/read API and user interface

The interface view points out the key communication messages, API access, databases, interfacing to the users and other systems. The key interconnection of the platform and the rest of the world is through its write and read APIs, which employ HTTP(S) requests similarly to ThingSpeak IoT Platform. The write API is utilized by sensor nodes to send the data to the platform, while the read API provides for new applications and/or third-party systems to access the integrated data. As for the user, the platform is accessible using any standard web browser and behaves as a web portal with dynamic content.

The write API utilizes messages as simple as HTTP request by forming a URL in the following format:

http://[IP]/data?wkey=key&field1=val$_1$*&field2=val*$_2$*...*

where *IP* is the address of the server, *key* is a unique write key designated for each sensor node, and val_1 and val_2 correspond to sensor measurements at that node. Currently, the platform is configured to allow up to 10 measurement fields in a single URL that can pass the information via GET or POST method. Additional optional parameters include a time stamp created at the sensor node, as well as longitude, latitude, and altitude for the GPS (or other) positioning-enabled nodes.

The read API for retrieving the data from the platform relies on HTTP requests by forming URLs in the following format:

http://[IP]/[format]?rkey=key

where *IP* is the address of the server, *key* is a unique read key designated for each configured sensor node, and the *format* can take value of JSON or CSV corresponding to the desired data format. Additional optional parameters include a time zone offset, date range by specifying from and/or to date points (YYYY-MM-DD format), and a time range by specifying start and end time points in HH:MM:SS format.

The platform users, researchers and developers on the project, access the platform's web interface using their computers or mobile devices. The platform allows users to register and create their own profiles. Once registered, the user creates and configures their sensor nodes. For each node, the user can set up its location and measurements collected by that IoT node. The node configuration includes node name, geographic home position, text description, measurement fields and their display settings, as well as selecting whether the node will be public or private. For each node, the platform generates set of API keys for data write and read operations as mentioned before. API keys are managed by the node owner. There is also a group API key, which allows sharing data within a private group of users.

The user views sensor data and node position using the platform's real-time visualization in the form of trend charts and geographical maps. This data view can easily be shared and embedded into third-party web applications via iframe HTML element. The platform user is provided with an option to manually export and import data using CSV or JSON file formats.

3.8. Non-functional requirements

Non-functional requirements are used to define various criteria that characterize the system as a whole, rather than a specific behavior. The main non-functional requirement is to keep the project data and system prototypes on the private server infrastructure. This means that the IoT platform needs to be deployable using a private cloud model. The target platform for the project is the use of VMs in a university environment setting. Another important non-functional requirement is to utilize open source software whenever possible. This initial development uses the Ubuntu Linux operating system with LAMP stack (Lee and Ware, 2002) and the Laravel PHP framework (Olanrewaju, 2015).

The platform shall provide open APIs for both sensor data integration and data exchange with new applications (or other third-party systems). As for the sensor nodes, they shall be able to work in remote and inaccessible locations, i.e. long-term battery power supply with solar panels may be necessary in order to support communication over mobile networks. The applicable standards and/or widely accepted formats shall be utilized whenever possible (CSV, JSON, HTTP, CoAP Shelby et al., 2013, MQTT: MQ Telemetry Transport).

The platform shall provide means for authentication, authorization, management of different types of users and user roles, restricted access, and encryption. The overall security concerns, especially regarding the IoT nodes deployed in the field, is challenging and goes beyond the scope of this paper (IBM, 2015). The solution shall be installed on the existing university cloud infrastructure assuming the security policies implemented within the university's information system. The platform shall stay accessible for all the researchers involved in the project and all of the data will be kept for at least five years. Finally, the proposed platform shall be easy to install, maintain, and open to further development.

4. Evaluation and practical usage

The proposed architecture has been implemented using an agile development process, and it is currently being evaluated by the researchers on the project. The platform is expected to further grow based on the feedback from researchers and end users.

4.1. In-house evaluation

The IoT platform has been tested using various IoT sensor nodes including Arduino, Raspberry Pi, and Libelium Plug and Sense (Plug and Sense platform). Fig. 11 shows an example use of the platform's write API implemented using the Python programming language, which can be used on both PC and Raspberry Pi. As seen in the figure, all it takes to send data to the platform is to implement an HTTP request where the measurements are encoded into the HTTP link. In addition to the physical IoT nodes, several virtual nodes have been simulated using weather data and posting them to the platform as it was coming from the nodes installed in the field.

```
import urllib
import httplib
import time

# obtain sensor data: temp, pressure, humidity,…
 …
# encode and send
params = urllib.urlencode({'field1': temp, 'field2': pressure,
                           'field3': humidity, 'field4': windspeed,
                           'field5': clouds, 'wkey': 'writeAPIkey'})
headers = {"Content-type":
     "application/x-www-form-urlencoded", "Accept": "text/plain"}
conn = httplib.HTTPConnection("blueleaf.ac.me:80")
conn.request("POST", "/Data.php", params, headers)
response = conn.getresponse()
 …
```

Fig. 11. Making a write API request using python.

4.2. Evaluation with field-deployed sensor nodes

The platform has been used with IoT nodes that have been installed at various research and end user facilities. Smart Agriculture, Smart Water, and Smart Radiation IoT nodes from Libelium have been deployed in the field (Fig. 12). All of the nodes have been equipped with a GPRS modem (3G), 6600 mAh battery, and a solar panel for autonomous operation. They were configured to report the sensor data every 15 min. The Smart Agriculture node shown in Fig. 12a has been deployed in an experimental field that belongs to the Biotechnical Faculty in Podgorica, Montenegro. It has been equipped with sensors for temperature, humidity, pressure, soil temperature, wind speed, leaf humidity, and precipitation. The Smart Water IoT node has been equipped with sensors for water temperature, pH, and conductivity, which represents the salinity of the sea (Fig. 12b). The unit in the photo has been deployed at the fish and mussel farm that belongs to our industry SME partner located in Kotor Bay, Montenegro. Finally, the Smart Agriculture Pro unit has been installed at another SME partner's facility in Danilovgrad (Fig. 12c). Viewing sensor data for the unit installed at the vineyard that belongs to the Biotechnical Faculty's research facilities is shown in Fig. 13. This unit reports on air temperature, relative humidity, pressure, soil temperature, wind speed, and leaf wetness. More details on the Libelium sensor nodes can be found in Plug and Sense platform. All of the sensor readings, together with the battery status are sent to the cloud platform for storage, processing, visualization, and further use. Each IoT node has its home position, but can also report on the current position using longitude, latitude, and elevation.

4.3. Prototyping end user applications

The platform has already been put to use for prototyping the end user applications. As mentioned, the smart spraying system is being developed for disease forecasting of grapevine downy mildew (*Plasmopara viticola*) in the vineyard region of Montenegro (Sekulić et al., 2016). The smart spraying application is based on the 3 10 model and it collects weather data from the sensor nodes located in three vineyards. Based on the measurements (temperature, precipitation, relative humidity, and duration of leaf wetness), grapevine growth stages and field monitoring of the pathogen development, this system for disease forecasting is currently being calibrated and improved.

As for the ecological and seawater monitoring at the research facilities, the platform has been used for collecting data from commercial sensor nodes, but also to support the in-house development of Arduino-based sensor nodes (Bulatović et al., 2017). These sensor nodes report measurements on environmental and seawater parameters at multiple locations. The sensor data is being validated by comparison with the manually sampled periodic measurements. These development efforts supported by the use of the IoT platform are expected to result in fully functional applications for ecological monitoring, assessment of the state of marine environment, fish/mussel farm and port water monitoring, and assessment of beach water conditions.

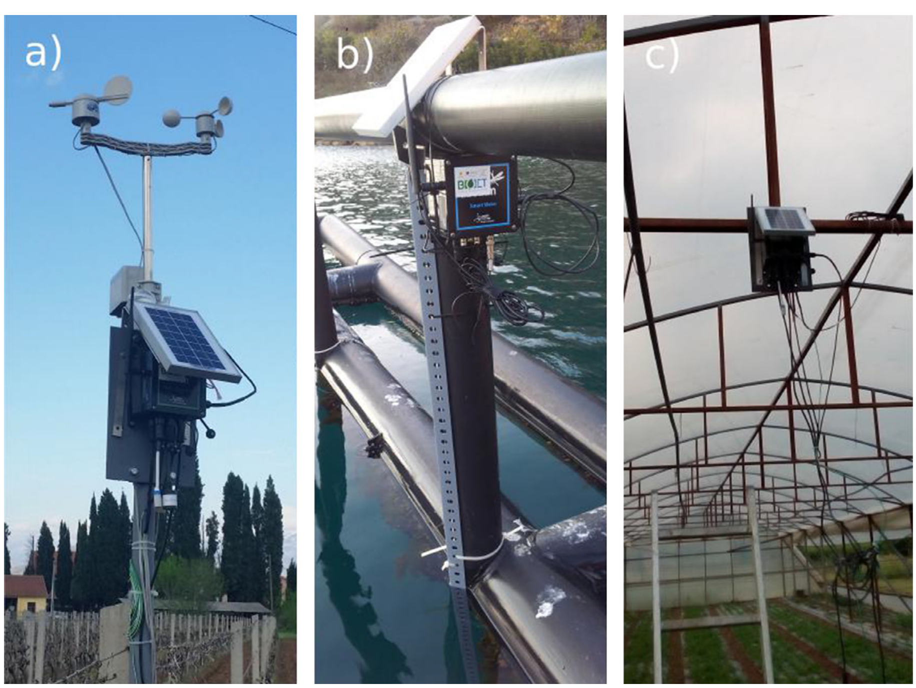

Fig. 12. Field evaluation with sensor nodes: (a) Smart agriculture node in the experimental field, Biotechnical Faculty, Podgorica. (b) Smart water node at the fish and mussel farm facility, SME partner in Kotor Bay. (c) Smart Agriculture Pro node at the greenhouse facility, SME partner in Danilovgrad.

Fig. 13. Viewing sensor data for Smart Agriculture IoT node in the experimental field.

5. Conclusions

This paper describes a case study of deriving the architecture for a private IoT-enabled platform for the research in precision agriculture, aquaculture, and ecological monitoring. In this study, the proposed solution is described using the approach of architectural views and viewpoints. The paper illustrates one implementation of the proposed architecture and its initial evaluation. The main contributions of the research are:

- The case study describes practical implementation of the platform and experience gained with its usage. The work has been done as collaboration of members of different faculties and SME partner institutions that provided support in the domains of system usage.
- The platform has been installed, evaluated, and validated with the IoT nodes based on Arduino, Raspberry Pi, and PC. It has been in use for several months, using commercial IoT nodes coming from Libelium's Plug and Sense family of products. The platform has also been used to support the in-house development of sensor nodes.
- The proposed solution will continue to be used for data collection and it is entering the stage of its use for prototyping the end-user analytic functions. The platform has already been used for development of smart spraying and irrigation, assessment of the marine environment and fish/mussel farm monitoring. The feedback from the researchers and end users will be used to improve and grow the platform.
- Further development is planned to include support for additional IoT protocols (MQTT, CoAP) and more elaborate integra-

tion with Open Source programming languages such as R and Python in order to expand the solution with comprehensive data analytics capabilities. Also, future work will include development of metrics and detailed experimental evaluation of performance, scalability, and reliability.

Acknowledgements

This work has been supported by the Montenegrin Ministry of Science and Ministry of Education under the HERIC project: BIO-ICT Centre of Excellence (Contract No. 01-1001).The authors would like to thank to all of the researchers and partners on the BIO-ICT project, for the support and constructive suggestions during the development and evaluation of the platform.

References

Bulatović, N., et al., 2017. Real-time measurements analysis of seawater physical parameters with SEMAR buoy system. In: Proceedings of the 22th Information Technology IT'17 Conference. Žabljak, Montenegro, pp. 153–156.
Cater, N.E., 2008. Smart ocean sensors web enabled ocean sensors for aquaculture. In: OCEANS 2008, pp. 1–9.
Cho, Y. et al., 2012. An Agricultural Expert Cloud for a Smart Farm. Springer, Dordrecht.
Directive 2006/113/EC of the European parliament and of the Council, 2006. Official Journal of the European Union (L376), pp. 14–20.
Directive 2009/128/EC of the European parliament and of the Council, 2009. Official Journal of the European Union (L309), pp. 73–86.
Evans, D., 2011. The internet of things: how the next evolution is changing everything. Cisco White Paper, 1–11.
Helmi, A.H.M.A., Hafiz, M.M., Rizam, M.S.B.S., 2014. Mobile buoy for real time monitoring and assessment of water quality. In: 2014 IEEE Conference on Systems, Process and Control (ICSPC), pp. 19–23.
Holler, J. et al., 2014. From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence. Elsevier, Atlanta.
IBM point of view: Internet of Things security, 2015. IBM Analytics White Paper, pp. 1–16.
ISO/IEC/IEEE systems and software engineering – architecture description, 2011. ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), pp. 1–46.
Jhuria, M., Kumar, A., Borse, R., 2013. Image processing for smart farming: detection of disease and fruit grading. In: 2013 IEEE Second International Conference on Image Information Processing, pp. 521–526.
Kale, S. et al., 2015. Data and monitoring needs for a more ecological agriculture. Int. J. Adv. Res. Comp. Sci. Softw. Eng. 5 (12), 804–807.
Kaloxylos, A. et al., 2012. Farm management systems and the future internet era. Comp. Electron. Agricult. 89, 130–144.
Krčo, S., Pokrić, B., Carrez, F., 2014. Designing iot architecture(s): a European perspective. In: 2014 IEEE World Forum on Internet of Things (WF-IoT), pp. 79–84.
Kruize, J.W. et al., 2016. A reference architecture for farm software ecosystems. Comp. Electron. Agricult. 125, 12–28.
Law on Marine Fisheries and Mariculture, 2009. Official Gazette of Montenegro 09 (49), pp. 1–35.
Lee, J., Ware, B., 2002. Open Source Development with LAMP: Using Linux, Apache, MySQL, and PHP. Addison-Wesley, Boston.
Mell, P., Grance, T., 2011. The NIST Definition of Cloud Computing, NIST Special Publication 800-145, pp. 1–7.
Montalvo, M. et al., 2013. Automatic expert system for weeds/crops identification in images from maize fields. Exp. Syst. Appl. 40 (1), 75–82.
Moscicki, J.T., Lamanna, M., 2014. Prototyping a file sharing and synchronization service with Owncloud. J. Phys.: Conf. Ser. 513 (042034), 1–7.
MQTT: MQ Telemetry Transport <http://mqtt.org>.
Nam, S. et al., 2005. Application of realtime monitoring buoy systems for physical and biogeochemical parameters in the coastal ocean around the korean peninsula. Mar. Technol. Soc. J. 39 (2), 70–80.
Ojha, T., Misra, S., Raghuwanshi, N.S., 2015. Wireless sensor networks for agriculture: the state-of-the-art in practice and future challenges. Comp. Electron. Agricult. 118, 66–84.
Olanrewaju, R.F. et al., 2015. An Empirical Study of the Evolution of PHP MVC Framework. Springer, Cham.
Papoutsa, C., Hadjimitsis, D., 2013. Remote Sensing for Water Quality Surveillance in Inland Waters: The Case Study of Asprokremmos Dam in Cyprus, Remote Sensing of Environment Integrated Approaches. InTech.
Plug and Sense platform <http://www.libelium.com>.
Regulation (EC) No 853/2004 of the European parliament and of the Council, 2004. Official Journal of the European Union (L139), p. 55.
Regulation (EC) No 854/2004 of the European parliament and of the Council, 2004. Official Journal of the European Union (L155), p. 206.
Rossi, V., Salinari, F., Poni, S., Caffi, T., Bettati, T., 2014. Addressing the implementation problem in agricultural decision support systems: the example of vite.net®. Comp. Electron. Agricult. 100, 88–99.
Rozanski, N., Woods, E., 2011. Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives. Addison-Wesley Professional.
Sekulić, P., Djukanović, S., Djurović, I., 2016. Detection of downy mildew in grapevine leaves using support vector machine. In: Proceedings of the 21th Information Technology IT'16 Conference. Žabljak, Montenegro, pp. 169–172.
Shelby, Z., et al., 2013. Constrained application protocol (CoAP), draft-ietf-core-coap-18 <http://tools.ietf.org/html/draft-ietf-core-coap-18>.
Shirish, P., Bhalerao, S., 2013. Precision farming: the most scientific and modern approach to sustainable agriculture. Int. Res. J. Sci. Eng. 1 (2), 21–30.
ThingSpeak IoT Platform <https://github.com/iobridge/thingspeak>.
Vermesan, O., Friess, P., 2013. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers, Aalborg, Denmark.
Vujović, S., et al., 2016. A web service for grapevine monitoring and forecasting a disease. In: Proceedings of the 21th Information Technology IT'16 Conference. Žabljak, Montenegro, pp. 25–35.
Xu, G., Shen, W., Wang, X., 2014. Applications of wireless sensor networks in marine environment monitoring: a survey. Sensors 2014 (14), 16932–16954.
Zaks, D., Kucharik, C., 2011. Data and monitoring needs for a more ecological agriculture. Environ. Res. Lett. 6 (1), 014017.