IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 7, 2020, accepted October 11, 2020, date of publication October 16, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031634

Laying the Foundations for an IoT Reference Architecture for Agricultural Application Domain

BASMA M. MOHAMMAD EL-BASIONI AND SHERINE M. ABD EL-KADER
Computers and Systems Department, Electronics Research Institute, Cairo 11843, Egypt
Corresponding author: Basma M. Mohammad El-Basioni (bbasioni@eri.sci.eg)

ABSTRACT The Internet of Things (IoT) relates to many billions of various applications and devices scattered around the world talk to each other and can exchange data and perform cooperative tasks without the intervention of humans. Towards efficiently realizing this, many things are needed to be achieved in advance, such as common language, basis of work and cooperation, roles distribution, resources availability, and security. Here comes the role of humans to build a reference architecture represents the common communication framework among the Internet things. There is no doubt that in order for the IoT to meet expectations, it needs to follow standardization; therefore, this paper addresses the IoT standardization by formulating the basis of an IoT reference architecture for the agriculture domain. The proposed Agricultural IoT Reference Architecture (AITRA) is based on a defined architecture generation process incorporates analysis of the IoT and the application domain ecosystems. AITRA is composed of three tiers: Device, Cloud, and Business, described in the paper including architectures, conventions, frame format, applications and services, and illustrative examples for utilizing the architecture at its highest abstraction level. The proposed design resulted in a foundation for a reference architecture combines the three main required features: best practices, common vocabulary, and reusable designs; characterized over the other architectures by its efficient low abstraction level meanwhile giving design freedom, lower time-to-market, standardization in its interfaces and communication protocol. It connects to its outside world with authorization rules and at any scale: individual, company, government(s), and global levels.

INDEX TERMS IoT, precision agriculture, reference architecture, standardization, vertical solutions.

I. INTRODUCTION

The IoT [1], [2] refers to a system of interconnected smart things built on top of the existing global system of interconnected networks of human-operated devices: the Internet; such that the human-operated devices' applications and the Internet-connected things can communicate with each other; the Internet-connected things communicate with each other, wherever and whenever they are; and the things can use the Internet resources and services. All of this with consideration to the nature of the things, which is different than the computers and mobile phones, and its lesser capabilities. Such system can cause radical changes to our life towards more efficient automated cost-effective implementation and management of different things; this appears well in being one of the most important engines of the fourth industrial revolution. The IoT applications [3], [4] span a wide range of application domains such as agriculture, constructions, smart homes and cities, medicine, transportations, etc. The IoT adoption all over the world continues to grow, and it is expected that the number of Internet-connected things reaches 22 billion by 2025. But nevertheless, can this huge number of things, which serve diverse application domains, talk and understand each other; can all the things that belong to the same application domain understand each other, cooperate, or share information; can any Internet application talk to them; can I improve my IoT solution by selecting the required functionality and performance of its components from different vendors; is IoT a global system such as its underlying Internet infrastructure.

With its enormous benefits, and at the same time its implementation challenges, the IoT represents a fertile subject for scientific research. The researchers do research on IoT security [5], embedded system designs for the smart things, low-power wide area network (LPWAN) technologies [6], Wireless Sensor Network (WSN) protocols [7]–[11], a number of IoT platform silos [12], etc.

The associate editor coordinating the review of this manuscript and approving it for publication was Nabil Benamar.

The IoT reference architecture also represents an important hot area of research. The IoT reference architecture can be defined as a collection of distributed software blocks, functions, services, relations, interactions, communication means, interfaces, abstracted functional descriptions, implementation targets, and non-functional requirements, represented in a certain architectural style. It provides guidelines for the IoT solution architects to implement a concrete architecture based on its incorporated best practice design patterns. This is usually without imposing specific implementations and technologies but only high level descriptions of possible solutions' components and structures.

The main argument behind using reference architectures in the design of IoT solutions is that the different solutions which are derived from the same reference architecture will be interoperable which increases the scalability of the overall solution. In addition, reference architectures approach aims to simplify the concrete architecture development process.

Most of the IoT reference architectures employ the layer-based architecture design approach [13], [14]. The layered architecture consists of various layers work as logical separations of different responsibilities in the architecture. The layer can contain components, may have dependencies on the other layers, and communicates with the other layers through defined interfaces via defined primitives.

The conventional IoT architectural models are composed of three layers: the Perception layer - is related to the smart devices and edge elements, the Network/Transport layer - relates to gateways and routers between the smart devices and the cloud, and the Application layer - relates to cloud servers, services, and database. For better performance and to address more complex systems and keep up with rapid technology advancements, this conventional model can be extended by adding one or more layers of abstraction such as, Support layer for APIs transparent communication between Network layer and Application layer; Business layer which always represents the higher abstraction layer manages the Application layer and takes care of various users' needs; Processing/Middleware layer takes the responsibility for dealing with the big data by discarding not useful or redundant data; Data analysis layer which is responsible for the data processing for producing reports, data mining, implementation of machine learning, etc.; Edge and Fog layers contain intermediate network devices between the cloud and the smart devices for offloading some of the cloud computing to the edge of the things network to decrease latency and network congestion, and for device-to-device connectivity, mobility support, security, etc. [15], [16].

In addition to these horizontal layers, some cross-cutting layers may exist which offer functions to all or a lot of the other layers, such as, a Security layer, a Trust and Privacy layer, and a Connectivity layer. The cross-layer design principle get rid of the strict horizontal layered-architecture and allows cross communication between the layers for the purpose of optimizing the performance of the overall system. The cross-layer and cross-cutting layer designs can be regarded as the same approach, but the closest to truth is that cross-cutting is a form of cross-layer realization; however, the cross-layer design can be realized by different forms: a common layer from where information can be shared among the other layers, direct interaction and information sharing by creating interfaces between non-adjacent layers, merging layers, manipulation of layer-specific parameters across all layers, or completely replacing layers with bidirectional links graph [15], [17].

The connectivity and communication issue is one of the most important enabling cross-cutting functions in the IoT architecture, especially the application layer communication technologies of the IoT protocol stack.

The two most commonly used communication paradigms in the IoT application layer are: request/reply and publish/subscribe. In the request/reply communication pattern, the client sends the request to the server, the server receives the request, processes it appropriately, and sends a response to the client. The request/reply (also known as request/response) is a synchronous communication pattern in which the client and the server exchange messages through one session in a blocking mode; the client is blocked waiting for the server reply before it closes the session. The most notable example of the request/reply pattern is the HTTP protocol.

Using the publish/subscribe communication pattern, there is no longer a direct connection between the sender and the receiver, instead, an intermediary broker/server receives the messages and distributes them. The message sender is known as publisher publishes its messages to a topic which represents a logical channel or queue. The topics are organized in a hierarchical topic tree for multicast or group messaging. Any application is interested in this topic's data can subscribe to it, and the broker will take responsibility for distributing the received message on the topic to all of its subscribers. With this paradigm, there is no need to continuously poll the server for new data and this is more suitable to real-time applications. This paradigm also has a lot of good features, such as scalability and decoupling between publishers and subscribers in synchronization, space, and time. The most notable example of the publish/subscribe pattern is the MQTT protocol. Fig. 1 illustrates these two communication patterns in a graphical representation.

With various combinations of these features, there are many IoT reference architectures were developed by different organizations, guide the development of IoT platforms used by the IoT applications developers as an infrastructure of their applications such that the development process of IoT solutions becomes easier and more accurate, the final product becomes more reliable and reaches the market faster; the following section highlights some of these reference architectures.

Each platform is conformed with its reference architecture generic specifications, but the same architecture's platforms may utilize different technological implementations which affects the previously mentioned argument behind using reference architectures. In addition, with this large number of

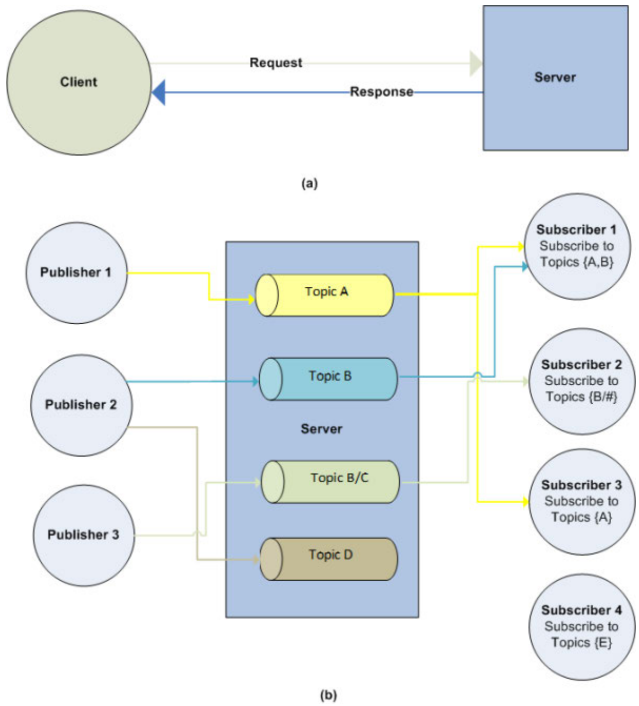

FIGURE 1. The two most common IoT communication paradigms, a) request/reply paradigm, b) publish/subscribe paradigm.

reference architectures and platforms available, the accurate selection between them is not an easy task at all, moreover, the time taken by the application developer to implement his solution specific features on top of the highly abstracted building blocks of the platform may be still considerably long and increases his product time-to-market (the time taken until the solution is ready to be used by the end-user starting from exploiting an idea or an existing infrastructure).

To be more beneficial to the overall performance of the IoT solution development process, the reference architecture should be domain-specific; considers the details of the intended domain upon an accurate analysis of its ecosystem; gives the developer low level of abstraction, in the form of a rich best practice-based set of building blocks related to the application domain as well as the IoT enabling technologies, but in a friendly understandable efficient way. For the IoT reference architecture to achieve its purpose at a global scale, it needs to be standardized; this is a vital issue that the world has already started paying attention to its importance recently. These issues represent the heart of the work on this paper.

This paper is intended to concentrate only on one application domain of IoT. From the broad range of IoT applications, the agricultural application domain was chosen to be the subject of study which deals with matters related to the previously raised inquiries. Sustainable agriculture, food security, and ending hunger formulate top goals the world seeks for achieving. According to that they represent a top priority in the UN Sustainable Development Goals (SDGs), and the 2030 Agenda emphasized the role of Science, Technology and Innovation (STI) to achieve the SDGs, particularly these goals.

The main contribution of this paper is the formulation of a standardized agricultural IoT reference architecture as a guide for developers targets providing templates facilitate the production of different complete solutions customized to each agricultural field and achieve interoperability between them and other beneficiary or helper external applications. The proposed reference architecture has other good features, such as modularity, low time-to-market, cross-layer design, friendly rich set of user interface components and design, vendor-neutrality, security support, reliability, scalability, standardized description of the local things network physical and logical topologies, and consideration of application modules dedicated to telemetry, management, and manipulation of the things network.

Other contributions of the paper work include: review of some IoT reference architectures and conducting a comparison between them and the proposed one, drawing a generation process for the reference architecture, farming ecosystem analysis including categorization of farming systems, definitions and specifications of the farming systems, and agricultural management practices analysis.

The remaining part of this paper is organized as follows: Section II reviews the related work; Section III states the problem the paper revolves around; Section IV mentions the methodology of the work and draws the reference architecture generation process; Section V performs analysis on the agricultural ecosystem; Section VI performs analysis on the IoT ecosystem; Section VII describes the structure of the proposed IoT Reference Architecture; Section VIII represents a simple proof of concept implementation of the proposed IoT Reference Architecture; Section IX discusses the proposed architecture characteristics; finally, Section X gives conclusions and suggests future work.

II. RELATED WORK

The reference architecture can be generic applies to all IoT application fields, fits to only one field, or deals with some fields' patterns and requirements. Also, it may be open-source or proprietary to one entity; It is intended for the proposed architecture foundation, AITRA, to fall under the open-source some-fields-dedicated type (some modules and libraries can be payable according to the developer identity). Both the research and the industrial sectors have witnessed the birth of a number of real applications and research efforts for developing IoT reference architectures with different types. The following discussion will shed the light on examples of each type.

The WSO2 open-source technology provider developed the extendable WSO2 IoT reference architecture [18] as a basis for creating IoT solutions. It is composed of these layers: the Device layer – each device must have a direct or indirect connection to the Internet, a unique identifier preferred to be specific to the device's hardware, and an OAuth2 Refresh and Bearer token [19] stored in a nonvolatile memory. The Communication layer – is responsible for the communication between the device layer and the cloud. The Aggregation/Bus layer – provides the support for specific communication protocol(s) (may be legacy protocols) and the bridging between different communication protocols, performs the aggregation and routing of the communications it receives, and acts as the OAuth2 resource server and policy enforcement point. The Event processing and analytics layer – provides near real-time data analytics on the data received from the Aggregation/Bus layer, stores it in a database, and allows querying it with simple SQL-like language. The Client/External communications layer – is an API-centric architecture composed of web-based front-ends, dashboards for data visualization, and an API management system. The Device Management layer – a cross-cutting layer maintains a list of devices with their identities, and composed of a Device manager manages the devices remotely and a Device management agents vary according to the platforms' and devices' types; this is upon three levels of management: fully managed, semi-managed, and non-managed. The Identity and Access Management layer – is a cross-cutting layer provides token-based and policy-based access control to the data. Table. 1 identifies some features of the WSO2 IoT reference architecture.

The European IoT-Architecture project (IoT-A) [21] aimed at developing an architectural reference model for the interoperability of IoT systems, outlining principles and guidelines for the technical design of its protocols, interfaces, and algorithms. It proposed an IoT Reference Model consists of three sub-models: the IoT Domain Model which introduces the main concepts of the IoT and their relations like Devices, Services, and Virtual Entities (VE); the IoT Information Model which defines the structure (e.g. relations, attributes) of IoT related information in an IoT system; and the IoT Functional Model which represents a number of Functionality Groups (FG) derived from the IoT Domain and Information Models. The IoT Architectural Reference Model (IoT-ARM) is built based on the IoT Reference Model from a set of FGs. The first one is the IoT Process Management FG, in which the Process Modeling FG is responsible for providing the tools used in modeling processes using a standardized notation, where these models are serialized and executed in the Process Execution FG. The Service Organization FG, is responsible for organizing Services of different levels of abstraction, it translates high level internal or external Service requests down to the concrete IoT services. The Virtual Entity and IoT Service FGs include functions that relate to interacting with the Virtual-Entity and IoT-Service abstraction levels. The Device and Communication FGs, the communication FG is based on the technologies related to the IoT systems and it takes care of the devices' communication technologies, the network communication, and the whole end-to-end communication abstraction. The two cross-cutting FGs are the Management and the Security FGs. IoT-A cannot be considered as an IoT reference architecture by itself, but it defines standardized architecture methodology with different views and perspectives from which different IoT architectures for special use cases can be derived. Table. 2 identifies some features of the IoT Architectural Reference Model.

TABLE 1. WSO2 IoT reference architecture features.

WSO2	
Type	Generic
Layers	5 horizontal layers and 2 cross-cutting layers
Supported devices vendors	Vendor neutral; started with supporting only mobile devices management and then evolved to support any IoT device management
Devices types	Internet-connected devices and gateways distinguished by 3 levels of management: non-managed, semi-managed and fully managed
Standardizing specification for georeferenced physical and logical topologies of the local non-IP-enabled devices network	Not considered
Security support	Identity & Authentication: SAML2 SSO and OpenID. Authorization: OAuth2 and attribute-based access control policy: XACML
Interface to users and developers (level of abstraction)	API
Mentioned user interfaces	Generic web portal and dashboards
Modularity and Extendibility (add/delete/replace components)	Modular architecture
Recommended IoT messaging technologies	MQTT over WebSockets.
Standardization of communication packets structure	Not considered
M-to-M and M-to-application communication support	Supported
Telemetry, configuration, and control of local non-IP-enabled devices network	Not considered clearly
External application interaction support	Supported
Naming convention	Not used
Ecosystem analysis and best practices	Lacks an application-specific ecosystem analysis
Resilience to changes in the underlying communication infrastructure	Not addressed
Completeness	It has a real instantiation, WSO2 platform [20]

The Industrial Internet Reference Architecture (IIRA) is a standards-based architectural methodology developed by the Industrial Internet Consortium (IIC) [22] enabling Industrial Internet of Things (IIoT) system architects to design their own systems based on a common framework and concepts [23]. An Industrial Internet Architecture Framework (IIAF) based on 'ISO/IEC/IEEE 42010:2011' [24] was defined to standardize the description of IIoT architectures and accordingly aid in the development, documentation and communication of the IIRA. The essence of IIAF is the viewpoints and stakeholders which are specified upon analysis of different IIoT use cases. The viewpoint frames the system concerns which represent the focus of attention of the system stakeholders. IIRA has main four viewpoints: Business viewpoint – is related to the identification of the stakeholders' business-oriented concerns in establishing an IIoT system; Usage viewpoint – is related to the identification of the sequences of activities involving human

TABLE 2. IoT-ARM features.

IoT-A	
Type	Generic
Layers	7 cross-layer-designed functional layers (FGs) and 2 cross-cutting layers
Supported devices vendors	Vendor neutral
Devices types	Constrained devices, IP-enabled devices, and gateway; managed by strategies for mitigating unexpected events, fault handling, management of the membership
Standardizing specification for georeferenced physical and logical topologies of the local non-IP-enabled devices network	Not considered
Security support	Provides functions for initial registration, identity management, trust–and-reputation model Authorization, and communication security
Interface to users and developers (level of abstraction)	Conceptual interfaces to IoT-augmented process models
Mentioned user interfaces	Additions and extensions to industry standards, for instance, BPMN 2.0
Modularity and Extendibility (add/delete/replace components)	Modular architecture
Recommended IoT messaging technologies	Constrained protocols (e.g., 6LoWPAN, UDP, CoAP, etc.)
Standardization of communication packets structure	Not considered
M-to-M and M-to-application communication support	Supported
Telemetry, configuration, and control of local non-IP-enabled devices network	Not considered clearly
External application interaction support	Supported
Naming convention	Not used in the architecture description but will appear in any Business Process Model will be employed
Ecosystem analysis and best practices	The IoT Reference Model should describe all the concepts and definitions on which the IoT architectures can be built. It doesn't provide application-specific ecosystem analysis, but allows for integrating IoT-aware process modeling standards that can describe the application ecosystem
Resilience to changes in the underlying communication infrastructure	The proposed IoT Communication Model consider the need for supporting emerging solutions, but it is not a specific communication stack, but a guide or approach from which one or more communication stacks can be derived
Completeness	Complete, but it appears to have very little instantiations

or logical users that deliver intended functionality to ultimately achieve the fundamental system capabilities; Functional viewpoint – represents the functional components in the IIoT system, their structure and interrelation, the interfaces and interactions between them, and the relationships and interactions of the system with external elements in the environment; Implementation viewpoint – is related to the specification of the technologies needed to implement functional components. The functional components/domains of the architecture highlight the important building blocks that have a wide applicability in many industrial verticals and address the actual integration of the Industrial Control Systems (ICS) into the IIoT systems where they include: Control Domain – a Functional Decomposition represents the ICS; Operations Domain – a Functional Decomposition provides general management of the Control Domain; Information Domain – a Functional Decomposition for managing and processing data; Application Domain – implements the application logic; Business Domain – provides implementations for business functions, such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Product Lifecycle Management (PLM), asset management, etc. This is in addition to enabling cross-cutting functions, such as Connectivity, Distributed data management, Industrial analytics. Table. 3 identifies some features of the IIRA.

In [25], an architecture design approach for IoT-based Farm Management Information Systems (IoT-based FMIS) is proposed. The approach consists of two basic activities: Domain Engineering and FMIS Development. In the Domain Engineering phase, consecutive steps are taken to develop an IoT FMIS family feature model that defines the common and variant features of the different FMISs, develop the reference architecture, then develop the reusable components that will be necessary to develop the FMIS based on the reference architecture. A specific IoT-based FMIS is then developed, using the Domain Engineering phase outputs, including activities as selecting IoT and FMIS features appropriate for the application, developing a specific FMIS application architecture, finally, implementing the specific IoT-based FMIS. The layered view of the IoT-based FMIS is composed of six layers, from bottom to top they are: the Device layer, the Network layer, the Session layer, the FMIS-Data acquisition layer which represents the Session layer interfaces, the FMIS-Application layer which contains IoT services and system management functions, and the FMIS-Business layer which provides the farm management functions such as Irrigation management, Pest management, Nutrient management, etc. Also, it incorporates two cross-cutting layers for security and management. Table. 4 identifies some features of the IoT-based FMIS reference architecture.

At the same time, there are a lot of existing IoT platforms that are implemented without following a specific reference architectural approach but upon a specific proprietary architecture often with commercial vendor lock-in nature. Amazon Web Services IoT (AWS IoT) [26] is a cloud platform enables secure, bi-directional communication between Internet-connected things (such as, sensors, actuators, embedded devices, and smart appliances) and the AWS Cloud over MQTT and HTTP. AWS IoT Core service makes it easy to use AWS services like AWS Lambda, Amazon Kinesis, Amazon S3, AmazonSageMaker, Amazon DynamoDB, Amazon CloudWatch, AWS CloudTrail,

TABLE 3. IIRA features.

IIRA	
Type	Industrial control application dedicated
Layers	5 functional domains and as many as cross-cutting functions needed by the functional domains
Supported devices vendors	Vendor neutral
Devices types	Edge nodes (e.g. sensors, actuators, devices, control systems and assets), gateways, and edge gateways, managed by commands for the control system manages its operations (e.g., onboarding components, configuration, policy, and software/firmware updates), and by commands from the control system to the system's actuators
Standardizing specification for georeferenced physical and logical topologies of the local non-IP-enabled devices network	A standardization is not specified, but as examples of architecture patterns and possible design space considerations to be refined in upcoming revisions, it states some local network topologies, such as: mesh, hub-and-spoke, and some possible variations of location awareness
Security support	Provides a guidance for improving organizational approaches, processes and the use of technologies for creating a trustworthy system in the informational document "Industrial Internet Security Framework"
Interface to users and developers (level of abstraction)	APIs and UI
Mentioned user interfaces	Collection of functions implement application logic, rules and models that realize specific business functionalities, such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Product Lifecycle Management (PLM), Manufacturing Execution System (MES), etc.
Modularity and Extendibility (add/delete/replace components)	Modular architecture
Recommended IoT messaging technologies	No recommended protocols, but possible design space considerations, such as a suggestion for Pub-sub (having some known structured information) and API based communication
Standardization of communication packets structure	Not considered
M-to-M and M-to-application communication support	Considered
Telemetry, configuration, and control of local non-IP-enabled devices network	Considered
External application interaction support	Depends on how well the IIoT architecture for the considered use case is described benefiting from the IIAF and its business viewpoint is exploited to guide in describing and developing the other architecture viewpoints
Naming convention	Used only in describing the architectures viewpoints not in the actual implementations of the concrete architectures' functions and components
Ecosystem analysis and best practices	Provides a framework guides in making this analysis
Resilience to changes in the underlying communication infrastructure	Depends on the final design of the concrete architecture
Completeness	Complete

and Amazon QuickSight, to build the IoT applications. It offers the possibility of custom IoT-Data endpoints discovery and offers some other IoT services, such as AWS IoT

TABLE 4. IoT-based FMIS features.

IoT-based FMIS	
Type	Smart farming application dedicated
Layers	6 horizontal layers and 2 cross-cutting layers
Supported devices vendors	It could be said that it is vendor neutral as the authors stated that the device layer specifications have not been discussed in detail, but can be easily added without loss of generality and applicability of the proposed approach
Devices types	IoT devices have a direct connection with the physical devices
Standardizing specification for georeferenced physical and logical topologies of the local non-IP-enabled devices network	Not considered
Security support	Mentioned as a cross-cutting layer without details, except for using SSL/TLS for TCP transport protocol
Interface to users and developers (level of abstraction)	Not specified
Mentioned user interfaces	Some interfaces were mentioned with no specification, such as: External Weather Forecast Interface and Finance Interface Module. Generally they are considered as interfaces with other external systems. Also it exposes the Business layer function such as Irrigation management, Pest management, etc.
Modularity and Extendibility (add/delete/replace components)	Modular architecture
Recommended IoT messaging technologies	Publish/subscribe or request/reply patterns: MQTT, CoAP, XMPP, AMQP, DDS
Standardization of communication packets structure	Not considered
M-to-M and M-to-application communication support	Supported
Telemetry, configuration, and control of local non-IP-enabled devices network	Considered
External application interaction support	Supported
Naming convention	Not used
Ecosystem analysis and best practices	Based on an ecosystem analysis via a feature-driven domain analysis approach
Resilience to changes in the underlying communication infrastructure	Not addressed
Completeness	A research proposal

Greengrass which extends the AWS to devices so as while they use the cloud services they can process the data locally; the FreeRTOS operating system for the constrained devices to connect to the cloud or to an Greengrass edge; AWS IoT Analytics for building analytics platform on the highly unstructured IoT data; AWS IoT Device Defender; AWS IoT Device Management; AWS IoT Things Graph; etc. AWS IoT Device and Mobile SDKs include open-source libraries, developer guides with samples, and porting guides to connect any IoT hardware platforms to the AWS IoT solution [27].

FIWARE [28], [29] is a framework of open source platform components which can be assembled together and with other third-party platform components to accelerate the development of smart solutions in multiple sectors. To set up a ''Powered by FIWARE'' platform, there are mandatory and optional components (Generic Enablers (GEs)) to include. The FIWARE Context Broker GE is mandatory as it is responsible for the core context management. Other optional GEs are available for enabling processing, analysis, and visualization of context information, and for enabling context data/API management, publication, and monetization. FIWARE Next Generation Service Interfaces (NGSI) is the API exported by a FIWARE Context Broker enables the integration of the platform and other applications to query, subscribe, update the platform context information. The IoT devices/gateways connect to the FIWARE platform need to support the NGSI API or use appropriate IoT Agent.

Other existing commercial IoT platforms for agriculture such as rayven [30] and thethings.iO [31], each of them offers some services to its customers allow for real-time information access and analyzing the information using different facilities such as, applying predictive and prescriptive maintenance to the farm machinery, monitoring cattle and control its feeding remotely, applying machine learning algorithms to data; and each of them has different features such as security, hardware-agnostic, multiple transport protocols agnostic, dashboards creation facility, etc.

III. PROBLEM STATEMENT

A lot of application diversities exist underneath the agricultural domain such as: open field farming, horticulture, greenhouses, livestock farming, urban farming, soilless culture, and aquaponics. In the same time, there are a lot of commercial IoT-based complete solutions tackle different agriculture's aspects, and a lot of implementations and research efforts built on existing generic IoT platforms.

The IoT products represent vertical solutions; each one has specific bottom-up architecture and it is directed to a vertical market to meet its customers specific needs. Therefore they lack the interoperability feature which enables the IoT solutions to grow at a large-scale (one can imagine the terrible difficulty to integrate in one application a big number of IoT enabled devices manufactured by different vendors based on different standards).

The ultimate goal is to develop an IoT enterprise solution addresses the main interoperability axes: the technical, syntactic, semantic and organizational interoperability issues. This enterprise solution allows different systems, companies, and organizations to communicate, share resources, and work cooperatively; offers applications-related ready features; thus aids in eliminating the need to develop from scratch, shortening the time-to-market, and consolidating reusability; and as a consequence facilitates the IoT solutions development and improves its overall performance.

This is achieved by standardization, but standardization is not an easy task as the IoT ecosystem is multidimensional incorporates a lot of domains, vertical solutions, and domain-specific factors. It is very difficult to devise a standard IoT architecture fits to all the points of the IoT ecosystem space, therefore, the trend is to develop IoT reference architecture with design-choices used to produce concrete architectures fit the diverse IoT space points. Again this requires an accurate definition of the IoT ecosystem which indoors is a difficult matter. This difficulty is reduced and the solution becomes more accurate when we convert the problem to developing a reference architecture for a sub-space of the IoT ecosystem corresponding to a specific IoT application domain; this is the approach followed in this paper where the agricultural applications domain is considered.

TABLE 5. IoT application ecosystem.

	Stakeholders					
	S_1	S_2	S_3	....	S_{n-1}	S_n
Domain$_1$-Use case$_1$						
Domain$_1$-Use case$_2$						
Domain$_1$-Use case$_3$						
....						
Domain$_2$-Use case$_1$						
Domain$_2$-Use case$_2$						
....						
Domain$_f$- Use case$_1$						
Domain$_f$- Use case$_2$						

TABLE 6. IoT application domain ecosystem.

	Stakeholders					
	S_1	S_2	S_3	....	S_{n-1}	S_n
Domain$_1$-Use case$_1$						
Domain$_1$-Use case$_2$						
Domain$_1$-Use case$_3$						
....						

In order to further illustrate the problem, Table 5 through Table 9 contain more explanations.

Table 5 shows that the IoT application ecosystem is composed of different domains, each domain is represented by large number of use cases which are analogue to different scenarios, features, and requirements, in addition each scenario is relevant to a different set of stakeholders. The effective analysis of the IoT ecosystem to create scalable efficient IoT solutions entails the consideration of all of these scenarios which is nearly impossible and not available in the existing platforms.

Table 7 illustrates how the existing IoT platforms and commercial vertical solutions form separate silos have different architectures and implementations, supported H.W. and S.W., achieved QoS metrics. On the other hand, efforts and endeavors have been and are still being made to produce IoT reference architectures try to consider the IoT ecosystem including the application requirements and describe this for the architects in an abstracted way to guide them in implementing a more specific instances based on the reference architecture suitable to the use case in hand.

TABLE 7. **IoT platforms and vertical solutions.**

	Architecture					H.W. vendors and types					S.W. technologies and protocols vendors and types, components and interfaces					QoS groups			
	A_1	A_2	...	A_{n-1}	A_n	H_1	H_2	...	H_{m-1}	H_m	W_1	W_2	...	W_{k-1}	W_k	QG_1	QG_2	...	QG_r
Platform/vertical$_1$																			
Platform/vertical$_2$																			
Platform/vertical$_3$																			
Platform/vertical$_4$																			
....																			

TABLE 8. **IoT application ecosystem analysis-based highly abstract Reference Architectures (RA).**

	Architecture					H.W. vendors and types					S.W. technologies and protocols vendors and types, components and interfaces					QoS groups			
	A_1	A_2	...	A_{n-1}	A_n	H_1	H_2	...	H_{m-1}	H_m	W_1	W_2	...	W_{k-1}	W_k	QG_1	QG_2	...	QG_r
RA_1 instantiation$_1$																			
RA_1 instantiation $_2$																			
RA_1 instantiation $_3$																			
....																			
RA_2 instantiation$_1$																			
RA_2 instantiation $_2$																			
RA_2 instantiation $_3$																			
....																			

TABLE 9. **IoT application domain ecosystem analysis-based less abstracted standardized Reference Architectures (RA).**

	H.W. vendors and types					Selection from the implemented S.W. technologies and protocols					QoS groups			
	H_1	H_2	...	H_{m-1}	H_m	W_1	W_2	...	W_{k-1}	W_k	QG_1	QG_2	...	QG_r
RA concrete architecture$_1$														
RA concrete architecture $_2$														
RA concrete architecture $_3$														
....														

The reference architecture can make all of its derived solutions vendor-neutral with respect to the H.W., but still some of these solutions cannot use the services offered by each other due to different inclusion and implementations of the reference architecture's components; Table 8 gives an explanation of this.

Table 6 shows, how the IoT application ecosystem shrunk when one application domain only is considered, while Table 9 shows the characteristics of concrete system architectures derived from a standard reference architecture based-on the analysis of the application domain ecosystem.

Upon the above discussion, the possible followed activities to produce an IoT solution for the use case in hand using different approaches are presented in Fig. 2. From Fig. 2, it could be concluded that it is easier for the architect to build his solution on existing ready-to-use components or even to follow reliable guidelines in implementing his solution, but this also costs him additional effort and time in reviewing all the possible available options, and practice using the selected one (of course this cost may be reduced or eliminated in the subsequent times of building solutions).

It is better for the quality of the produced solution to be built upon the best practice ready components or guidelines, but the selection criteria may limit the possible options which may threaten the solution's quality, also, the resultant solution in this case is subject to be isolated from other applications and solutions.

Regarding the time-to-market, the commercial vertical customized solution represents nearly zero time-to-market. The time consumed in case of using a reference architecture if the design process steps followed from its beginning can be equal to or greater than the time consumed in case of designing from scratch.

Fig. 2(d) represent the case of standardized domain-specific reference architecture with small time to market; in a nutshell, the three previously highlighted problems, which are the effective comprehensive analysis of the IoT ecosystem, the vendor-neutrality, and the interoperability among the solutions and external applications, can be addressed by

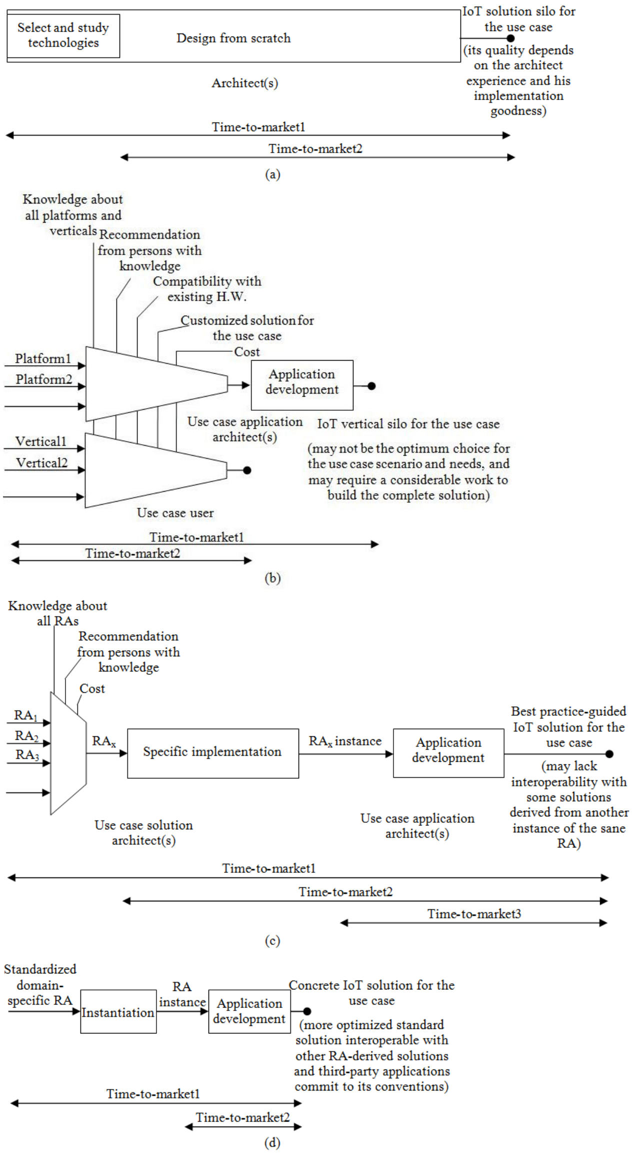

FIGURE 2. Illustration of the time taken until the IoT solution becomes ready to use in different development cases.

this approach. It targets developing a global standard in the intended application domain; by following its specification, the systems and applications all over the world can share and update data, use services, and cooperate; it targets to represent a comprehensive guide for architecting instantiations may implement all of its services as design-choices for all the domain's known use cases, at the same time, inspires formulating more use cases and new services, allow for extensions and implementation updates without affecting its existing derived solutions.

The design of such architecture determines further features, it may make its instantiation/platform approximately equal to one vertical solution for the entire application domain enriching the platform with its advantages especially the near zero time to market. This what this paper is trying to lay the foundation for, taking the agriculture field as the application domain, starting with specifying a design generation process and analyzing the domain ecosystem in the following sections.

IV. DEVELOPMENT METHODOLOGY

The development of an agricultural IoT architecture requires describing the ecosystem of the farming activities - the agricultural ecosystem, analyzing the farming system components and their interactions, and defining the stakeholders, reflecting their views and perspectives of the system, their roles in designing the architecture, and their requirements from it. Fig. 3, shows a heuristic scheme for the generation continuous process of the agricultural IoT reference architecture. The process starts with the analysis of the farming activities, which will be addressed in Section V, and can be used as a guide for possible architecture's functionalities. The process also depends on the analysis of the IoT ecosystem including illustration of its components - will be addressed in Section VI - and the IoT architects' briefing on IoT enabling technologies. Then the step of the architecture development comes, taking into account the stakeholders, the possible mechanisms, protocols, models, existing solutions, and assumptions. The proposed IoT architecture will be introduced in Section VII.

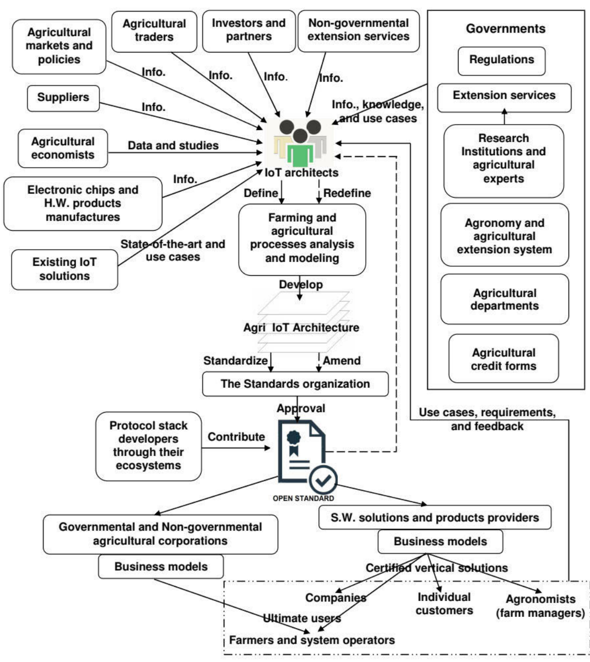

FIGURE 3. Stakeholders definition and IoT ecosystem's architecture generation process.

V. ANALYSIS OF THE FARMING ECOSYSTEM

This section will talk about the main building blocks that may constitute a farm, then it will touch the agricultural practices and their different implementation variations, finally, it will shed the light on the concept of Best Management Practices (BMPs).

A. FARMING SYSTEM TYPES AND SPECIFICATIONS

The first thing to talk about and to be considered in the farming activities is the farming system. The farming system is an appropriate combination of farm enterprises (e.g., cropping systems, horticulture, livestock) for better utilization of resources.

Fig. 4 shows different types of farming systems (mainly based on what you are growing on your farm) categorized by its main purpose to subsistence, semi-commercial, and commercial. In subsistence farming, the farmer cultivates crops or raises livestock mainly to feed him and his family being self-sufficient. In semi-commercial or semi-subsistence farming systems, the farmer targets supporting his family by producing the foodstuffs needed for their consumption and producing materials needed by his farm. To meet the other non-farm needs of his family and his farm such as fertilizers and pesticides, he may resort to generating cash income from selling the surplus farm products to their needs or cultivating cash crops dedicated for the purpose of selling; this, in essence, serves also the domestic consumption. The commercial farming is the farming method in which crops and livestock production is practiced with the intention of selling the products on the market. Commercial farms usually involve large tracts of land. Some farming systems may be established in the form of a small project generates cash income for a family or used for domestic use or established in a large commercial bases.

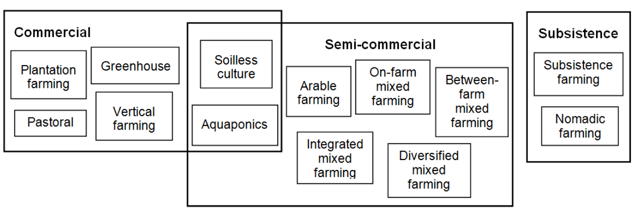

FIGURE 4. Farming systems types categorized by its main purpose.

In arable farming, only crops are planted in fields. This is in contrast to pastoral/livestock farming/grazing where only livestock and not crops is produced for dairy, beef, leather, wool, etc. The plantation farming (tree crop farming) is like arable farming in producing only crops; the difference between them lies in that the plantation farming grows only one crop on a relatively large piece of land (plantation farming requires more than 40 hectares in order to be successful); it is usually done on commercial bases with an aim to serve large market and sustain huge population.

Nomadic farming is similar to pastoral farming aims at producing livestock, however, the herdsman carries his belongings accompanying his entire family, clan, or even community, and moves his animals around in search of suitable grazing fields and water.

The shifting cultivation is a repeated process of the farmer moving with his household/community to a new forest land, clearing it by felling and burning of the vegetation residue, using it to grow crops until it loses its fertility, coming back to the previously cleaned lands when it regains its fertility. This practice requires a special life style and it is discouraged in modern days due to the scarcity of fertile lands. The government also discourages this practice due to the dangers it poses to forest reserves and nature: it is an unsustainable agricultural practice.

In mixed farming, the farmer increases income and minimizes risk through employing different sources by cultivating crop(s) and raising animal(s) simultaneously on the same piece of land on a small to medium scale. It can be categorized as diversified, where crops and livestock co-exist independently, versus integrated systems where the resources are recycled; or categorized as on-farm mixing, where resources are recycled on the same owned farm, versus between-farm mixing.

In addition to these types of farming systems, the following types can be added:

Horticulture, the agricultural science of growing and caring for plants, especially its two variations: greenhouse horticulture and soilless culture.

Greenhouse horticulture in its most modern and sophisticated form "Controlled Environment Horticulture/Agriculture (CEA)" incorporates the cultivation of horticultural crops within structures to modify their growing conditions and protect them from pests and diseases. By utilizing high technology greenhouses, it is possible to continuously monitor and adjust the growing conditions such that we can produce high yields and high quality crops, even under adverse growing condition and all over the year.

Soilless culture, is increasingly adopted as a major technological component in the modern greenhouse industry [32]. As its name indicates, dispensing of the soil as a medium for plants cultivation; instead mineral nutrient solutions (dissolved fertilisers in water) are provided to support plant growth and development. So that, we are kept away from the pathogens accommodated by the natural soil, its effect in monoculture degradation, and its possible infertility or salinity, and we can overcome the lack of land available for soil-based farming.

Another type is aquaponics [33]: the combination of aquaculture (fish farming) and hydroponics. The hydroponics system needs to be flushed regularly, this raises the issue of the need for regular waste disposal. The aquaculture system needs that excess nutrients be constantly removed from the system which means replacing this nutrient-rich water with fresh water on a daily basis. An aquaponic system combines the two systems exploiting these two needs where the aquaculture system feeds the hydroponics system with nutrient-rich water, and during this process, nitrifying bacteria in the hydroponics system breaks down the aquaculture waste into nitrates which are subsequently used by plants as nutrients.

A. Farm enterprises

ID	Name
1	Crop processing factory
2	Cropping field/Cropping system
3	Greenhouse(s)
4	Plant factory
5	Livestock
6	Natural grazing field
7	Owned pasture land
8	Poultry mobile coop
9	Soilless growing media
10	Air stones and air pumps and timer
11	Grow baskets/Net pots/ Grow bed containers
12	Inclined grow tray and a drain pipe
13	Nutrient solution reservoir
14	Mist nozzles
15	Biofilter
16	Fish tanks
17	Bee hives
18	Animal Feeding Operations (AFO)

B. Cropping patterns

ID	Name
1	Monocropping
2	Crop Rotation
3	Sequential Cropping
4	Mixed Intercropping
5	Row Intercropping
6	Stir Intercropping
7	Relay Cropping
8	Medium culture
9	Aeroponics
10	Open solution culture hydroponic gardening
11	Closed solution culture hydroponic gardening
12	Cascade cropping
13	Aquaponics

C. Crop types

ID	Name
1	Grain crops
2	Pulse crops
3	Oil seed crops
4	Forage crops
5	Fiber crops
6	Tuber crops
7	Tree crop
8	Perennial shrubs
9	Spice
10	Some annual crops
11	Horticultural crops
12	Starch crops/domestic food crops

D. Livestock

ID	Name
1	Cattle
2	Sheep and lambs
3	Poultry
4	Fish
5	Goats
6	Camels
7	Horses
8	Donkeys
9	Mules
10	Bees

E. Level of capital and labor use

ID	Name
1	Intensive Farming
2	Extensive Farming

F. Main Inputs/Needs/Factors

ID	Name
1	Specialized skill and know-how
2	Slightly sloping land
3	Fertile soil with balanced moisture
4	Warm climate
5	Dedicated infrastructure
6	Fertilizers
7	Tropical climate
8	High annual rainfall
9	Efficient management
10	Mechanization
11	Irrigation facilities
12	Special life style
13	Pesticides
14	Climate change adaptation
15	Soil order and type
16	Steep area slopes
17	Cold strong winds
18	Wet climate
19	Social organization and transaction costs
20	Regulations

FIGURE 5. Farming systems specifications.

TABLE 10. A map between the different farming systems and their specifications.

Specification type / Farming system	A	B	C	D	E	F
Plantation farming	1,2	1,2,4-7	7-10	--------	1	1,6-8,10,11,13
Pastoral	5-8,17,18	1-7	4	1-10	2/1	4,6,8,11-13,15-18
Greenhouse	3	3-7	11	--------	1	1,6,9,10,11,13
Vertical farming	3,4	8-13	11	4	1	1,6,9,10,11
Soilless culture	3,4,9-14	8-12	11	--------	1/2	1,6,9,10,11
Aquaponics	3-5,9-16	8-13	11	4	1/2	1,9,10,11
Arable farming	2	1-7	1-6	--------	2/1	1-4,6,10,11,13
On-farm mixed farming	2,5,7,18	1-7	1-10	1-9	2	1,6,8,11,13,15
Between-farm mixed farming	2,5,7,18	1-7	1-10	1-9	2	1,6,8,11,13,15,19,20
Diversified mixed farming	2,3,5,7-16,18	1-13	1-11	1-10	2/1	Depending on the mixed enterprises
Subsistence farming	2,3	4-6	12	--------	2	6,14
Nomadic farming	5,6	-------	-------	1,2,5-9	2	12

The water now is free of nutrients and can be recirculated back to the aquaculture system.

The vertical farming (plant factory or indoor farming) uses CEA technologies to grow high-value horticultural crops in vertically stacked layers in pre-existing urban warehouses or shipping containers using environmental control (humidity, temperature, gases, etc.), fertigation, and artificial control of light (as a supplement in the case of a rooftop greenhouse, or as a unique light source in vertical farms). This yields crops per square meter more than that traditional farming or greenhouses yields. Vertical farming also uses less water, grows plants faster, and can be used year-round – not just in certain seasons. Its facilities also can, in theory, be built anywhere [34].

As the mixed farming can be achieved by mixing different crops or different types of livestock or mixing both crop(s) and livestock, the sources mixing can also be between the traditional horizontal farming and horticulture, aquaponics, and vertical farming. More detailed specifications of a farming system are depicted in the tables shown in Fig. 5, categorized by specification types which are distinguished by a different letter in each table header and include: various possible enterprises and components that can constitute a system, various cropping patterns, various crops' groups, livestock variations, level of capital and labor use: whether the farming system is intensive uses big capital investment or labor intensively, or it is extensive uses little capital or labor. Finally, a separate table lists the main requirements and inputs of the farming systems, its characteristics, and the most critical risks and important crucial treatments that should be taken into account. Table. 10 maps the different farming systems with their different specifications exist in Fig. 5.

B. AGRICULTURAL PRACTICES ANALYSIS

Considering the indicated variations of the farming systems, cropping systems, crops and livestock, and the variations in the agricultural management practices, the farm manager needs to put a system and a plan to manage his farm and

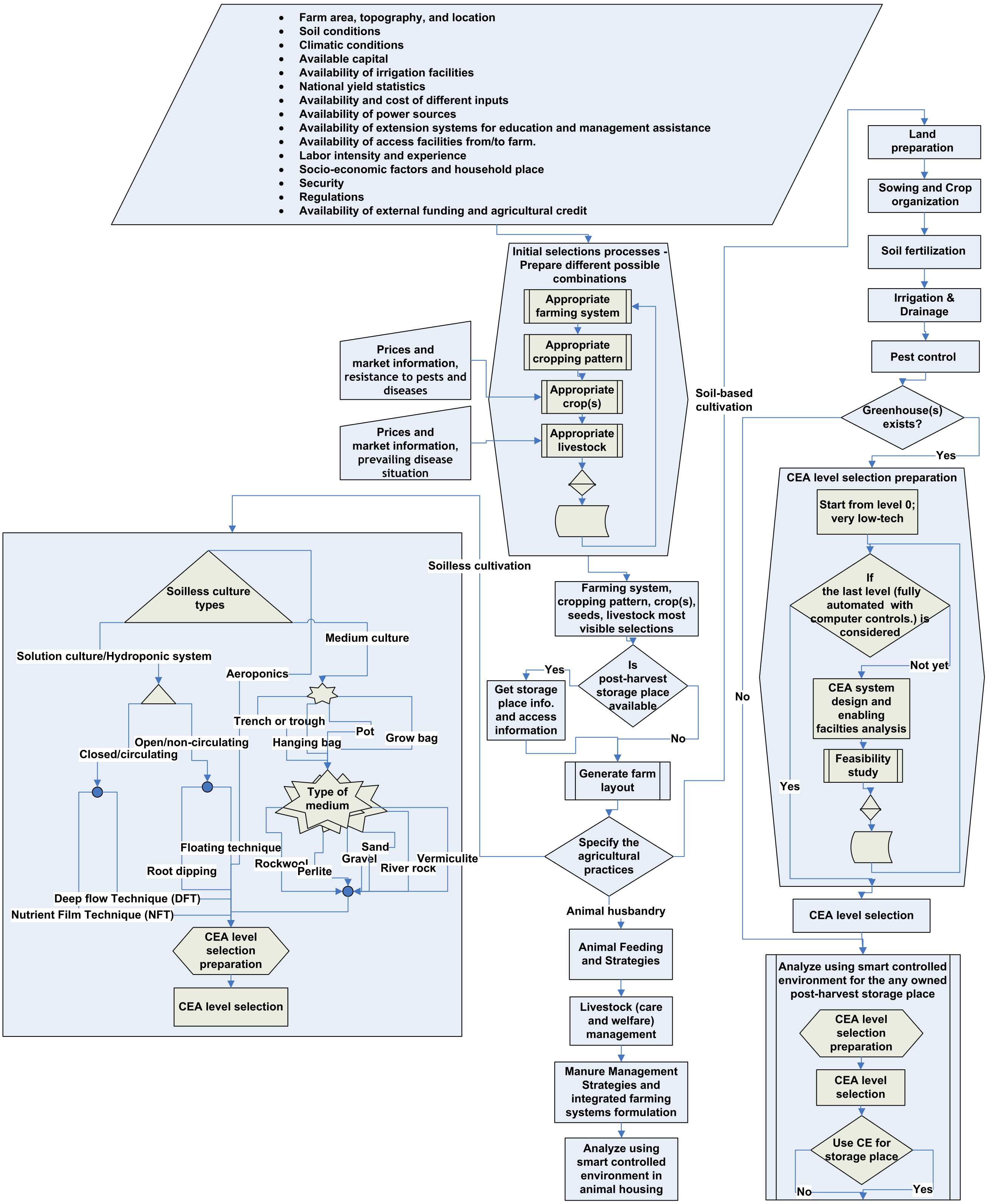

FIGURE 6. Farming activities management planning.

select among different variations. Fig. 6 shows a flowchart for managing the farming activities; it differentiates among three types of agricultural practices management: soil-based cultivation, soilless cultivation, and animal husbandry. It mentions the use of agriculture automation and controlled environment via well-defined standard controlled environment levels for well-defined standard cases of farming systems and enterprises. As the cultivation in the soil requires a lot of practices and special considerations, these practices are put as empty blocks in Fig. 6 to be described in more details with separate flowcharts in Fig. 35 - Fig. 39 in Appendix A.

C. AGRICULTURAL BEST MANAGEMENT PRACTICES (BMPS)

BMPs are methods developed for performing the different agricultural practices, would improve the quality and quantity of the farm outputs, aid in using agricultural resources efficiently in an environment-friendly manner, and cause economic growth. These management methods include all the agricultural practices such as: conservation tillage management, pest management, nutrient management, irrigation management, grazing management.

Precision Agriculture (PA) technology is a type of BMPs or, in better words, PA is a set of enabling technologies for the BMPs; it is a concept deals with managing crops based on the status of the different sites in the field so that the right amount of inputs are applied to different sites at the right time in the appropriate way. PA technologies include remote sensing, GPS, GIS, variable rate technology, automated steering systems, process-based modeling such as crop modeling, statistical and prediction models, embedded systems, communication technologies, drones, and sensor networks. IoT is the technology that is used to raise the efficiency of the remote monitoring and control in PA applications, thus, the IoT architecture should support the use of PA technologies.

VI. IOT ECOSYSTEM ANALYSIS

After analyzing the IoT application domain ecosystem - the agricultural ecosystem - and its stakeholders, defining possible architecture's functionalities, and according to the heuristic scheme for the agricultural IoT architecture generation process given previously, this section describes the IoT ecosystem which is considered as the last step in paving the way for designing the agricultural IoT reference architecture. The IoT ecosystem and accordingly the ecosystem of an IoT reference architecture is composed of six components namely: gateway, smart devices, the Internet, Cloud, features and functions, and entities.

1) IOT GATEWAY

It acts as the two-way bridge between the smart devices and the Internet. It can be a dedicated device or an existing device augmented with the appropriate IoT gateway software, or an existing thing or legacy equipment prepared with the appropriate hardware and IoT gateway software. Its functions include: solving the issue of interoperability among different technologies and manufacturers exist in an IoT system, connection management between the smart devices and the Internet servers or other smart devices, handling of security issues such as network-based encryption, functioning as IoT edge computing device provides a data preprocessing power on behalf of the less powerful smart devices and close to them improving the efficiency of data transport and such performs related tasks, e.g., data aggregation, storage of big amount of data, devices management and control.

2) SMART DEVICES

Any device that can process data and communicate it, can be described as smart; and if it communicates the data over the Internet to another device or application and receives data from them, it is considered as an IoT device. According to this, the IoT gateway represents an IoT device but the communicated data is just conveyed for other smart devices which can sense, process, actuate, and communicate data to the gateway through single hop or multi-hops. These devices represent an essential part of the IoT ecosystem. However, IoT devices can be able to sense, process, actuate, and communicate data over the Internet directly, and also may contain the thing itself which is referred to the asset we want to monitor or control. This results in having an IoT system of IoT devices with different capabilities.

3) THE INTERNET

The global communication system over which the things communicate using IP stack via different types of media.

4) CLOUD/SERVER

The IoT system, especially the one with big generated data, uses the cloud computing (or simply "Cloud") benefits and different service modes to build the platform that will provide the architecture features and functions to its users in an efficient and scalable way.

5) FEATURES AND FUNCTIONS OF THE ARCHITECTURE

This includes the implementation of services the architecture users/application developers can utilize to build their applications, the application programming interface, the implementation of these services' enabling technologies, techniques, tools, protocols and frameworks, and ready-made graphical interfaces and application building blocks. This can be offered through three components: database management system, dashboard, and analytics to analyze a big volume of data producing other meaningful data for achieving desired real benefits.

6) ENTITIES

They are all the entities or individuals involved in the complete cycle of the architecture development and usage. From Fig. 3, they include non-governmental extension services, investors and partners, agricultural traders, suppliers, agricultural economists, electronic chips and H.W. products manufactures, agricultural markets, governments, IoT architects, the standards organizations, protocol stack developers, S.W. solutions and products providers, governmental and non-governmental agricultural corporations, agronomists (farm managers), individual customers, companies, farmers and system operators.

Based on this analysis, a representative system architecture diagram of the IoT ecosystem is depicted in Fig. 7.

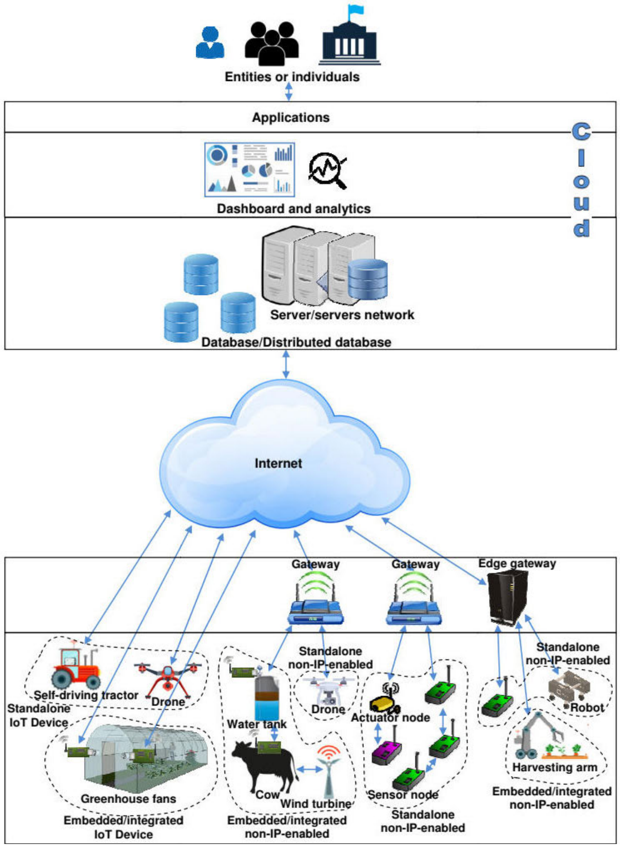

FIGURE 7. System architecture diagram of the IoT ecosystem.

The devices in the Device layer are classified as will be described later in the proposed AITRA Device tier.

VII. THE PROPOSED AGRICULTURAL IOT REFERENCE ARCHITECTURE (AITRA) DESIGN

The previous analysis of the IoT and the agricultural application ecosystems is exploited to formulate the proposed agricultural IoT reference architecture. AITRA can be described in terms of a high abstraction layered system architecture as well as a corresponding lower abstraction three-tier system architecture.

A. AITRA LAYERED SYSTEM ARCHITECTURE

The AITRA layered system architecture is shown in Fig. 8. The users can build a basic front-end solution, using the ready-made very user-friendly rich GUI components; the solution communicates directly with the devices via the Transport layer services and can be customized with coding.

The user can also use the same rich set of GUI components to include ready-made applications from the Application layer and enabling services from the Service layer in his solution. The ready-made application may be built by the developers to use Service layer and Transport layer services. The Information and data management layer services are also abstracted by the friendly GUI for Business applications, and

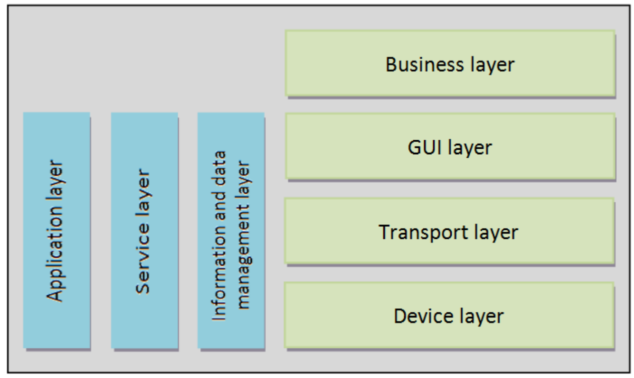

FIGURE 8. AITRA layered system architecture.

all the other layers can use this layer exposed services' APIs to build their functionality. As noticed from Fig. 8, the Device layer in turn needs applications and services suitable to its nature and role, and of course it needs data management functions.

As the implementation of these layers spreads over different distributed hardware, the details of the architecture are best to be described through a three-tier system architecture.

B. AITRA THREE-TIER SYSTEM ARCHITECTURE

AITRA is composed of three tiers, namely Device tier, Cloud tier, and Business tier, communicate through messaging and web browsers-servers communication protocols.

1) THE DEVICE TIER

This tier is composed of the smart devices which represent a part of the communicating things over the Internet, and they have two main types, namely, Gateway and IoT Device. On behalf of local networked non-IP-enabled devices of limited capabilities, the Gateway communicates the data to the other Internet things through the Cloud tier.

The gateway can be an Edge gateway where edge computing paradigm is employed for achieving better performance for certain agricultural applications with respect to real-time response, bandwidth consumption reduction, and offline local devices interactions and management.

The local non-IP-enabled devices transmit their data to the Gateway through certain topology; they can be standalone devices such as: sensor and actuator nodes, drones, robots, self-driving tractors, or may be embedded/integrated to the monitored/controlled thing such as: vehicles, pivots, sprinklers, fertilizer injectors, greenhouse and plant factory actuators like fans, heaters, ventilation system, lightening system, carbon dioxide injection unit, shutters, tanks filled with water, feed, fertilizers, and fuel. Both standalone and embedded device can contain sensors as well as actuators.

The IoT Device may be as the standalone or the embedded devices, but it always has the capability to securely send its data though the Cloud directly. The architecture of the Gateway, Edge gateway, and IoT Device are shown in Fig. 9.

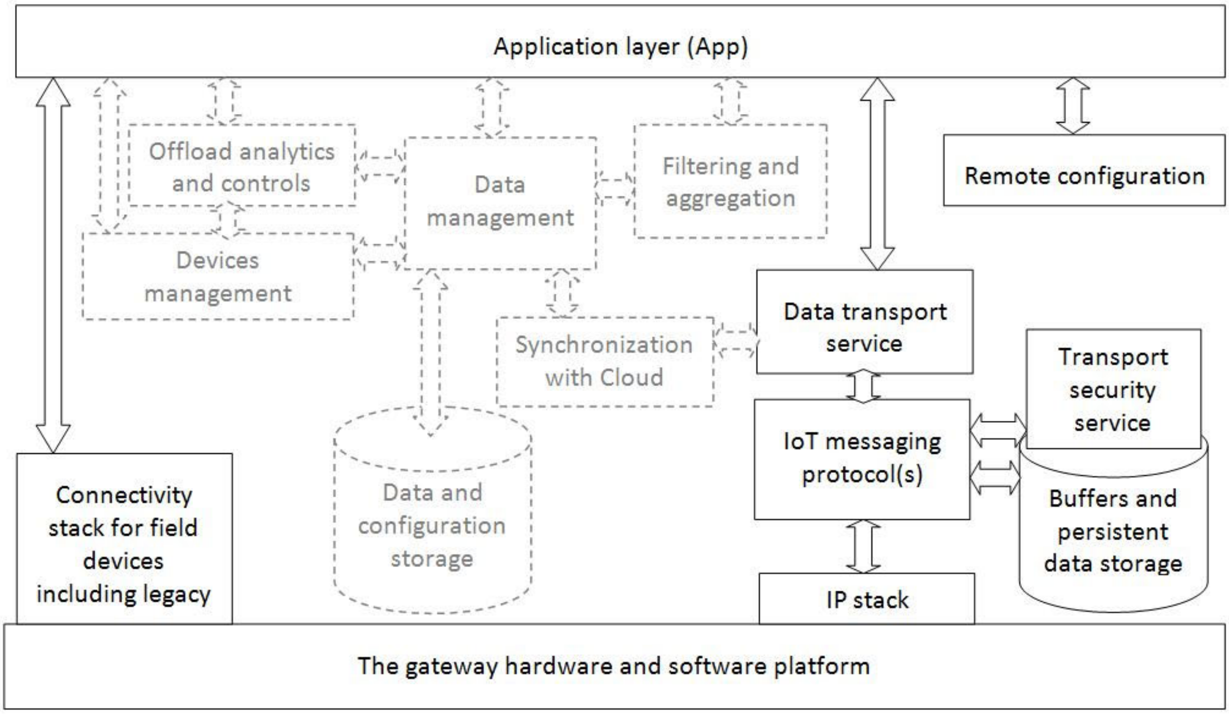

FIGURE 9. Illustration for Gateway, Edge gateway, and IoT Device architectures.

In Fig. 9, the solid-line shapes represent the main components of the Gateway, while the dashed ones represent additional components if it is an Edge gateway. The IoT Device contains all the solid-line shapes except the connectivity stack for field devices; it is replaced with sensors and actuators drivers.

The up-down arrow indicates information flow in the two directions between two components through their interfaces; the flow can be a request to a service offered by the component, or a confirm of the completion of the requested service, or an indication that an event relevant to a component's functions has occurred.

The Data transport service is responsible for all the communications to and from the Cloud. It indicates to the App the reception of a message from the Cloud, receives message transmission request from App, receives IoT messaging configuration requests from App, creates and handles the IoT messaging client, configures the IoT messaging dynamically by monitoring the transactions' quality, selects communication server for allowing scalability and availability (servers network/bridge or any other type of routing employed), detects the cut of the Cloud connection, and may store the messages directed to the Cloud during disconnection to be resent after reconnection through the Synchronization component. The Synchronization component also synchronizes the stored data marked by its source component as synchronizable.

When the Gateway receives the configuration and control commands sent by the Cloud for one or more constrained smart device, it can appropriately command the specified Device(s) or direct the received commands to the Devices management component and Offload analytics and controls component, respectively, to appropriately process them before commanding the Devices.

The Filtering and aggregation component contains techniques for eliminating redundant data, and it analyzes the devices' aggregated data for transmitting only a useful representative value of it. The Devices management component deals with the remote requests of setting the constrained devices' parameters and configuring their network, also it performs locally the network telemetry function and sends only alerts to the Cloud for events that warrant attention.

The Offload analytics and controls component executes trained AI models for image processing and voice recognition and takes locally the specified action. Also, it keeps the schedules of devices' different operations, such as the irrigation schedule, and operates them accordingly.

The App is the Gateway highest abstraction level provides the APIs of the services to the Gateway application developer through which he can control the initiation and configuration of the Gateway and develop a higher configuration GUI abstraction layer for the Gateway users. Furthermore, App code is responsible for translating the common structures of the Cloud packets data into the appropriate instructions to the Gateway components and the local devices network regardless of their interfaces implementation and software and hardware technologies used; likewise, it prepares the data to the Cloud in the common structures it understands.

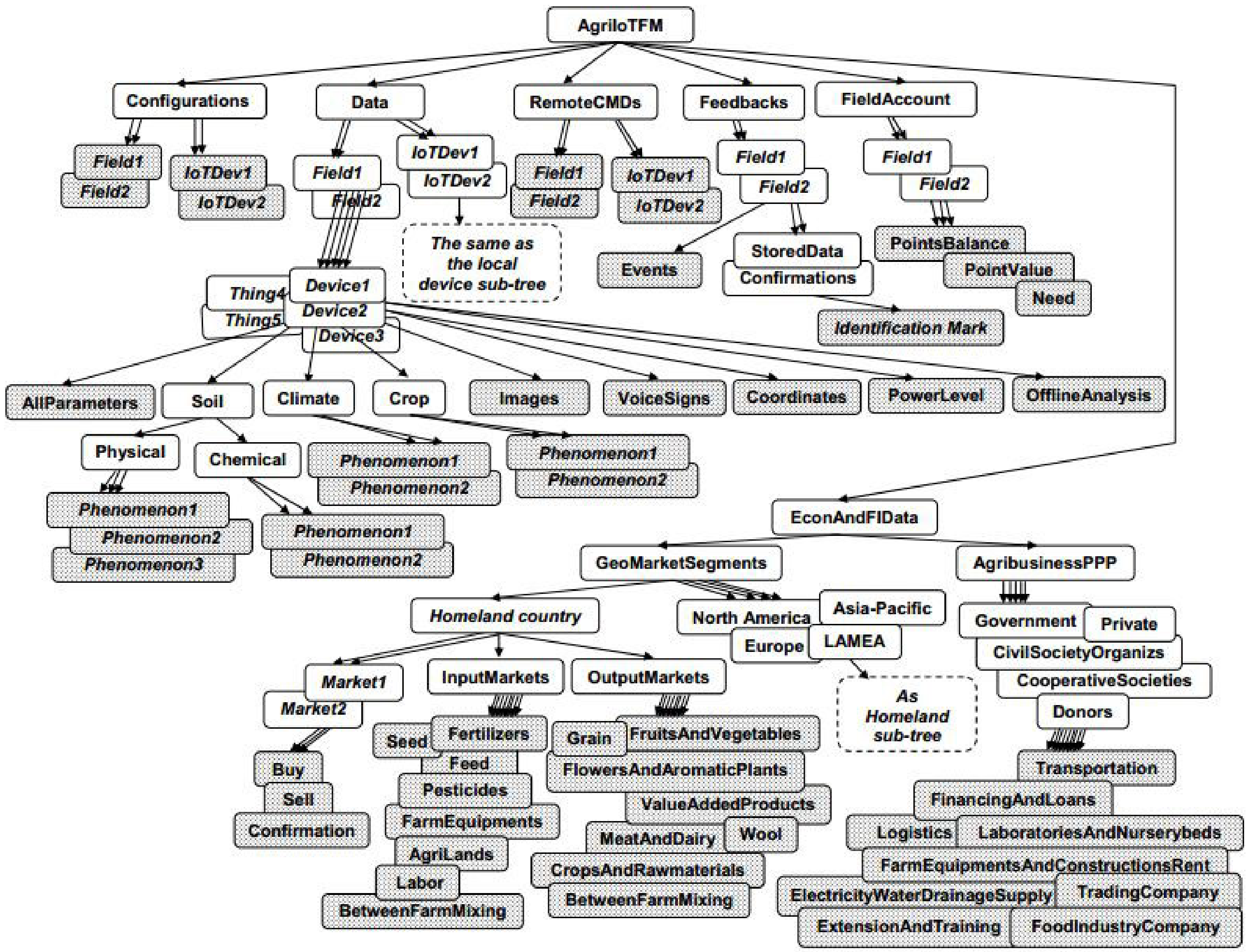

FIGURE 10. AITRA tree-based topic structure (the text is italicized when needs to be replaced by a corresponding value and the numbering means that the current tree level contains items as much as its higher level contains variations of these values, grouping rectangles means that they are in the same level and identical in their lower branched sub-tree).

The Gateway interfaces' conventions, and examples for preparing and passing the payload frame with settings for messaging are out of the scope of this paper.

The Gateway and the IoT Device communicate with the Cloud tier through certain frame structures encapsulated into the messaging protocol packet payload. The Gateway App can selectively switch among different IoT messaging protocols enabled in it and the Cloud server, or simply for lightweight IoT device software implementation, they can communicate using one publish/subscribe protocol utilizing its tree-based routing structure to send all types of messages.

The topic tree structure is shown in Fig. 10, there are two main branches related to the communication between the Cloud modules and applications: EconAndFIData and FieldAccount. The EconAndFIData is related to Economics and Financial Inclusion services categorized as data of services offered by different sectors and important for achieving Public–Private Partnerships for Agribusiness development, and data of local and international markets categorized mainly according to market geographic segments. The data contained in the payload of the application packets published to these topics incorporates further specifications, standardized categorizations and format, and information of the sector, the store and the sold thing.

The FieldAccount is related to an account dedicated to each field's owner contains special balance as a count of points; each point has a corresponding monetary value. The points added to its balance by the owner of the system, in this case it is the government, according to the field's transactions in the system and its production as bonuses. Also, the government determines its value at the country level based on the current state of the national agricultural economy. The field can use its points balance in its transactions on the system. In addition to that, through the FieldAccount, the field can distribute its needs.

The branching of the topics' levels and the published data to different topics are made to be almost comprehensive to aid in data visualization, finding matches automatically, doing statistics and economic analysis, and efficiently developing agribusiness PPP.

Octet: 1		1	1			1			Variable							
Bits:4	4	8	1	2	5	2	4	2	0/16	0/32	16	Variable	Variable	0/32	0/32	Variable according to area shape
Version	Reserved	Transmission options	Initial or Update	Contact type	Reserved	Device type	Capabilities	Field area shape	Pass code	Mobile phone no.	No. of PAN devices	PAN physical topology	PAN logical topology	Lat.	Lon.	....
														Device location		Field area coordinates
Frame control			Payload control						Payload							

FIGURE 11. Configuration message frame format.

Bits: 8	16	16	48	8	16	16	48	8	Variable according to no. of devices' groups
No. of devices' groups	Group1 start range device ID	Group1 end range device ID	Group1 devices type	Group1 associated farm enterprise ID	Group2 start range device ID	Group2 end range device ID	Group2 devices type	Group2 associated farm enterprise ID	....

FIGURE 12. PAN physical topology subfield structure.

The frame format of a Configuration frame published to the topic "AgriIoTFM/Configurations/*Identifier*" is shown in Fig. 11, where Identifier refers the unique Client Identifier with which the Gateway or the IoT Device connects to the IoT messaging server, recommended to be a word distinguishes the agricultural field or the IoT device. The Identifier can be preceded by a higher level represents an Identifier of a group of correlated fields/farms and IoT Devices. The interested applications of all of these configurations subscribe to "Agri-IoTFM/Configurations/#". If the value of the first subfield of the Payload control is set to 0, it indicates that this is the initial configuration, while a value of 1 indicates it is an update for the existing configuration.

The Transmission options subfield is a bitmap field dedicated for specifying transmission determinants, such as the case of transport change; mostly it will be set by the Data transport service. The Device type subfield can be 0 to indicate a Gateway, 1 to indicate an Edge gateway, or 2 to indicate an IoT Device and in this case the Payload first two fields related to PAN are set accordingly and the PAN logical topology field is omitted.

The Contact type and the Mobile phone no. payload fields are used to set the contact information of the farm. If the Contact type is the email method, the Mobile phone no. is omitted and the messages are sent to the farm using an automatically created account on the AITRA mail server with an ID equals to the farm registered name.

Capabilities is a 4-bit bitmap field indicates the device's characteristics; the 4 bits determine respectively continuous power supply availability, mobility support, GPS support, and code protection. The code protection is used to authenticate the devices' configuration/registration, especially, the devices that are belong to a correlated group.

Field area shape subfield can indicate a circle, square/rectangle, or a polygon; a value of it equals 0 means no area location information and the Field Area coordinates subfield is omitted. The Device location subfield contains the device coordinates if it is GPS-enabled.

The Field Area coordinates subfield contents depend on the area shape: radius and center location for a circle, locations of the diagonal two points for square or rectangle, and number and locations of a polygon corners.

The three Personal Area Network (PAN) related subfields (where PAN refers to the local devices network) give the possibility to the Gateway operator to describe the current distribution of the devices in relation to the incorporated farm enterprises and the functions/specifications of each device. They also allow the Gateway to describe the current PAN structure and communication pattern.

The structure of the PAN physical topology subfield is shown in Fig. 12; it assumes that the devices in the field are divided into groups, each group is homogeneous with respect to device type and is deployed for a specific farm enterprise identified by its identification number (ID), one farm enterprise can contain more than one group with similar or different type. The devices' IDs start with 1 (ID equals 0 means the Gateway) and each group spans a contiguous range of IDs, therefore can be indicated by its start and end IDs (or start ID and group members count). The specification characterizes each device capabilities can be well-known of commercial names or different combinations of different capabilities represented as devices examples; whether the

Bits: 8								8	16	16
Main category	Mobility	GPS	Mains	Battery	Rechargeable battery	Rechargeable battery with power harvesting	Reserved	Rechargeable battery type	Device/ Thing type ID	Model or Spec ID
General Specs										

FIGURE 13. Group devices type subfield composition.

Bits: 5	3	Variable	5	5	3	1	32
Wireless technology ID	Architecture ID	Encoding of cluster heads IDs	MAC type ID	Routing type ID	Data reporting pattern ID	Single channel or multi-channel	Bitmap of total allowed frequency channels

FIGURE 14. PAN logical topology subfield structure.

Octet: 1		1	1		Variable								
Bits:4	4	8	4	4	0/32	0/32	0/32	0/32	According to the number of parameters and their values lengths				
Version	Reserved	Transmission options	Aggregation Type (no aggregation, sum, average, min/max, variance, etc.)	Aggregation source type	Min. Lat./source Lat./ Device Lat.	Min. Lon./source Lon./Device Lon.	Max. Lat.	Max. Lon.	Parameter1 name	Parameter1 value	Parameter2 name	Parameter2 value	....
			Aggregation info.										
Frame control			Payload control		Payload								

FIGURE 15. Data message frame format.

commercial capabilities or the capabilities' combinations, each of them refers to a specific device model identified by unique Model or Spec ID.

The composition of the Group devices type subfield of the PAN physical topology subfield is shown in Fig. 13; the first byte is a bitmap determines general specification of the devices, where the first bit categorizes the device according to the main two categories indicated before: standalone and embedded/integrated. The Device/Thing type ID subfield identifies the device subcategory for standalone devices (sensor and actuator nodes, drones, etc.) or the monitored/controlled thing subcategory for embedded/integrated devices (sprinklers, fertilizer injectors, fans, heaters, etc.).

The PAN logical topology subfield, shown in Fig. 14, includes identifying the PAN architecture such as flat, hierarchical, and location-based; the reporting pattern such as event-based, time-based, query-based, and hybrid; and in case of hierarchical architecture, an encoding specifies the number and IDs of coordinators/cluster heads via bitmap spans over a number of bytes depends on the PAN devices' number where the bits are corresponding to devices' IDs in ascending order starting from the MSB - also any other encoding method for such information can be used.

The message exchange between the Gateway/IoT Devices and the Cloud also contains Data and Feedback upward messages and Command messages sent downward from the Cloud. Fig.15 shows the frame format of a Data frame published to the topic "AllParameters". The Gateway publishes a Data frame per Device/Thing contains all the data the Device/Thing has. Any application module interested in a specific data related to a Device(s) subscribes to the appropriate topic under the Data branch, and the Pub/sub distributor module at the Cloud will receive the Device's data and publish again each parameter separately to its corresponding topic with other common information such as the Device/Thing ID. The Aggregation source type sub-field determines the scope of the aggregated data, for example, a cluster head collected data, or data in a specific area identified by its maximum and minimum coordinates. The remaining part of the Data frame is a JSON key-value pairs for the sensed phenomena and other parameters such as power level, path length, parent ID, parameters and hash values required for offline analysis of the underlying local devices network behavior.

The application module which requires to send a command to the devices or offload some of its processing and analytics to a Gateway, sends a Command frame as shown in Fig. 16, identifying the requested standard operation by its standardized ID and data format. The commander application module generates a random number or name, puts it in the dedicated frame subfield and subscribes with it as a temporary topic

Octet: 1		1	Variable			
Bits:4	4	8	16	16	16	According to the content data length
Version	Reserved	Transmission options	CMD/offload request ID	Intended Device/Group ID	Random number for marking the CMD/offload request	CMD/offload request content
Frame control			Payload			

FIGURE 16. **Command message frame format.**

Octet: 1		1	Variable	
Bits:4	4	8	16	According to the event data length
Version	Reserved	Transmission options	Event ID	Event data
Frame control			Payload	

FIGURE 17. **Feedback message frame format.**

under the Confirmations level, or the StoredData level if the command is requesting stored data retrieval from a Gateway. Through these temporary created channels, the commanded Gateway or IoT Device will respond with the command execution confirmation (with optional frame payload) or the requested data in the frame payload, respectively (the data may be requested by a query command, in this case the confirmation is a reply with a Data frame contains the requested aggregated or unaggregated data).

The standard events and notifications generated from the local devices network, the Gateway, or the IoT Device are reported to the subscribed applications using the format shown in Fig. 17.

2) THE CLOUD TIER

The layers constitute the Cloud tier structure are not strictly up-down stacked above the IP stack as shown in Fig. 18, rather, each layer is interfaced to all the other layers. The Transport layer facilitates the connection among the Internet things whether they are sensing and actuating devices or Cloud modules or external applications allowed to publish their information and get some farms' information. The Service layer contains modules for libraries and frameworks needed by other layers for adding intelligence, numerous generic and agricultural customized visualizations and reporting, and libraries register the standardized definitions and the corresponding management libraries. The Information and data management layer is responsible for the tasks related to storing, retrieving, and processing all modules-related data types. The Application layer contains modules implement the core functions of the smart agriculture system.

The Application modules can be nested modules. In Fig. 18, the module frame can be an outer frame groups sub-modules perform different tasks related to its general function; from this, the mentioned Application layer modules in Fig. 18 contain such modules:

- Farming and cropping systems, crops, seeds, livestock suggestion (FCSLS)
- Farm layout suggestion (FLS)
- Animal husbandry, housing, and care needs formulation (AHCN)
- Defining a soilless culture solution and the appropriate CEA level considering different options and the enclosed farming system case (CEASC)
- Defining appropriate CEA level for a greenhouse (CEAGH)
- Defining appropriate CEA level for a post-harvest storage place (CEAST)
- Defining a land preparation solution (LPS)
- Defining planting and crop organization solution (PCOS)
- Defining soil fertilization solution (SFS)
- Defining pest control solution (PCS)
- Defining irrigation and drainage solution (IandD)
- Field configuration monitor (FCM)
- Generate field phenomena spatial variation maps (PSVM)
- Generate variable-rate seeding prescription map (VRSEM)
- Generate fertilizer variable-rate application prescription map (VRFM)
- Generate spray variable-rate application prescription map (VRSPM)
- Generate nutrient variable-rate application prescription map (VRNM)
- Generate lime and gypsum variable-rate application prescription map (VRLGM)

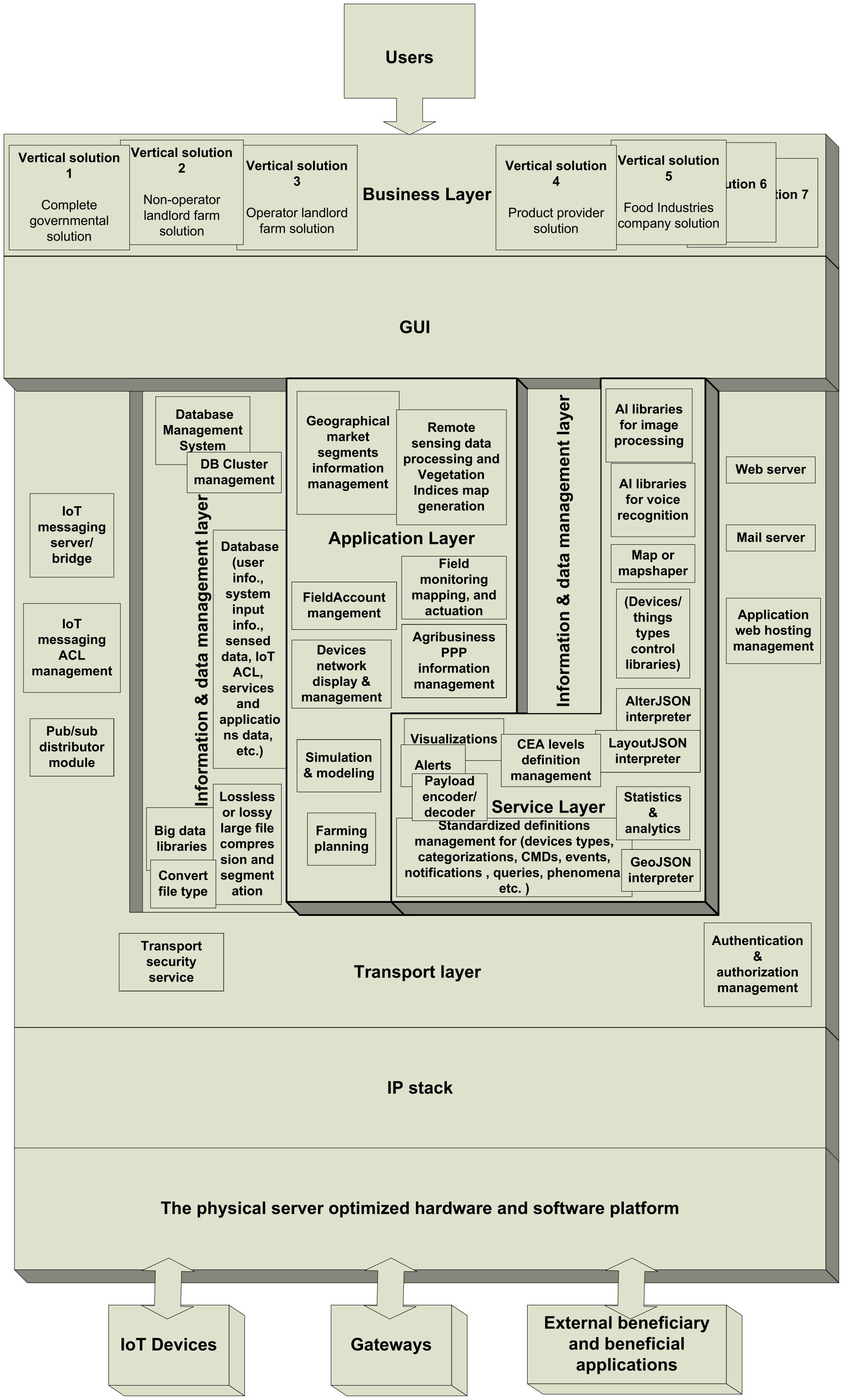

FIGURE 18. AITRA Cloud tier structure.

- Generate manure variable-rate application prescription map (VRMM)
- Generate site-specific variable rate irrigation prescription map (VRIM)
- Generate yield map (YM)
- Generate gross margin map (GMM)
- Drones control and monitor (DCM)
- Vehicle tracking (VT)
- Vehicle guidance system (VGS)
- Remote control of sprinkler irrigation network (RCSI)

- Remote soil salinity management (RMSS)
- Remote electronic pest control devices (RCPD)
- Yield prediction (YP)
- Yield variation diagnosis and long term decisions (YVDLTD)
- Crop simulation models (CSMs)
- Crop scouting (CS)
- Emulation for best count and placement of the devices in open fields with nodes assembly facility (BDNP)
- Reconstruction of the per-packet routing paths in devices network (RPR)
- Devices network display (DND)
- Devices network clustering and tree construction (CTC)
- Devices network communication technology parameters setting and control of operation (CCO)
- Multiple field solutions (MFS)
- Airborne-based mapping and data processing (ABMP)
- Satellite-based mapping and data processing (SMP)
- Unmanned Aerial Vehicle (UAV)-based mapping and data processing (UAVMP)

The user interface (GUI) is used by the application developers, and it is made to facilitate, accelerate, improve the application development, benefiting from the underlying standardization.

The design wizard of the solution will mainly appear as producer objects and consumer objects connected in some way. To link modules to the solution, they are instantiated (this means more than one instance from the same module can exist in the same solution) by dragging their representative icons into the design wizard of the solution's webpage intended to contain them. The instantiation means the Library's scripts are added to the solution's web application and its HTML representation is included in the specified webpage, conforming with the requested style, and functioning according to the input to the Library's module object. Library instantiation also may mean that the Library performs a specific mediatory task needed by the other parts of the design. With respect to the Application module, the instantiation implies the Application module is included in the solution, or in better words, is called to perform its task, may be on a supplied input, and it outputs the result to the connected consumer object.

Each Application module has a Module Access Point (MAP) through which it receives the request primitives, an output channel instantiated for connection to more than one consumer object, and an input channel can be connected to another module's output channel or generally to more than one producer object's output funneled to it; as a result, the required output may be produced by recursive calls in a chain of connected modules. It is not necessary for all the input and output channels to be connected, they can remain inactive. These are some examples of defining modules' different interfaces:

- Farming and cropping systems, crops, seeds, livestock suggestion (FCSLS)

- Request_Call_primitive {JSON object of input fields' values of the outer module "Farming planning"}
- Input channel {inactive or connected to one of the Simulation & modeling sub-modules}
- Output {AlterJSON object: standardized lightweight human readable Data Interchange Format for describing the determinants of prioritized alternatives for something}
- Output channel {connected to AlterJSON interpreter for visualizing the different alternatives or instantiated for connection to other modules' input channels}
- Subscriptions {AgriIoTFM/EconAndFIData/GeoMarketSegments/+/InputMarkets, AgriIoTFM/EconAndFIData/GeoMarketSegments/+/OutputMarkets, AgriIoTFM/AgribusinessPPP/+/FinancingAndLoans }

- Farm layout suggestion (FLS)

- Request_Call_primitive {JSON object of input fields' values of the outer module "Farming planning", an AlterJSON object}
- Input channel {inactive or connected to FCSLS module}
- Output {LayoutJSON object: standardized lightweight human readable Data Interchange Format for describing places layout}
- Output channel {connected to LayoutJSON interpreter for visualizing the layouts or instantiated for connection to other modules' input channels}
- Subscriptions {AgriIoTFM/EconAndFIData/GeoMarketSegments/+/OutputMarkets, AgriIoTFM/AgribusinessPPP/+/Transportation, AgriIoTFM/AgribusinessPPP/+/Logistics, AgriIoTFM/AgribusinessPPP/+/LaboratoriesAndNurserybeds, AgriIoTFM/AgribusinessPPP/+/ElectricityWaterDrainageSupply }

- Generate field phenomena spatial variation maps (PSVM)

- Request_Call_primitive {Type (real-time or historical), phenomenon identification, field name, timer for real-time, date and time for historical}
- Input channel {inactive or, if composite phenomenon, it is connected to the output channels of other PSVM modules}
- Output {GeoJSON object}
- Output channel {connected to map or instantiated for connection to other modules such as PSVM modules and VRIM modules}
- Subscriptions {AgriIoTFM/Data/*field name*/+/*the remaining sub-tree for the intended phenomenon*}

- Generate site-specific variable rate irrigation prescription map (VRIM)

- Request_Call_primitive {GeoJSON object(s) for the phenomena spatial variation}
- Input channel {inactive or connected to the output channels of PSVM modules for soil water content and temperature}
- Output {GeoJSON object}
- Output channel {connected to map or instantiated for connection to another suitable consumer object such as

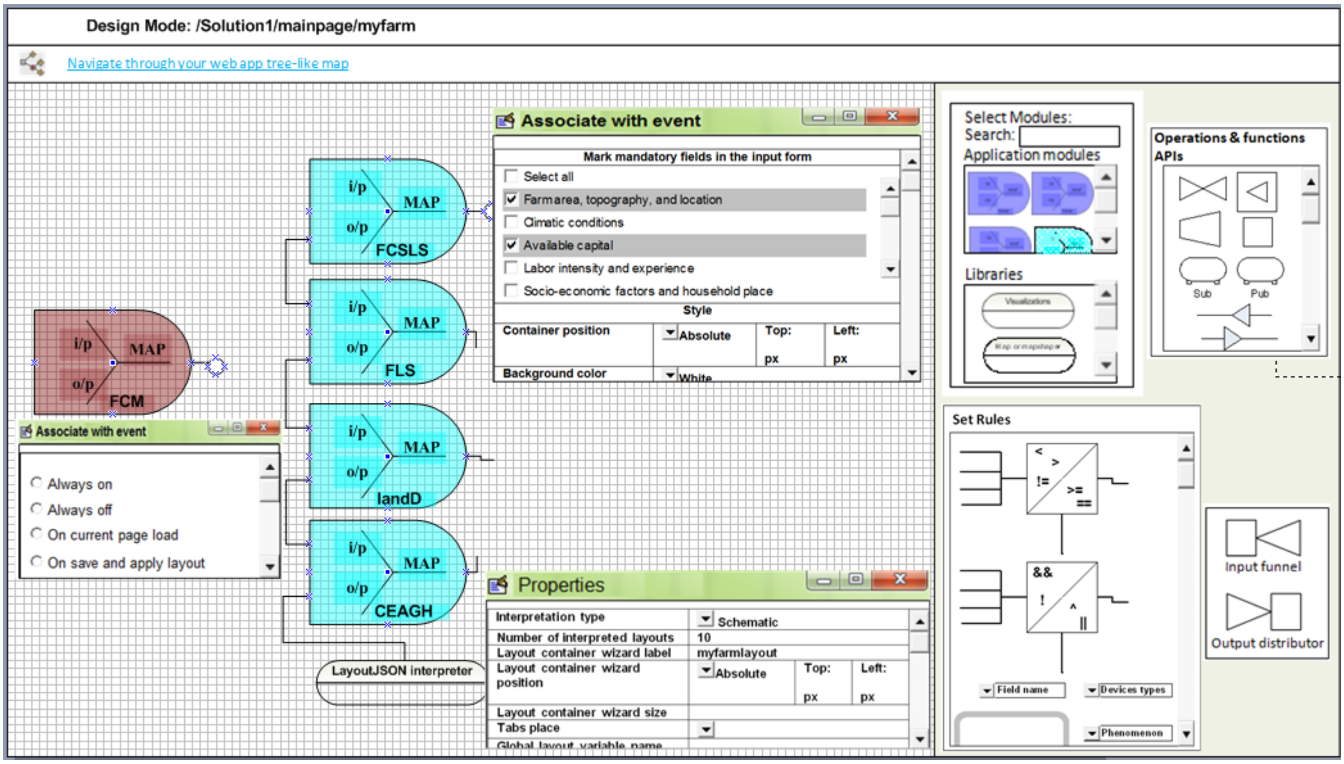

FIGURE 19. AITRA platform's design wizard for "Layout" page.

the input channel of Remote control of sprinkler irrigation network module}
- Subscriptions {N/A}.

The values of the input fields of a parent module are usually prompted in the request primitive to any of its sub-modules. The modules acquire their other required data by querying the database or subscribing to topics published by external entities through the IoT messaging, and they appropriately store the acquired data in the database.

New modules and new functions with appropriate APIs, new items in existing libraries, and modifications to the implementation of existing modules can be registered with the architecture as long as they are conformed with the conventions of the architecture. Any device communicates with the defined gateway structure or generally uses the proposed IoT architecture's frame structures and topic tree can be integrated with the architecture after registering its Device type ID and its management library.

3) THE BUSINESS TIER

The Business tier is where the AITRA users and their applications reside, whether the user is an application developer or company wants to build its specific product with special characteristics on the top of the AITRA platform and put it on the market for the end-users, a government customizes a national system of course connected to the AITRA-based foreign systems and compatible external applications, an individual uses the platform to formulate a suitable solution to his needs, etc.

The AITRA platform users communicate to the platform Cloud tier through its GUI layer. To elaborate more on the Cloud tier structure and how its platform can be implemented and used in the Business tier, this section is going to give a heuristic illustrative example.

Suppose an operator landlord has a new farm land and wants to use the smart technologies and PA to manage it. He decided to use by himself the proposed architecture to facilitate achieving this.

After creating an account on the platform, he started to create his solution's web application under the name "Solution1". Just after his solution project had been created, he was prompted to define its web application tree-like map, and he defined two pages branched from a main page. He decided to dedicate the first page to use the Farming planning sub-modules to give him assessed alternatives for some farming practices and layout. Therefore, in the design wizard of the first web page, he placed four sub-modules in a chain starting from FCSLS followed by FLS and then two modules for suggesting irrigation solutions and the CEA level for any incorporated greenhouse. All the suggestions are embedded in the LayoutJSON file that exits from the chain, and he used the ready-made LayoutJSON interpreter to visualize the different alternative layouts; the design wizard of the Layout page is shown in Fig. 19, the right click menus are shown beside the clicked objects.

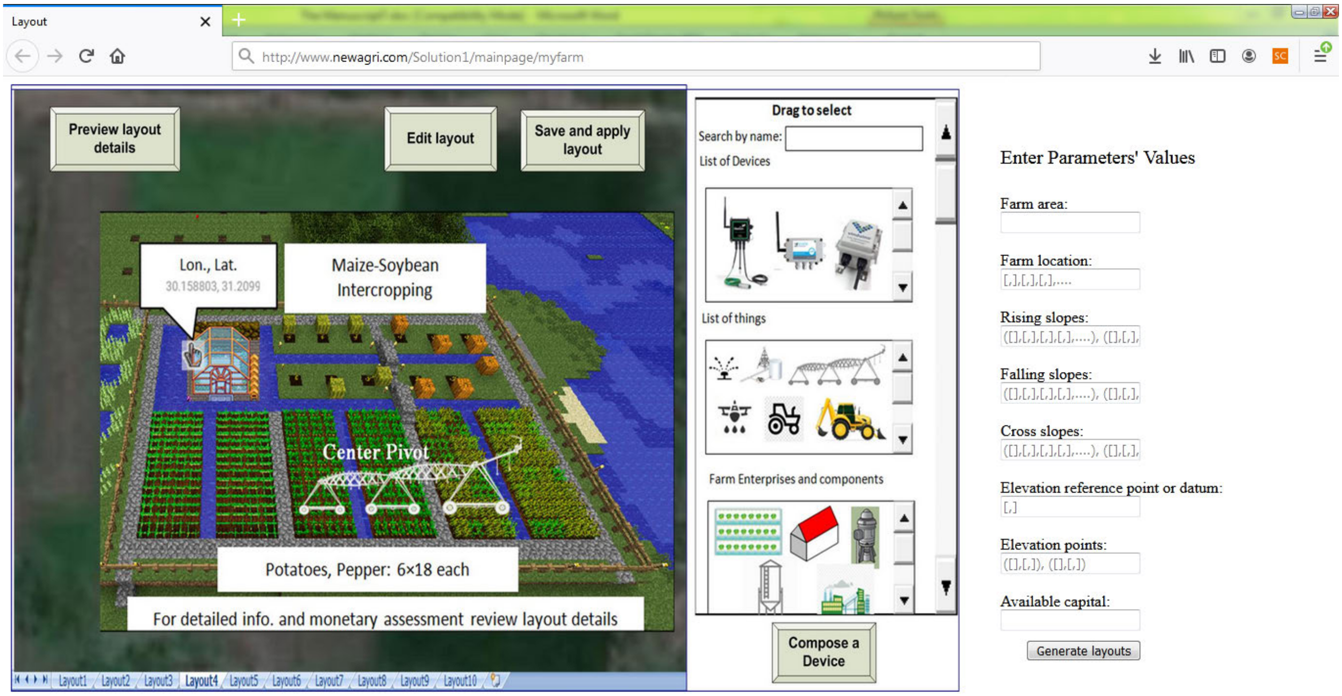

FIGURE 20. **The Layout page.**

Also, he placed an instance of the FCM module, setting it to be always on so that it always, besides its other functions, uses its subscriptions to "AgriIoTFM/Configurations/*Field name*" and "AgriIoTFM/Data/*Field name*/+/Coordinates" to update the selected layout saved in the layout global variable of the specified field name.

He stopped to design at this stage and selected to use the designed Layout page (Fig. 20) to plan his farm and prepare the required enterprises first. His choice fall on one of the suggested layouts without any editing; it consists of one greenhouse and uses one center-pivot irrigation system. In addition to that, he was going to deploy a number of sensor nodes in his farm to monitor its conditions in real-time and aid in developing a precision irrigation function to remotely control the center-pivot.

The design of the second page is shown in Fig. 21. As it appears from Fig. 21, he used the PSVM module for temperature and soil water content and the VRIM module to generate the variable rate irrigation prescription map, then he converted the file type, compressed and sent it to the center pivot using its appropriate command; he associated these actions with a button pressing.

Another feature of the "moncont" second page, is the usage of the Visualizations library to display a time-series line chart graph for the average value of the soil water content phenomenon of all nodes. Finally, he utilized the Set rules menu to define a rule generates a Dialog box/popup window Alert when the measurement of the soil moisture content taken by a specific device type exceeds a limit of 70. The "moncont" page is looked like Fig. 22.

As shown in the illustrative example, in a user-friendly way with few drag and drop actions and right click understandable settings, even the ordinary user can select from the rich set of exposed functionalities and finalize his effective solution in a short time.

VIII. PROOF OF CONCEPT (POC) EXPERIMENT

Apart from the heuristic illustrative example given previously for an AITRA-based business solution and to interpret more the basic concepts of the proposed design ideas and assess its feasibility, this section is going to introduce a proof of concept based on simple implementation of the AITRA platform includes some of its main features and functionalities. First, a description of the need, which the derived solution should tackle, is given; then the materials, tools, and technologies used are indicted. The scenario of interactions and procedures is then illustrated and their results are shown.

A. BUSINESS NEED DESCRIPTION AND ASSUMPTIONS

A management authority of a city called, AlexoCity, intends to smartly mechanize and centralize its farms management. It wants to have a solution allows for any authorized farm within its range to be subjected to the solution management. The farm will be equipped with a gateway and different groups of smart devices sense some soil parameters; some farms will actuate the operation of spraying the soil with the appropriate organic matter for increasing soil cohesion to control wind erosion. The city is interested in both the farm sensed phenomena data and the devices network information, where it wants to display the phenomena values and

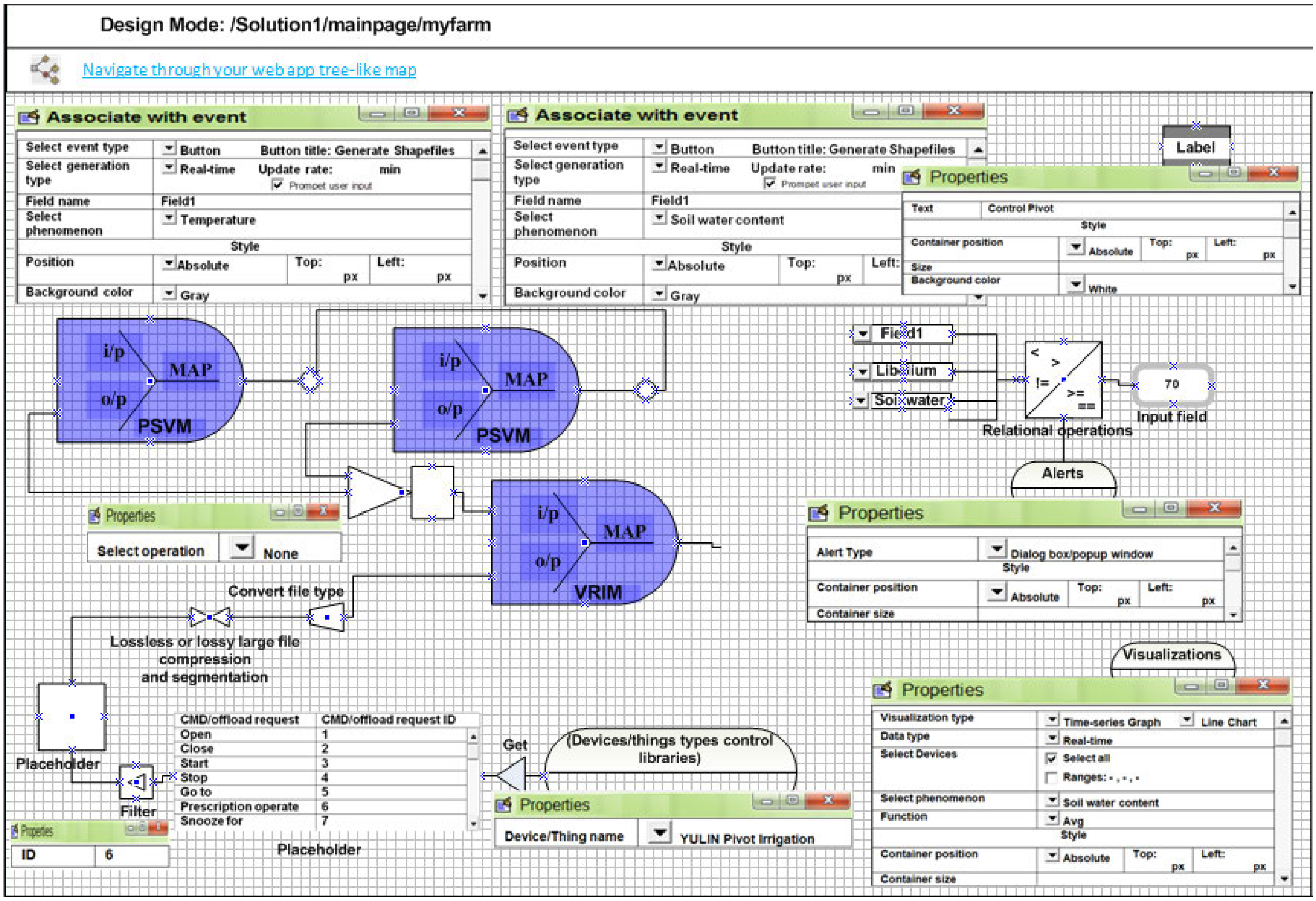

FIGURE 21. AITRA platform's design wizard for "moncont" page.

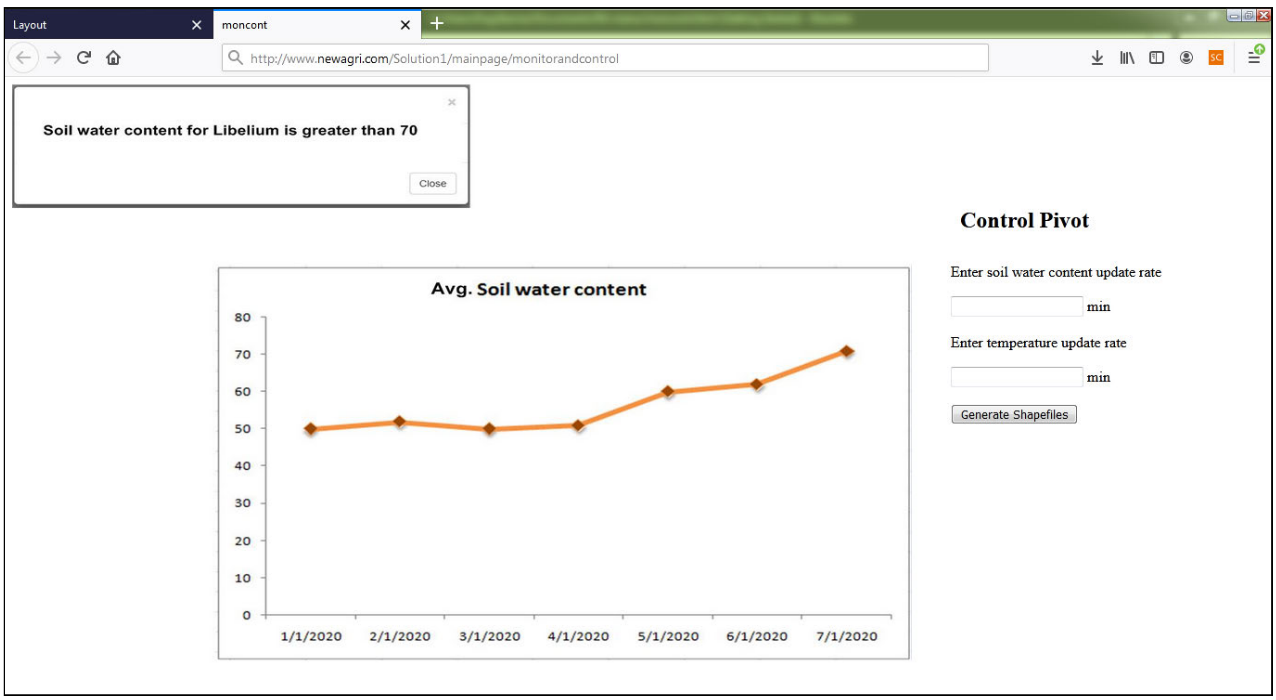

FIGURE 22. The moncont page.

the devices connection information to ease network issues debugging and parameters settings. It wants the spraying actuation to be a closed-loop control process relaying on the measurements of an external weather station. It wants to

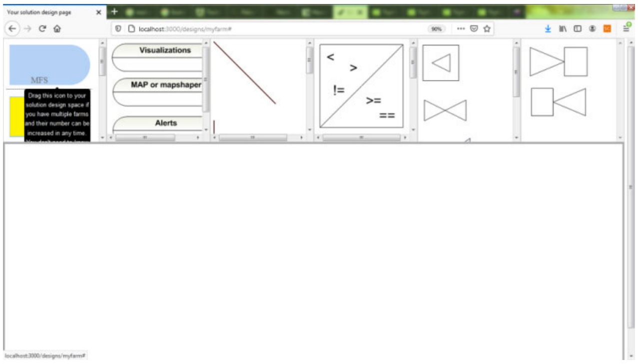

FIGURE 23. **The design page of the AITRA solution; tooltips on the components describe them and illustrate their usage method.**

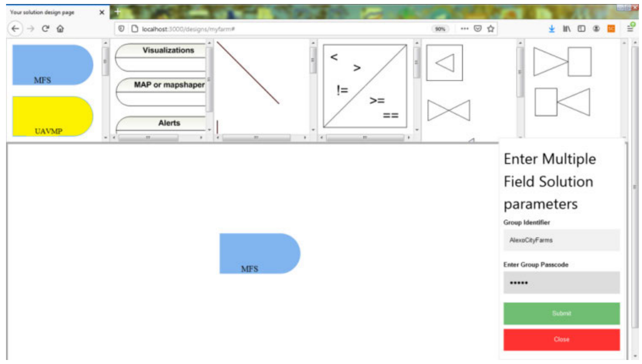

FIGURE 24. **The design page of the AITRA solution; using an MFS instance and setting its required parameters.**

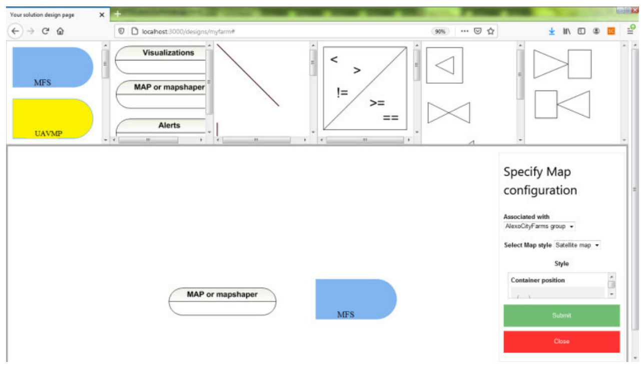

FIGURE 25. **The design page of the AITRA solution; using MAP library and setting its required parameters.**

dedicate a specific page for each farm to publish its needs and gets immediately an optimal set of diversified options meet its need.

B. KEY ENABLING TECHNOLOGIES AND TOOLS

The POC used mainly Node.js [35] for Cloud and Business tiers implementation besides other technologies and tools such as HTML, CSS, JavaScript, Mapbox [36], and WebSockets. The MQTT protocol is used to implement the publish/subscribe transport infrastructure for the AITRA standard communication protocol. The Paho MQTT C Client Library [37] is used to implement the Device tier and the external applications.

C. PROCEDURES

The person, who assigned by the city to develop the solution, initiated an AITRA solution contains one webpage, related it to the city by identifying its boundary coordinates. First, he separated each need of the city, and used only the tooltips provided by the platform when hovering on its components to select the appropriate component(s) (as shown in Fig. 23) for meeting each need and learn how to use it.

He selected the MFS module to help in adding any farm in the city automatically without inputting specific farm(s) data. Then as shown in Fig. 24, he clicked the module instance in the design space to enter the fields' group name that will be used in the topic-tree routing as the group identifier "AlexoCityFarms", and the required Configuration message passcode that will be used to authenticate the authorized fields/farms' network/devices. Upon the submission of this information, the MFS subscribes to "AgriIoTFM/Configurations/AlexoCityFarms/#" and the IoT messaging ACL management module puts the rules that prevent unauthorized subscriptions. The group identifier is saved and made globally available to all the modules, the same for the identification names of the authorized farms when registered. Similarly, all the related lists and drop-down menus are populated automatically with their names and the farms' devices models, and all the solution's module instances get notified when a new farm is registered.

He dragged the MAP library to include the digital mapping service in the solution and clicked it to specify its settings: the map style, what field(s) it is associated with, and what its display format is (Fig. 25).

Now, it is the time to start to specify what to be displayed in the included map: its layers and functions. He put an instance of the DND module and connected its output to the MAP library to display/update the areas' boundaries and the deployed devices of each field on the map.

To display the phenomena values, he incorporated a Visualizations library and set its parameters to display the required phenomenon values come from any device in any field in real-time in text format; and specified the output format style. He used the RRR module to reconstruct the per-packet routing paths and draw the devices' logical connections on the map (see Fig. 26).

He used the Weather station finder (WSF) module to search the standalone IoT weather stations exist in the system under the level "AgriIoTFM/Configurations/weatherstations" and find the appropriate nearest one to be incorporated in the solution. Once found, in this case "Centeral_lab_WS", the WSF subscribes to "AgriIoTFM/Data/Centeral_lab_WS/Climate/windspeed".

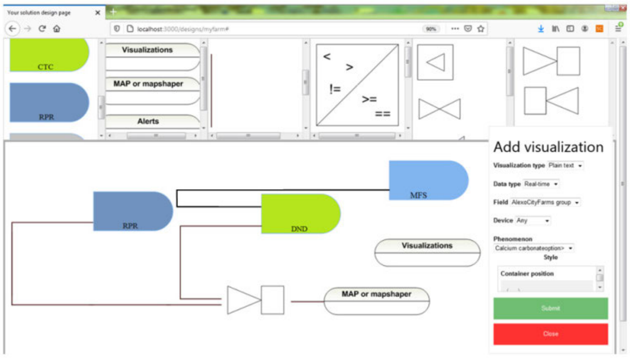

FIGURE 26. **The design page of the AITRA solution; adding more modules' instances, libraries, and connections.**

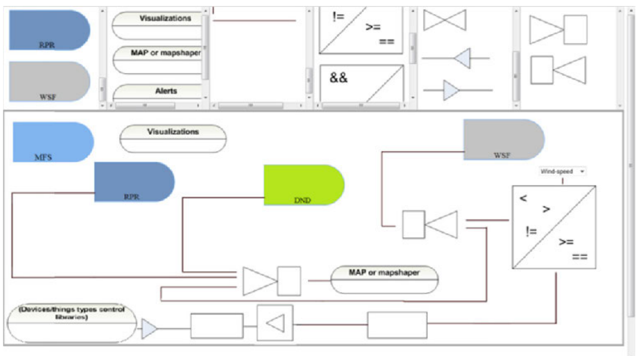

FIGURE 27. **The design page of the AITRA solution; adding modules and libraries to achieve the closed-loop control of all the registered farms' spraying systems.**

As the WSF has two outputs: a layer representation of the weather station position on the map and its sensed phenomena values, its output is distributed to the MAP library as well as a relational rule its inputs and parameters are set to detect the increase of the wind speed value over a certain value. This event is placed as a trigger for the command sent to the fields, which have a sprayer actuator system, to start its operation (see Fig. 27).

The Devices/things types control libraries instance is populated automatically with a list of all the available devices models in the solution and its information; and the required command for each model can be filtered by its ID. Since, until that time no field has been registered on the solution, the filter is set using keywords should by convention be incorporated in the description of all commands do the same thing for different devices models. In this case the keywords are operate, spray, and organic (as till now the standard description of the devices specification has not been defined yet, in the POC, the commands are sent directly to the fields when the test condition happed without using the Devices/things types control library and filtering).

He used the FieldAccount management module to automatically generate webpages dedicated to each farm protected by the link and credentials sent to its AITRA email.

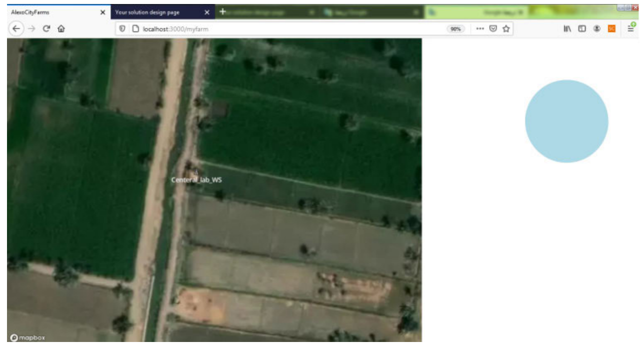

FIGURE 28. **The AITRA solution webpage; the map navigated to the location of the selected weather station.**

By doing so, the fields can benefit from the AITRA economic network and services.

Finally, just the current passcode needs to be confidentially delivered to each field wants to register on the solution.

D. RESULTS

Up to now, the solution webpage resulted from the previous steps appears like that is shown in Fig. 28, where just after the WSF was added to the solution and connected to the map, it performed its task and the map flew to the location of the selected weather station.

In the following, the results of the registration of two fields will be shown. The two farms are named Weast_Farm and Vegy_Farm. First, the operator in each farm uses the Gateway interface to enter the required inputs (an initial name for the Gateway, the passcode, the farms group identifier as the higher topic level name "AlexoCityFarms", the cloud server URI, the farm area coordinates, PAN physical topology, some PAN logical topology, set the contact type to indicate email), then it commands the Gateway to start work. Fig. 29 through Fig. 32 represent message sequence charts at the application level followed by each interaction result at the Business tier level. The payload and payload control fields are only considered, and for simplicity only the Model or Spec ID from the PAN physical topology, and the Architecture ID from the PAN logical topology are indicated.

Just after the authorized configuration is received (Fig. 29 and Fig. 30), the map flew to the field location, displayed its area and devices, and a link with the field name for easy navigation is created. The devices' model specification can be known by hovering in each device. The Weast_Farm contains seven devices divided into two groups and construct a flat network while Vegy_Farm contains only one group of five devices form a star topology with the Gateway. The indicated number of devices are placed at random locations updated later to real locations when the data messages are received from the GPS-enabled devices.

In any time the visualization settings can be altered, for example, after some farms are registered, the user can select

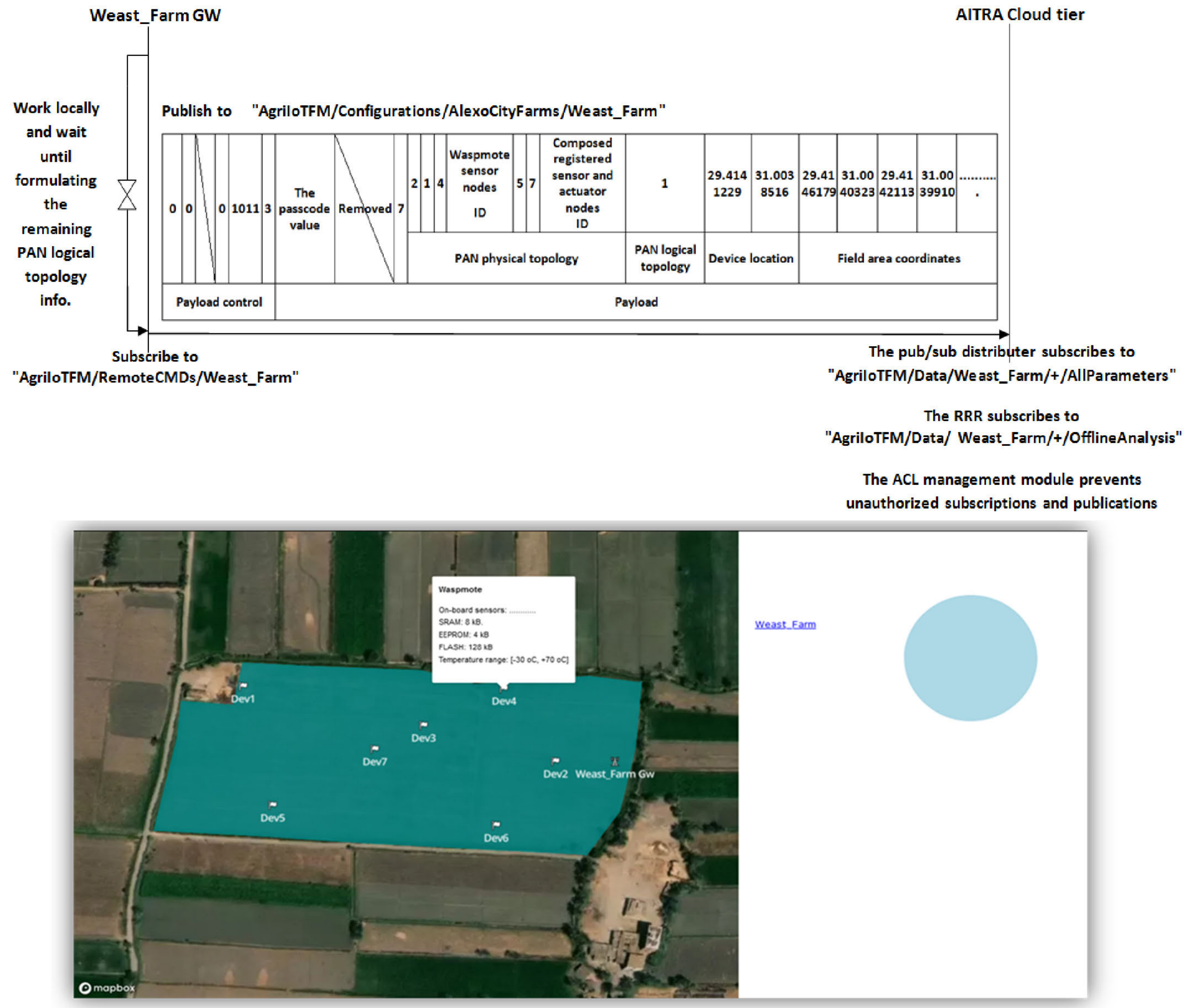

FIGURE 29. The start of Weast-Farm gateway work, the sending of the Configuration message, and the result on the solution's webpage.

to display only the phenomenon value of one field, this entails altering the subscription towards that selected field (e.g., "AgriIoTFM/Data/Vegy_Farm/+/CalciumCarbonate").

The chart in Fig. 31 represents the start of data sending by the devices. It is accompanied by the display of the phenomenon value, and the reconstruction and display of the networks routing paths.

Fig. 32 illustrates the sending of a command to all the registered fields. When the field executes the command successfully, its Gateway sends a positive confirmation to the application upon which a text confirmation is displayed to the user on the screen.

In the webpage of each farm (see Fig. 33 and Fig. 34), the operator of the farm can formulate his farm need and submit it to be published by the FieldAccount management module to the topic "Need" under the "AgriIoTFM/FieldAccount/" level for distributing it to all the interested applications, in the same time, the module subscribes to the corresponding level of the farm need under "AgriIoTFM/EconAndFIData/GeoMarketSegments/*segment name*/InputMarkets" in this case the topic name is "Pesticides".

The applications match the need with what they offer and publish they corresponding offers to the topic "Pesticides" where they are received by the FieldAccount management module and undergo an analytics process to extract the most relevant and optimal options to the farm need, then displays them on its webpage.

Also, as shown in Fig. 34, the farm can advertise its outputs for sale by publishing it, in this case, to the topic "AgriIoTFM/EconAndFIData/GeoMarketSegments/*segment name*/OutputMarkets/Grains".

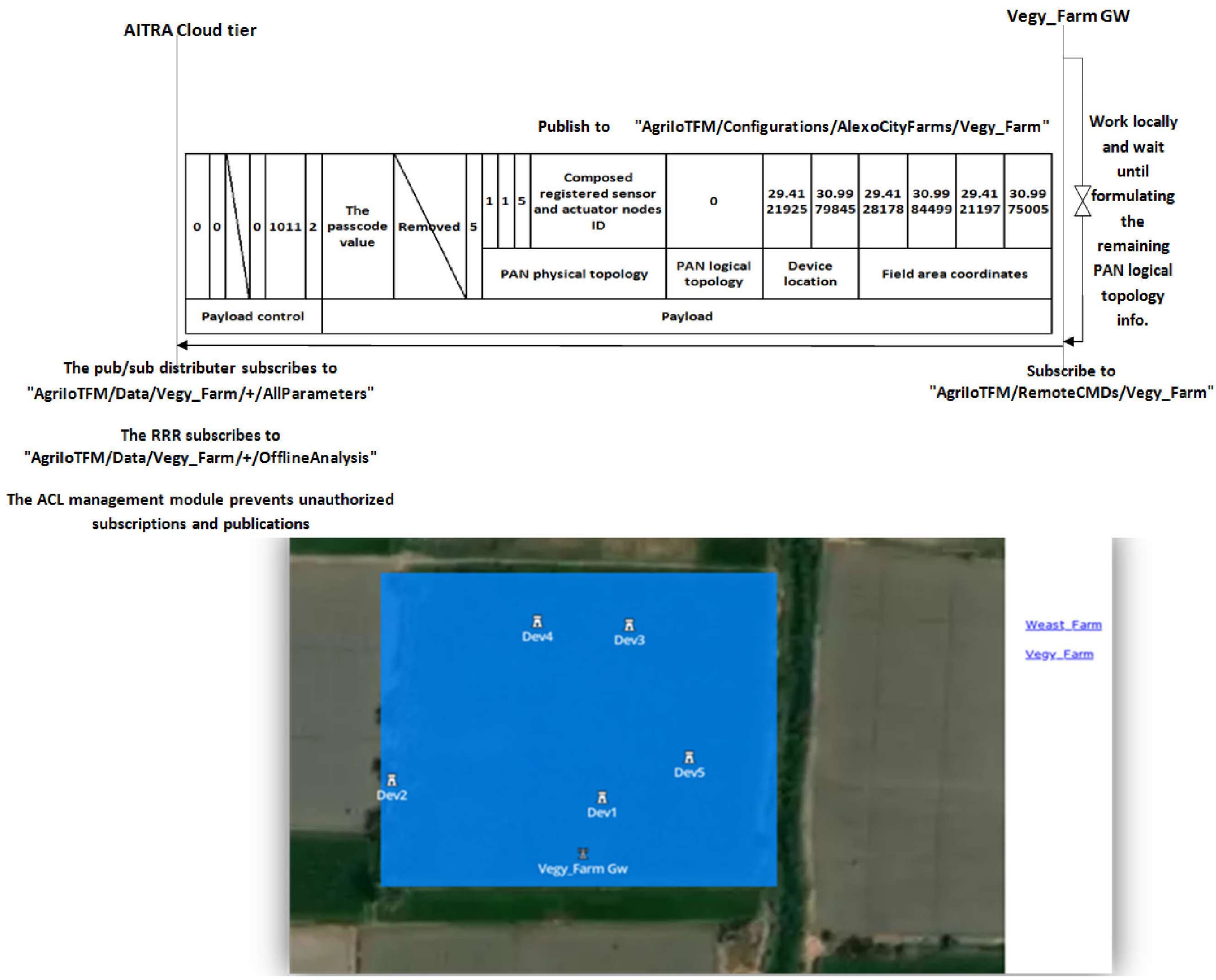

FIGURE 30. The start of Vegy-Farm gateway work, the sending of the Configuration message, and the result on the solution's webpage.

IX. DISCUSSION

This paper considered the stated IoT reference architecture design problem and tried to propose a design for a standardized Agricultural IoT Reference Architecture - AITRA - with good features. Section II reviewed some of the existing IoT reference architectures with different types: generic, industrial control application dedicated, and smart farming application dedicated, and identified some of their important features; Table 11 characterizes AITRA with respect to the same features.

Each one of the reviewed reference architectures (RAs) is built from different number and types of horizontal and cross-cutting layers (it may use different notation than layer such as functionality group or functional domain). One of them, IoT-A, employs the cross-layer design approach which doesn't follow a strictly layered interfaces but the layer can have more than two neighbor layers bi-directionally communicate, exchange information and notifications and use the services offered by each one, and consequently may be designed jointly for better design and overall system performance.

As a natural consequence of not including adequate analysis of the IoT application ecosystem in the reviewed RAs design, these RAs are represented with high levels of abstraction even the application-dedicated architectures. WSO2 and IoT-A are the most abstracted RAs as they are generic and incorporate general architecture principles which are not enough for actual guidance. However, it is worth noting that the IoT-A proposed an IoT Domain Model which helps in capturing the main concepts and the relationships that are relevant for IoT stakeholders thus structuring a concrete application scenario using common language which additionally provides common understanding of the main concepts of the target domain facilitating the comparison and evaluation of its concrete applications.

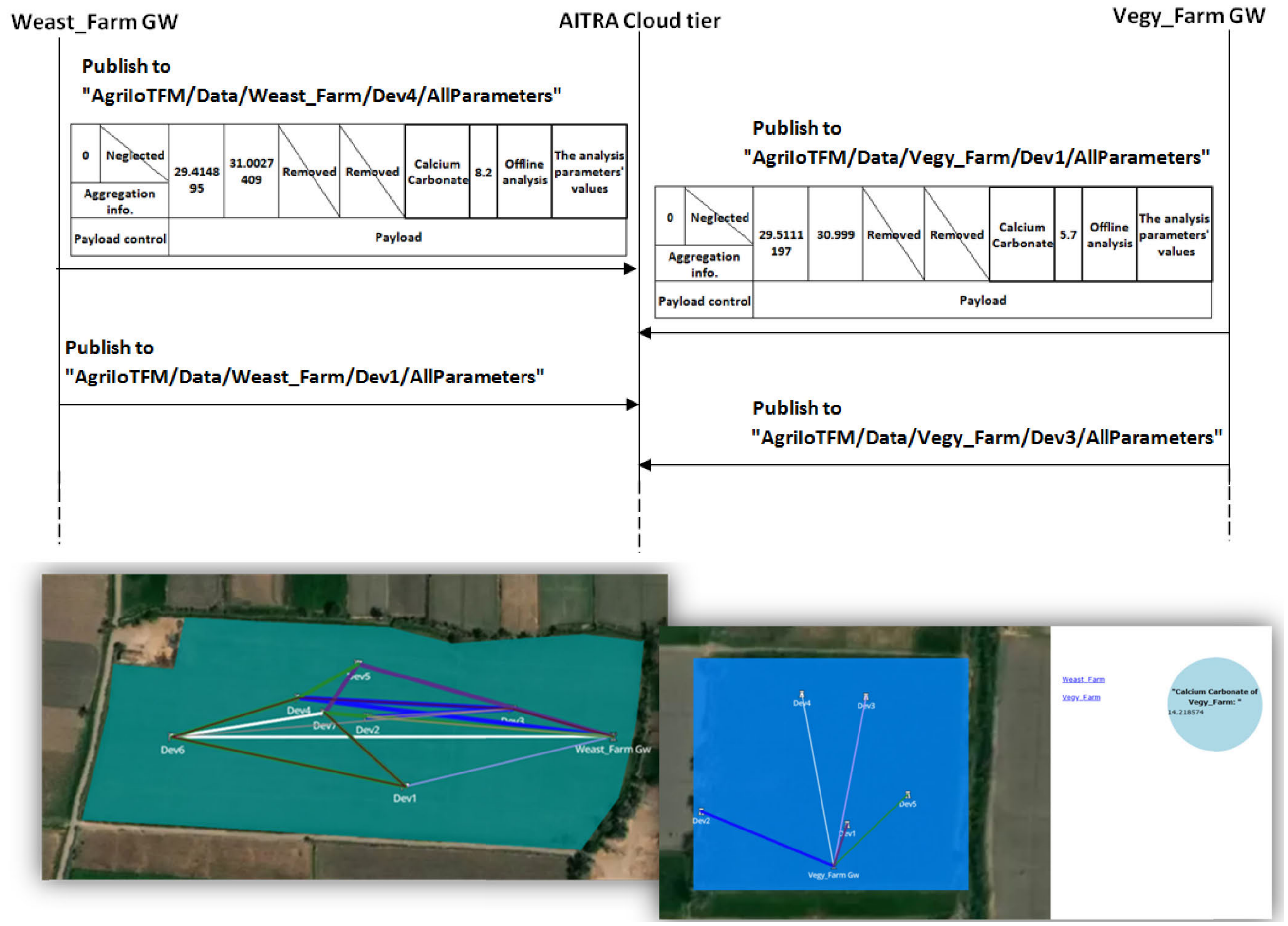

FIGURE 31. Message sequence chart of the Data messages are being sent by the two farms, and an illustration of the result on the solution's webpage.

The IIRA resorted to decompose a typical IIoT system into Functional Domains (FDs) have wide existence in the industrial applications, specify their relations, and use them to specify the main functionalities that should be incorporated in its derived solutions. It defined these FDs and further decomposed them into a set of common functions, and performed crosscutting functional analysis by suggesting some possible enabling crosscutting for the main system functions. This resulted in a lower level of abstraction than WSO2 and IoT-A, but it still guides only for a generic IIoT system, leaves for the architect a lot of work to do further analysis to meet the specific use case requirements, especially that the industrial sector is a very wide domain spans over a lot of the application fields - agriculture, medicine, mining, etc.

The IoT-based FMIS addressed a tighter application domain, even though, the conducted domain analysis is not adequate to facilitate for the architect producing concrete solutions, and the lower layers of its proposed architecture need to be decomposed into specific functions suitable to the application.

The IoT-based FMIS doesn't consider a specific type for the interface to the users and developers. WSO2 exposes only the API type which is suitable only for persons with programming background as well as it takes time and effort to be learned and practiced. IIRA provides, in addition to the API, the UI type which is more convenient for ordinary persons even for the programmers themselves, but the UI includes the Command Line Interface (CLI) which still needs some skills in dealing with it. The IoT-A concentrates on the conceptual interfaces approach which helps the user to stay at a conceptual level of the whole process without being involved in its steps complexity where the interface is built based on a conceptual model of the process, thus the integration cost is reduced. Due to their high level of abstraction, these RAs also expose generic web portal and dashboards interfaces to the users or only suggest some possible applications-specific interfaces.

All the reviewed RAs as well as AITRA are vendor neutral, modular, support external application interaction, consider M-to-M and M-to-application communication patterns, and support different types of security.

AITRA is characterized by specialization which means accuracy, efficiency, productivity, establishing normative standards, lower cost, and many other advantages with the

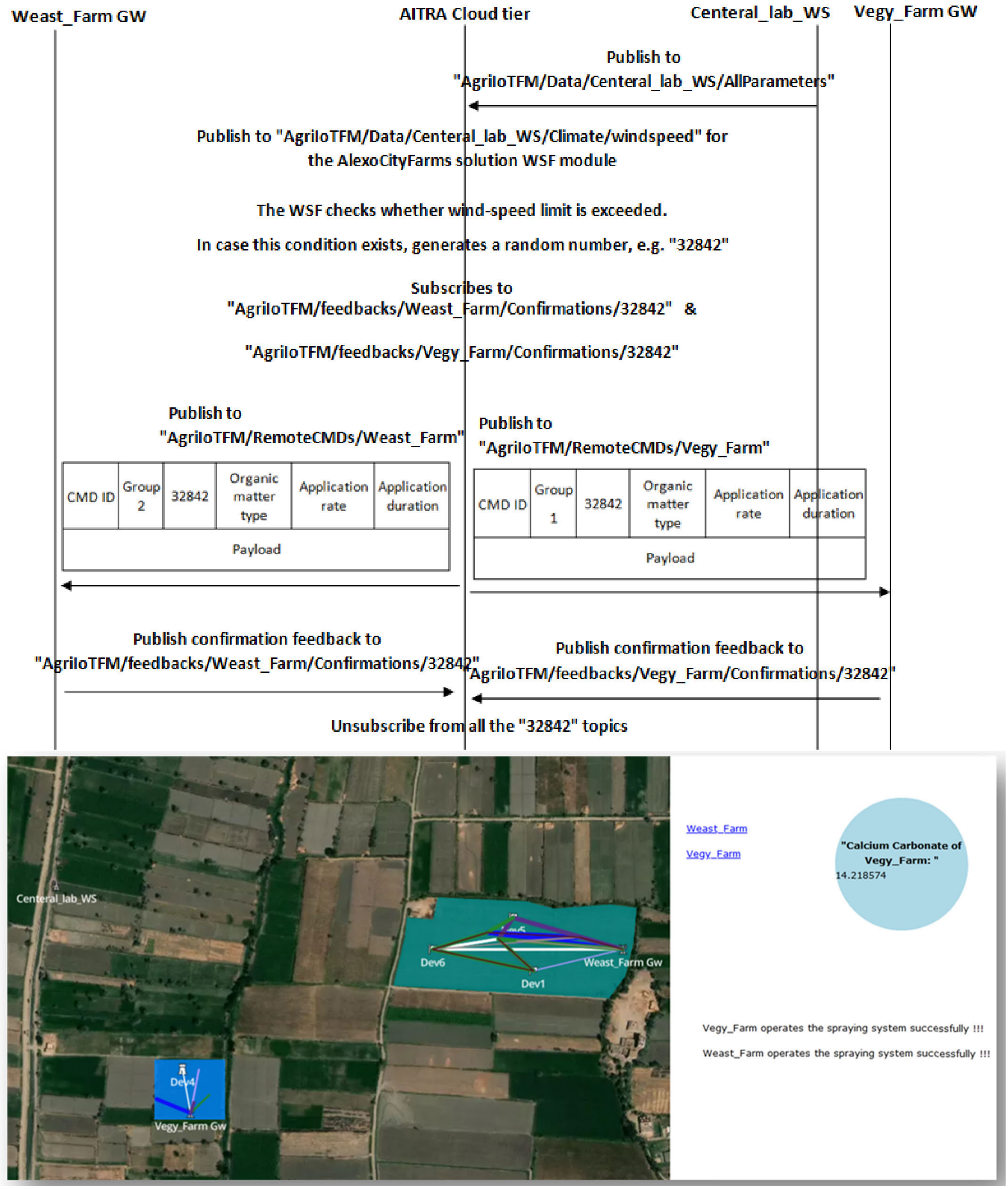

FIGURE 32. Message sequence chart of the closed-loop control interactions, and the resulting notifications appear on the webpage.

useful need to stay up-to-date with recent trends in the specialization field. AITRA is specialized in architecting IoT-enabled solutions for the agriculture domain. A continuous process for the generation of such architecture was outlined based on conducting extensive analysis on the agricultural ecosystem takes into account the identification of stakeholders and their roles, the existing solutions and use cases, market trends, policies, and feedback from the architecture's users;

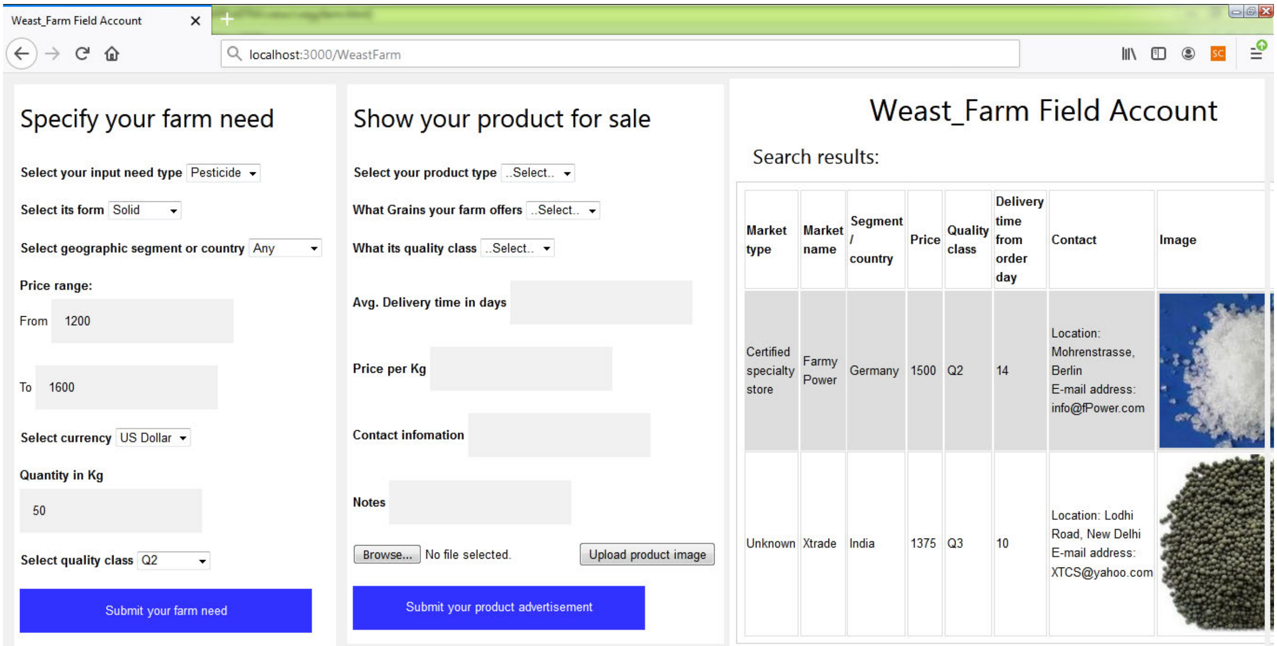

FIGURE 33. The Weast_Farm Field Account webpage, where it published its need in Pesticide and the corresponding found options are tabulated.

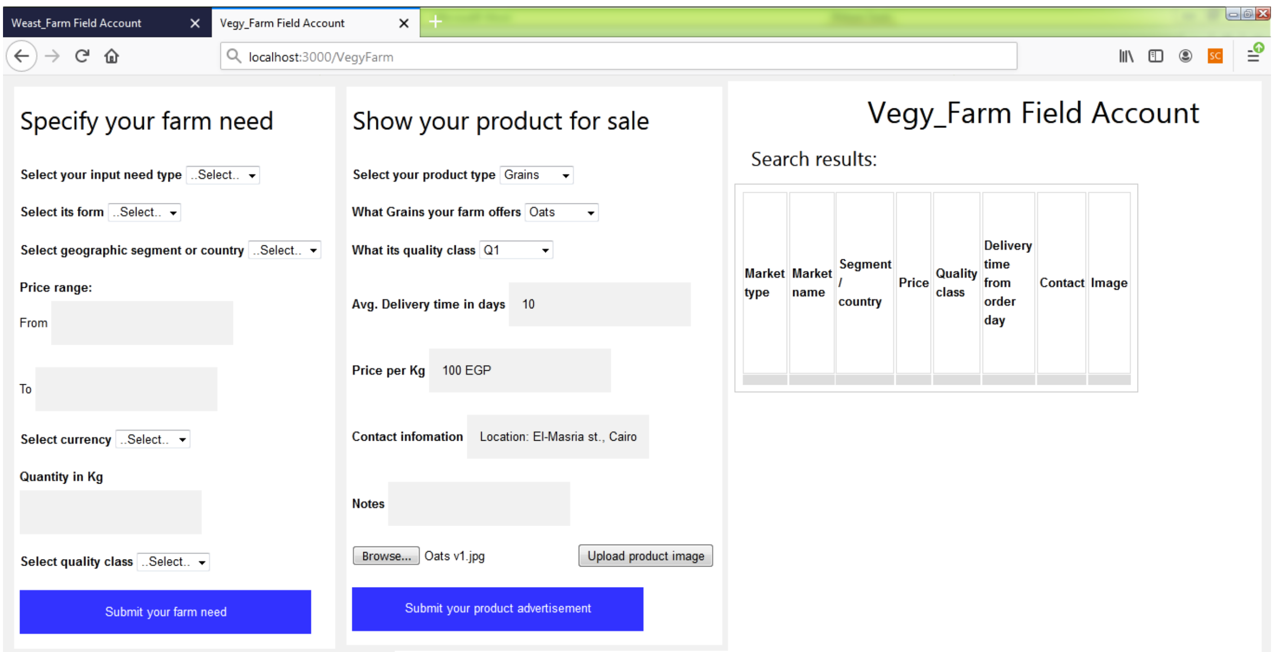

FIGURE 34. The Vegy_Farm Field Account webpage, where it published a sale advertisement of its produced Oats to any interested entity.

in addition the process also relays on the experience of the IoT architects and protocol stack developers gained from their knowledge of their work ecosystems in architecting, implementing, updating, enhancing the architecture and its instantiations.

The proposed AITRA design is already developed based on an accurate but of course incomprehensive analysis conducted on the farming ecosystem. The analysis was keen to consider all the details of the ecosystem; thus it started with its main building block, the farming system, and proceeded to define its types, the specification and requirements of each type; then, it drawn flowcharts to describe the different agricultural practices management processes. These details help in determining the stakeholders, the services useful for

TABLE 11. AITRA features.

AITRA	
Type	Agricultural application dedicated
Layers	3 main tiers analogue to 4 cross-layer-designed functional layers and 3 cross-cutting layers (also can be viewed as 7 cross-layer-designed horizontal layers)
Supported devices vendors	Vendor neutral, support any IoT device management
Devices types	- Gateway, Edge gateway, standalone or embedded IoT Device, standalone or embedded non-IP-enabled devices - No defined levels for management, the management commands of the IoT devices will be stable and accurately specified according to their proposed architectures in Fig. 9; and the commands manage the non-IP-enabled devices will be defined by their registered management libraries. All these commands are available through the Service layer's interfaces
Standardizing specification for georeferenced physical and logical topologies of the local non-IP-enabled devices network	Considered in the standard frame format PAN related subfields, and in the reliance on standardized lightweight human readable Data Interchange Format for describing PAN related information and different options for them
Security support	Identity & Authentication: the messaging protocol client id (a word distinguishes the agricultural field or the IoT device), and username & password, Authorization: Access Control List (ACL) and topic access control. Transport Encryption : TLS / SSL. Web Security: any appropriate technologies
Interface to users and developers (level of abstraction)	GUI and API
Mentioned user interfaces	Generic as well as application-specific interfaces help users to interact with the full Cloud tier layers' capabilities
Modularity and Extendibility (add/delete/replace components)	Modular architecture
Recommended IoT messaging technologies	Publish/subscribe pattern
Standardization of communication packets structure	Considered
M-to-M and M-to-application communication support	Supported
Telemetry, configuration, and control of local non-IP-enabled devices network	Considered clearly; supported in the topics tree structure, the frames format, and the edge computing capability of the Edge gateway.
External application interaction support	Supported
Naming convention	Appears mainly on the topic-tree-based routing structure and frames' fields naming
Ecosystem analysis and best practices	Based on a complete ecosystem analysis via the proposed architecture generation process
Resilience to changes in the underlying communication infrastructure	A Transmission options subfield is included in the fames to accommodate the transport changes and network scalability, such as options for different distributed brokers architecture methods, and options for different message filtering and routing
Completeness	Still a research proposal needs further specifications such as the primitives of the layer's interfaces, and needs a real functional complete implementation

them, the types of relevant outputs, the relation of the farming system and its components to the outside world, moreover, the detailed descriptions produced common definitions which help in standardization.

Likewise, the IoT ecosystem was analyzed to derive the appropriate layered architecture specifying the layers roles, decomposition, and interactions, and specifying the technological specifications that help in achieving the standardization target.

AITRA uses the cross-layer design principle, and it can be viewed as seven cross-designed horizontal layers characterized by low level of abstraction provides the users with detailed functions of an agricultural system. meanwhile it includes a GUI layer exposes the domain-specific conceptualized applications as well as most of the other Cloud tier's functionalities to the users such that the user is allowed to include also in his solution lower levels of abstractions using a rich plug-and-play easy-to-use graphical components; this reduces the time to market, but it allows the ordinary users to use the architecture's platforms to build their solutions themselves.

Moreover, it exposes standardized API interfaces and facilitates the registration of new modules and libraries to be integrated into the system and be able to connect to the other modules and libraries through their APIs.

As some of the other RAs recommended certain IoT messaging protocols or patterns, AITRA recommended a lightweight publish/subscribe standard messaging pattern for its various benefits - including security, scalability, availability, reliability - and suitable characteristics to the agricultural applications which have been further utilized towards standardizing AITRA. The publish/subscribe is taken as the transport for the AITRA standard communication protocol. The publish/subscribe payload is serialized to the AITRA standard frame structures. The topic-tree-based routing is exploited to establish a common language between AITRA-based solutions and AITRA IoT devices/gateways, among AITRA-based solutions, and among AITRA-based solutions and external applications including solutions follow the same approach of AITRA but specialized in another domain – all this with authorization rules and at any scale: at the individual level, at the company level, at the government(s) level, and at the global level, which makes AITRA far from the specialization's isolation risk. The topic-tree structure is made to be understandable and comprehensive according to the conducted ecosystem analysis, and it is extendable can bear further branching in the future.

From the IoT ecosystem analysis, the devices can be classified into four types each of them has different nature and needs different consideration in the design. No one from the reviewed RAs consider all of these types. The AITRA Gateway is a piece of software installed in any gateway hardware to perform its functionality. Besides its other roles, the Gateway shields the non-IP-enabled devices from the Internet and makes any translation needed to enable the bidirectional communication between them and the AITRA platform, which supports the vendor neutrality. The communication over the cloud is considered to be near-real-time, but the Edge gateway facilitates real-time operations and increases the system reliability.

The AITRA Gateway together with the local non-IP-enabled devices form the things network; the standard frames format allows it to describe and update its geographical boundary, physical distribution in relation to farm enterprises, the specification of each device, and the logical topology and communication-related specification. These capabilities are not offered by the other reviewed RAs.

Also, the data, commands, and notifications are exchanged in standard data format with standard rules for preparing it, considering the different possible types of generated data, events, commands; all in standard format. AITRA doesn't specify certain management levels or command types for the two classes of non-IP-enabled devices, leaves it for the devices vendors or developers through their management libraries registered with the platform (a support for general common types of commands can also be added by adding a corresponding branch in the topic tree under the level RemoteCMDs rooted at topic name ''GeneralCMDs'', the level under GeneralCMDs contains descriptive topic name for each command, and the gateways subscribe to the whole level under GeneralCMDs). All of these specifications are not considered in the other reviewed RAs.

AITRA includes a Transmission options subfield in the frame control field to accommodate the different existing possibilities of the underlying communication infrastructure, in addition to any possible capabilities appended to it in the future.

The AITRA platform offers applications related to farming practices, such as farming and cropping systems suggestion, farm layout suggestion and configuration update monitoring, defining an appropriate CEA level for different farm enterprises, and defining solutions for each farming practice; mapping applications such as generation of different prescription maps and satellite-based, airborne-based and UAV-based mapping; simulation and modeling applications such as generating crop simulation models and yield prediction; economics and financial inclusion; applications for monitoring and actuation of the field devices; application for managing the network of the field devices, such as determining the best number and placement, setting of the network communication technology parameters and control of operation; FieldAccount related applications.

Furthermore, AITRA platform provides libraries for visualization methods suitable to the applications' different outputs, standard definitions, solution-specific definitions, vendor-specific devices management libraries, analytics libraries, standardized lightweight human readable Data Interchange Format for different applications-related descriptions, etc. The modularity with the precise specialization of each module makes room for specialized companies and individuals to compete in developing different implementations with varying properties of the modules.

AITRA design combines the three main characteristics of a reference architecture: best practices, common vocabulary, and reusable designs; of course AITRA is still an incomplete standard, it is a research proposal laying the foundation for a standard reference architecture. It needs auditing, further details in its specification, revision and refinements using complete real instantiation. The farming ecosystem analysis method used in this paper can be replaced or augmented by the approaches proposed by the other RAs for guiding the domain analysis.

The existence of a global standard, smartly connects the agriculture world, results in its spread like its infrastructure - the Internet - spread, such that everyone in the world can have access to its functions. This would cause a revolution in the agriculture sector paves the way for a new concept of Internet of agriculture world ''inter-agriculture-networking'' can be generalized to other application domains: ''inter-trade-networking'', ''inter-industry-networking'', ''inter-tourism-networking'', etc., which appear as networking layers above the Internet. Developing a global standard and bringing it to light requires concerted efforts, worldwide cooperation, and support from the governments and the community, but this finds a great motivation for it where the world seeks towards the digital transformation.

X. CONCLUSION AND FUTURE WORK

The standardization of IoT architecture to make the Internet things interoperable is not an easy task especially that the IoT ecosystem incorporates a lot of application domains, each domain has its own complex ecosystem. The difficulty in developing an IoT reference architecture can be reduced if a separate reference architecture with design-choices is considered for each IoT application domain. The Agricultural IoT Reference Architecture (AITRA) proposed in this paper represents an example of this principle can be generalized to other application domains with customizations fit their ecosystems.

The development of the IoT reference architecture is a continuous process with repeated enhancements and amendments, needs first the drawing of a descriptive scheme defines

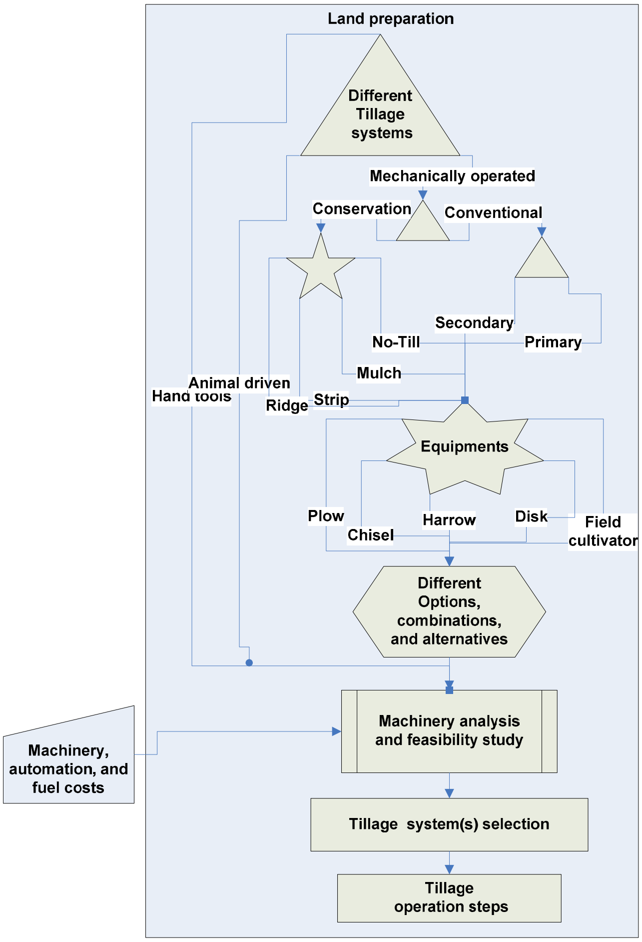

FIGURE 35. **Land preparation management.**

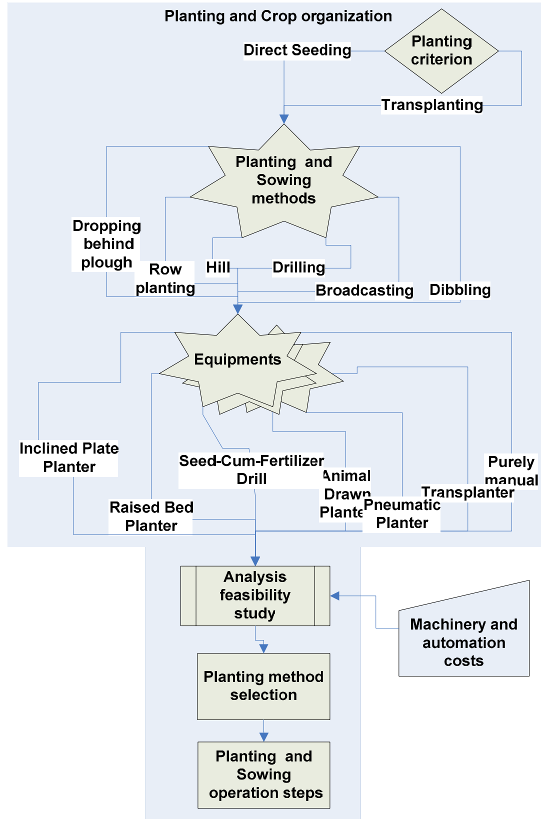

FIGURE 36. **Planting and crop organization management.**

the development steps, the stakeholders and their relations and roles.

The second step of AITRA design is the analysis of the farming ecosystem that is formulated using the inputs from the stakeholders; in the beginning, the farming was classified into different system types with different specifications, then the agricultural practices were analyzed. The third step of AITRA design was the analysis of the IoT ecosystem.

Exploiting the developed stakeholders scheme, farming ecosystem analysis, and IoT ecosystem components' definitions, AITRA was designed to include three tiers: the Device tier, the Cloud tier consists of five cross-designed layers, and the Business tier. The standardized description of these tiers including structures, naming conventions, frame format, assumptions, the highest abstraction level user interface illustration, etc. are presented in this paper.

The AITRA design results in standardizing an IoT reference architecture characterized by extendibility where its layered modular design approach facilitates services swapping and architecture amending, reliability and effectiveness in terms of a rich set of modules and libraries perform agricultural functions, including financial and economic services, based on analysis of the agricultural and IoT ecosystems, a rich set of modules and libraries take care of the devices network issues, easy and friendly interface to users, API as well as GUI access to the lower functions of the architecture to ease and enrich application development, interoperability between its different solutions and with external applications, scalability in terms of covered regions, number of fields, and connected applications, availability through the servers/brokers clustering, transport as well as authentication and authorization security considerations, descriptive nearly comprehensive easy-to-extend routing topic-tree, standard communication paradigm and format of packets, small time to market, vendor-neutrality, and edge computing possibility.

With commitment to AITRA standardized descriptions the agricultural equipments and applications/stakeholders can talk to each other and form complete diversified multi-level solutions. As illustrated by the heuristic example given in the paper, the farm landlord can by himself develop the AITRA solution suitable to his farm through which he can layout, monitor and control the farm. The AITRA simple implementation presented in this paper proved that with small number of mouse clicks, a city management authority could build its desired solution which would take several days of experts work without the aid of AITRA platform to be implemented and be functional; in addition, it can connect its farms to the

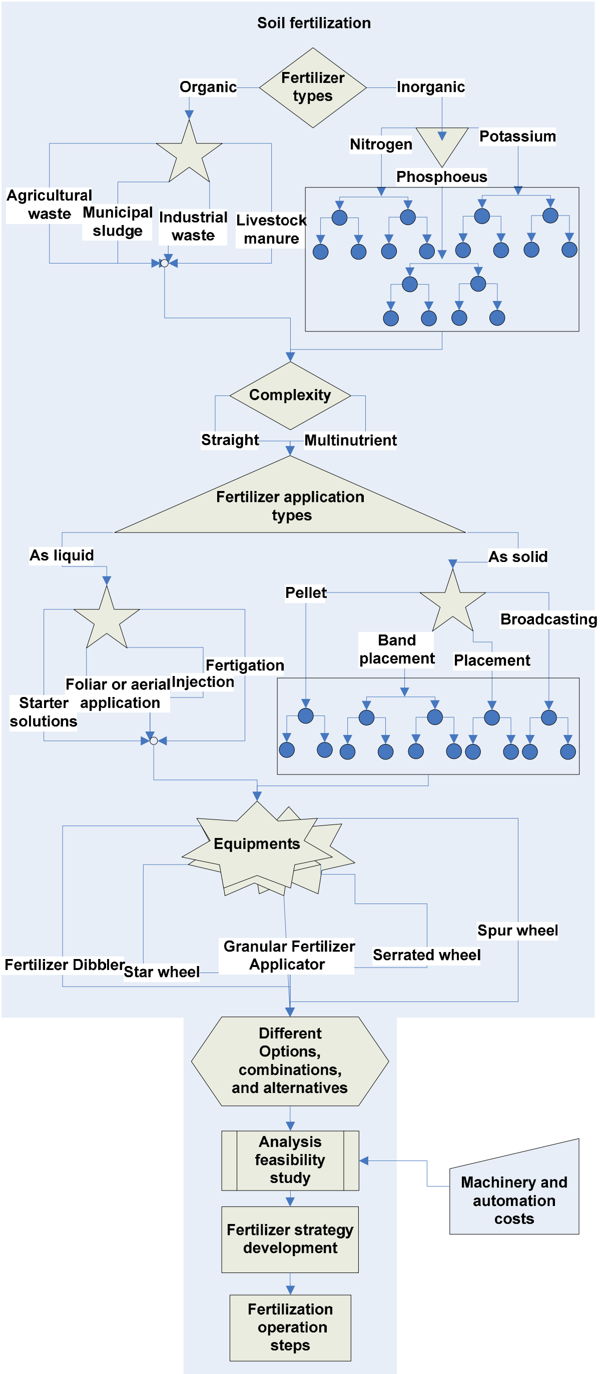

FIGURE 37. Soil fertilization management.

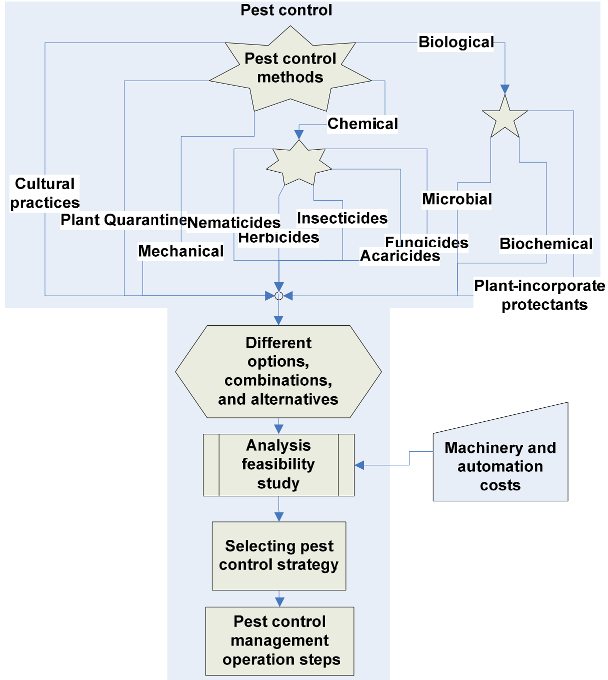

FIGURE 38. Pest control practice management.

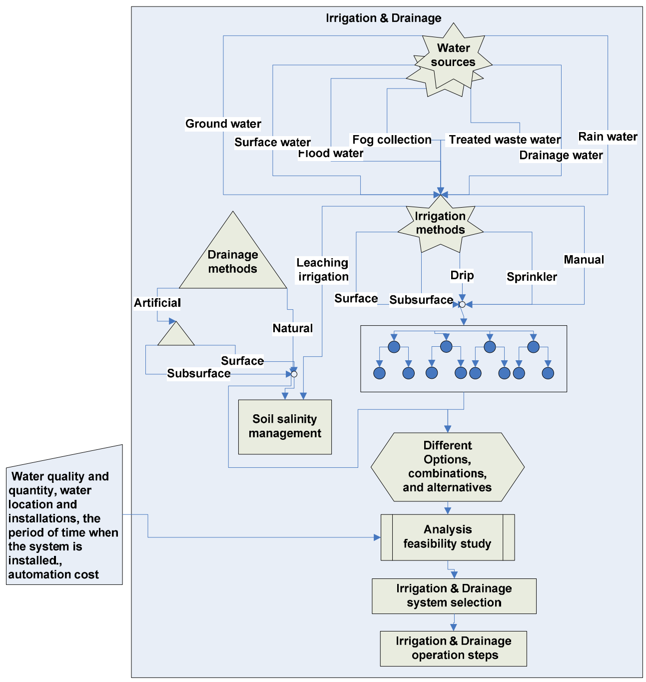

FIGURE 39. Irrigation and drainage management.

AITRA economical and financial global network to benefits from its offered services.

AITRA is still not in a complete form of standard, but it lays the foundation for a standard agricultural IoT reference architecture. The future work for developing AITRA includes strictly following the proposed IoT ecosystem's reference architecture generation process for refining the system functionalities, giving more consideration to security functions, completing its standard specification such as defining the interfaces and primitives, methods of new components registration, GUI layer conventions, Data Interchange Formats, devices types, commands types, notification types, etc. The last and most important future work represents the natural evolution of an IoT reference which is the derivation of a concrete architecture, or in better words, the implementation of the AITRA platform.

APPENDIX

A. FLOWCHARTS FOR MANAGEMENT OF SOIL-BASED CULTIVATION PRACTICES

See Figures 35–39.

REFERENCES

[1] Q. F. Hassan, *Internet of Things A to Z: Technologies and Applications*. Hoboken, NJ, USA: Wiley, 2018.

[2] S. J. Ramson, S. Vishnu, and M. Shanmugam, ''Applications of Internet of Things (IoT)–An overview,'' in *Proc. 5th Int. Conf. Devices, Circuits Syst. (ICDCS)*, Coimbatore, India, Mar. 2020, pp. 92–95, doi: 10.1109/icdcs48716.2020.243556.

[3] P. V. Dudhe, N. V. Kadam, R. M. Hushangabade, and M. S. Deshmukh, ''Internet of Things (IOT): An overview and its applications,'' in *Proc. Int. Conf. Energy, Commun., Data Anal. Soft Comput. (ICECDS)*, Chennai, India, vol. 2017, pp. 2650–2653.

[4] E. Selem, M. Fatehy, S. M. Abd El-Kader, and H. Nassar, ''THE (Temperature heterogeneity Energy) aware routing protocol for IoT health application,'' *IEEE Access*, vol. 7, pp. 108957–108968, 2019, doi: 10.1109/ACCESS.2019.2931868.

[5] S. Ziegler, *Internet of Things Security and Data Protection*. Cham, Switzerland: Springer, 2019.

[6] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, ''A comparative study of LPWAN technologies for large-scale IoT deployment,'' *ICT Express*, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.

[7] B. M. M. El-Basioni, A. I. Moustafa, S. M. A. El-Kader, and H. A. Konber, ''Timing structure mechanism of wireless sensor network MAC layer for monitoring applications,'' *Int. J. Distrib. Syst. Technol.*, vol. 7, no. 3, pp. 1–20, Jul. 2016, doi: 10.4018/IJDST.2016070101.

[8] B. M. Mohammad El-Basioni, S. M. Abd El-kader, and H. S. Eissa, ''Designing a local path repair algorithm for directed diffusion protocol,'' *Egyptian Informat. J.*, vol. 13, no. 3, pp. 155–169, Nov. 2012, doi: 10.1016/j.eij.2012.07.001.

[9] B. M. Mohammad El-Basioni, A. I. Moustafa, S. M. Abd El-Kader, and H. A. Konber, ''Designing a channel access mechanism for wireless sensor network,'' *Wireless Commun. Mobile Comput.*, vol. 2017, Jan. 2017, Art. no. 7493269, doi: 10.1155/2017/7493269.

[10] M. A. Salem, I. F. Tarrad, M. I. Youssef, and S. M. Abd El-Kader, ''QOS categories activeness-aware adaptive EDCA algorithm for dense IoT networks,'' *Int. J. Comput. Netw. Commun.*, vol. 11, no. 03, pp. 67–83, May 2019, doi: 10.5121/ijcnc.2019.11305.

[11] M. A. Salem, I. F. Tarrad, M. I. Youssef, and S. M. A. El-kader, ''An adaptive EDCA selfishness-aware scheme for dense WLANs in 5G networks,'' *IEEE Access*, vol. 8, pp. 47034–47046, 2020, doi: 10.1109/ACCESS.2020.2979052.

[12] J. Guth, ''A detailed analysis of IoT platform architectures: concepts, similarities, and differences,'' in *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*. B. Di Martino, K. C. Li, L. Yang, and A. Esposito, Eds. Singapore: Springer, 2018, pp. 81–101.

[13] H. Muccini and M. T. Moghaddam, ''IoT architectural styles: a systematic mapping study,'' in *Software Architecture, Springer Nature*. Cham, Switzerland: Springer, 2018, ch. 5, sec. 2, pp. 68–85.

[14] T. Jacobs, ''A Reference architecture for IoT-enhanced business processes,'' M.S. thesis, Dept. Bus. Inf. Manage., Fac. Econ. Bus., KU Leuven, Leuven, Flanders, Belgium, 2019.

[15] P. Singh, ''Cross-layer design for Internet of Things (IoT)-issues and possible solutions,'' Syst. Comput. Eng. Dept., Carleton Univ., Ottawa, ON, Canada, Tech. Rep. Dec. 2018. [Online]. Available: https://www.researchgate.net/publication/329970346_Cross-Layer_Design_for_Internet_of_Things_IOT-Issues_and_Possible_Solutions, doi: 10.13140/RG.2.2.12865.20327.

[16] D. Rakic. *IoT Reference Architecture*. Accessed: Oct. 2019. [Online]. Available: https://www.novatec-gmbh.de/en/blog/iot-reference-architecture/

[17] P. Hurni, T. Braun, B. K. Bhargava, and Y. Zhang, ''Multi-hop cross-layer design in wireless sensor networks: A case study,'' in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Oct. 2008, pp. 291–296, doi: 10.1109/WiMob.2008.28.

[18] P. Fremantle, ''A reference architecture for the Internet of Things,'' WSO2 White Paper Version 0.9.0, Oct. 2015. [Online]. Available: https://wso2.com/whitepapers/a-reference-architecture-for-the-internet-of-things/#03

[19] *The OAuth 2.0 Authorization Framework*, OAuth 2.0, Internet Soc.—IETF OAuth Working Group, Reston, VA, USA, Oct. 2012.

[20] WSO2. *WSO2 IoT Server Documentation*. Accessed: Sep. 2, 2020. [Online]. Available: https://docs.wso2.com/display/IoTS330

[21] A. Bassi *et al.*, Eds., ''The IoT-architecture project (IoT-A),'' in *Enabling Things to Talk—Designing IoT Solutions With the IoT Architectural Reference Model*, 1st ed. Berlin, Germany: Springer-Verlag, 2013. [Online]. Available: https://www.springer.com/gp/book/9783642404023

[22] *Industrial Internet Consortium, The Industrial Internet Consortium: A Global Not-for-Profit Partnership of Industry, Government and Academia*. Accessed: Sep. 2, 2020. [Online]. Available: https://www.iiconsortium.org/about-us.htm

[23] S.-W. Lin, ''The industrial Internet of Things,'' Ind. Internet Consortium, Milford, CT, USA, Tech. Rep. IIC:PUB:G1:V1.80:20170131, (Volume G1: Reference Architecture), 2017. [Online]. Available: https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf

[24] *Systems and Software Engineering–Architecture Description*, ISO/IEC/IEEE Standard 42010-2011, Dec. 2011.

[25] Ö. Köksal and B. Tekinerdogan, ''Architecture design approach for IoT-based farm management information systems,'' *Precis. Agricult.*, vol. 20, no. 5, pp. 926–958, Oct. 2019, doi: 10.1007/s11119-018-09624-8.

[26] S. Mathew, ''Overview of Amazon Web services: AWS Whitepaper,'' Amazon Web Services, Seattle, WA, USA, White Paper, Aug. 2020. [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/aws-overview.pdf

[27] *Amazon Web Services, Inc. and/or its Affiliates, AWS IoT: Developer Guide*. Accessed: Sep. 3, 2020. [Online]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

[28] P. Salhofer, ''Evaluating the FIWARE platform: A case-study on implementing smart application with FIWARE,'' in *Proc. 51st Hawaii Int. Conf. Syst. Sci. (HICSS)*, Big Island, HI, USA, Jan. 2018, pp. 5797–5805.

[29] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, ''A standard-based open source IoT platform: FIWARE,'' *IEEE Internet Things Mag.*, vol. 2, no. 3, pp. 12–18, Sep. 2019, doi: 10.1109/IOTM.0001.1800022.

[30] Rayven. *IoT in Agriculture: Smart Farming Platform*. Accessed: Sep. 3, 2020. [Online]. Available: https://www.rayven.io/agriculture-iot-solutions/

[31] *Thethings.iO, Agriculture: IoT Solutions for Agriculture and Farming*. Accessed: Sep. 3, 2020. [Online]. Available: https://thethings.io/iot-agriculture/

[32] D. Savvas and N. Gruda, ''Application of soilless culture technologies in the modern greenhouse industry–A review,'' *Eur. J. Hortic. Sci.*, vol. 83, no. 5, pp. 280–293, 2018, doi: 10.17660/eJHS.2018/83.5.2.

[33] S. Goddek, A. Joyce, B. Kotzen, and M. Dos-Santos, ''Aquaponics and global food challenges,'' in *Aquaponics Food Production Systems*, S. Goddek A. Joyce B. Kotzen, G. Burnell, Eds. Cham, Switzerland: Springer, 2019, pp. 3–17.

[34] C. Baraniuk. *How Vertical Farming Reinvents Agriculture*. Resource document. BBC. Accessed: Apr. 27, 2020. [Online]. Available: https://www.bbc.com/future/article/20170405-how-vertical-farming-reinvents-agriculture

[35] *Node.js. (12.18.4 LTS), Joyent*. Accessed: Oct. 1, 2020. Available:. Accessed: Oct. 1, 2020. [Online]. Available: https://nodejs.org/en/

[36] (2013). *Mapbox*. Accessed: Oct. 1, 2020. [Online]. Available: https://www.mapbox.com/

[37] Paho MQTT C Client. (2009). *IBM Corp*. Available:. Accessed: Oct. 1, 2020. [Online]. Available: https://www.eclipse.org/paho/files/mqttdoc/MQTTClient/html/index.html

BASMA M. MOHAMMAD EL-BASIONI received the M.Sc. and Ph.D. degrees from Al-Azhar University, Cairo, Egypt, in 2011 and 2017, respectively, both in electronics and communications engineering. She is currently a Researcher with the Computers and Systems Department, Electronics Research Institute (ERI), Egypt, where she is also the Supervisor of the Technology and Innovation Support Center (TISC). She has a good background on Intellectual Property (IP). Her current research interests focus on the IoT technology and communication networks, especially WSN in different applications. She has published more than twelve paper and chapter, and has two copyrights and a patent application. She is a Reviewer for some International publishers.

SHERINE M. ABD EL-KADER received the M.Sc. degree from the Electronics and Communications Department, Faculty of Engineering, Cairo University, in 1998, and the Ph.D. degree from the Computers Department, Faculty of Engineering, Cairo University, in 2003. She was the Head of the Internet and Networking Unit, Electronics Research Institute (ERI), from 2003 to 2014, and was also the Head of the Information and Decision Making Support Center, ERI, from 2009 to 2014. She was an Associate Professor with the Faculty of Engineering, Akhbar El Yom Academy, from 2007 to 2009. She was the Head of the Technology Innovation Support Center (TISC), ERI, from 2013 to 2017, where she has been a Professor with the Computers and Systems Department, since April 2014. She has been a member of the Technological Development and Linkage with the Industry and Civil Society Committee, since 2016, and has been the Head of the Media Committee, since 2017. She has been the Head of the Computers and Systems Department, ERI, since 2018. She has also been the Head of the Technology, Innovation and Commercialization Office (TICO) since 2018. She has supervised more than 20 M.Sc. and Ph.D. students. She has published more than 50 papers, six book chapters in computer networking area, and an Editor of a book in 5G. She is working in many computer networking hot topics such as: the IoT, 5G, cognitive radio, Wi-MAX, Wi-Fi, IP mobility, QoS, wireless sensors Networks, ad-hoc networking, real time traffics, and localization algorithms. She has supervised many automation and web projects for ERI. She was also a Technical Member with the ERI Projects Committee and at the Telecommunication Networks Committee, Egyptian Organization for Standardization and Quality, from 2007 to 2011. She has two copyrights and one patent application with a great background on Intellectual property. She is also a Technical Reviewer for many International journals.