WAZIUP: An Open Innovation Platform to Accelerate Innovation in Africa through cutting edge IoT cost effective communications and Big Data applications

Charlotte Dupont[1], Sabrine Fatnassi[1], and Corentin Dupont[2]

[1] Create-Net via Alla Cascata, 56/D 38123 Trento, Italy
[2] Easy Global Market Espace Beethoven 1200 Route des lucioles 06560 Valbonne, France
charlotte.dupont,sabrine.fatnassi@eglobalmark.com cdupont@create-net.org

Abstract. Abstract...

1 Introduction

WAZIUP is a 3 years H2020 international cooperation action started on February 1st, 2016. The project is driven by a consortium of 5 EU partners and of 7 partners from 4 sub-Saharan African countries. Furthermore, it has support from multiple African stakeholders with the aim of defining new innovation space to advance the African Rural Economy. It will do so by involving end-users communities in the loop, namely rural African communities of selected pilots, and by involving relevant public bodies in the project development. WAZIUP will accelerate innovation in Africa by coupling with current EU innovation in the sector of IoT and Big Data: this EU technology will be specialized to generate African cost effective technologies with an eye to preparing the playground to the future technological waves by solving concrete current needs. WAZIUP will deliver a communication and big data application platform and generate locally the know how by training by use case and examples. The use of standards will help to create an interoperable platform, fully open source, oriented to radically new paradigms for innovative application/services delivery.

1.1 ICT African context

ICT, in Sub-Saharan Africa, in several cases has enabled convergence of productive sectors, serving as platform for more holistic development. In fact, there are many examples of ICT developments in Africa that cut across traditional sectors: notable examples are the introduction of micro-health insurance and health-savings accounts through mobile devices; index-based crop insurance; crowd-sourcing to monitor and manage the delivery of public services, etc. These innovative applications for several reasons more disruptive in social terms than many counterparts in the EU - recognize and leverage commonalities between sectors, blur traditional lines, and open up a new field of opportunities.

The opportunity for ICT intervention in Africa is huge especially of IoT and big data: those technologies are promising a big wave of innovation for our daily life. It is widely accepted that the Era of IoT can potentially connect billions of sensors, devices, equipment, systems, etc. In turn, the challenge is about driving business outcomes, consumer benefits, and the creation of new value. The new mantras for the IoT Era are becoming collection, convergence and exploitation of data. The information collection involves data from sensors, devices, gateways, edge equipment and networks on to their respective siloed IoT platforms in order to increase process efficiency through automation while reducing downtime and improving people productivity.

WAZIUP targets the rural community in Sub-Saharan Africa because about 64% of the population is living outside cities. The region will be predominantly rural for at least another generation. The pace of urbanization here is slower compared to other continents, and the rural population is expected to grow until 2045. The majority of rural residents manage on less than few Euros per day. Rural development is particularly imperative in sub-Saharan Africa, where half of the rural people are depend on the agriculture/micro and small farm business, other half faces rare formal employment and pervasive unemployment. For rural development, technologies have to support several key application sectors like living quality, health, agriculture, climate changes, etc. WAZIUP project consider how to best design and deploy the IoT-Big Data technology considering cost and energy challenges in the first place.

Beside the cost and power consumption, the robustness of hardware is a core requirement: hardware has to be robust enough so as to require lower maintenance and handle environmental and deployment threats as well. WAZIUP will collect the grand challenge: reduce costs, reduce power consumption but at the same time increase the robustness of the hardware.

1.2 Project challenges

WAZIUP presents several challenges as listed below:

- *Innovative design of the IoT platform for the Rural Ecosystem.* Low-cost, generic building blocks for maximum adaptation to end-applications in the context of the rural economy in developing countries.
- *Network Management.* Facilitate IoT communication and network deployment. Lower cost solutions compared to state of the art technology: privilege price and single hop dedicated communication networks, energy autonomous, with low maintenance costs and long lasting operations.
- *Long distance.* Dynamic management of long range connectivity (e.g., cope with network & service fluctuations), provide devices identification, abstraction/virtualization of devices, communication and network resources optimization.
- *Big-data.* Exploit the potential of big-data applications in the specific rural context.

From a technical standpoint, WAZIUP introduces innovation by constructing on the following pillars of IoT/Big Data technology, specifically tailored for the rural ecosystem:

- *Privacy and security:* through attention to all related privacy and security aspects with specifics addressing the involved communities (farmers, developers);
- *Personalized and user friendliness:* models will receive requirements from users needs and will ensure compliance with all most common usability standards (e.g., Web Accessibility Initiative - WAI or ISO/TR 16982:2002);
- *An Open interoperable platform:* through open standard and protocols from the Geospatial Consortium (OGC), W3C, IEEE from the European SDOs (CEN, CENELEC and ETSI, etc.) for all its key technology
- *Continued Openness:* through the release of open specification and open software components and/or algorithms;
- *Low-cost and low-energy consumption:* through the design of innovation hardware (sensors/actuators), and of IoT communication & network infrastructure.

2 IoT & Big Data Platform

Challenges of WAZIUP will be tackled using an open IoT-big data platform with affordable sensors connected through an IoT-Cloud open platform. This platform will also make use of mobile phones and real-time processing to empower users and deliver the needed services. The project will not develop any new IoT/Big Data software platform, but, rather exploit the existing solutions and adapt them for the purpose. Hereafter a compact list of core technical functionalities encompassed by the platform:

- *Cloud-based real-time data collection combined with analytics and automation software:* thus, the platform will offer cost-effective solutions for aggregating different machines and sensor types to engender efficiency, smart automation and optimization in the rural context.
- *Intelligent analytics of sensor and device data:* studied in order to optimize for performance of the rural workplace, detect potential outages, and finally reduce overall maintenance costs.
- *Integration to 3rd parties' platform:* enables customers' benefit of scaling fast and easy.
- *PaaS (Platform-as-a-Service) provider:* WAZIUP will provide to business clientele with independently maintained platform upon which their web application, services and mobile applications can be built.

2.1 Functional overview

This section presents the functional view of the architecture.

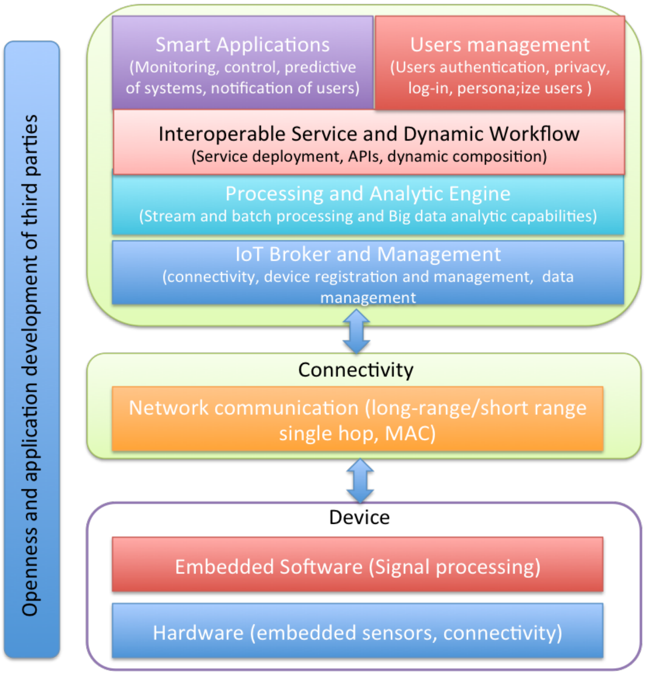
Openness and application development of third parties
Smart Applications
(Monitoring, control, predictive of systems, notification of users)
Users management
(Users authentication, privacy, log-in, persona;ize users)
Interoperable Service and Dynamic Workflow
(Service deployment, APIs, dynamic composition)
Processing and Analytic Engine
(Stream and batch processing and Big data analytic capabilities)
IoT Broker and Management
(connectivity, device registration and management, data management
Connectivity
Network communication (long-range/short range single hop, MAC)
Device
Embedded Software (Signal processing)
Hardware (embedded sensors, connectivity)

Fig. 1. Functional overview of WAZIUP

The Figure 1 displays the functional overview of WAZIUP. The topmost block represents the Cloud platform, the middle one is the network connectivity while the bottom one is the local deployment, including gateway and sensors. The following functional domains have been identified:

- *Application platform* Application writing, deploying, hosting and execution.
- *Stream and data analytic* Data brokering, stream processing and data analytics.
- *Users Management* Management of the identification, roles and connections of users.
- *Gateway, sensors and networks* The IoT connectivity, the sensors data and metadata.
- *Security and privacy* The anonymisation of the data, securisation of the transmissions.
- *Platform Management* Status of the components, deployment of the platform

Based on the functional domains, actors have been identified. An actor is either a physical person or an external system. There are five actors: the developer, the data provider, the sensor owner, the application user and finally the administrator.

Developer The developer uses WAZIUP platform to compile and deploy his application. The application is then hosted on the platform and accessible. Their tasks are:

- *Collect requirements for WAZIUP app*
- *Design app architecture*
- *Realize app*
- *Interact with WAZIUP API*
- *Deploy App in WAZIUP PF*
- *Maintain app*

The data provider The data provider is a third party owning an internet API to which WAZIUP is connecting in order to retrieve data. Their tasks are:

- *Integrate third party API*
- *Access control of API*
- *Maintain the API*

The sensor owner The sensor owner is deploying the sensors in the field, and then registering them on the WAZIUP platform. Sensor data then becomes available to applications. Their tasks are:

- *Analyze deployment*
- *Install sensors*
- *Register sensors with WAZIUP*
- *Configure sensors*

The application user The application user is accessing the application developed by the Developer and deployed on WAZIUP. Their tasks are:

- *Register with WAZIUP*
- *Access/use application*
- *Provide feedback on app*

The administrator The administrator manages and configure the platform; and administrates the users. Their tasks are:

- *Manage users accounts*
- *Configure the platform*
- *Control resources usage*

2.2 IoT platform

IoT architecture overview WAZIUP IoT platform architecture is presented in Figure 2. It has three main layers: device layer, gateway layer and the cloud layer.

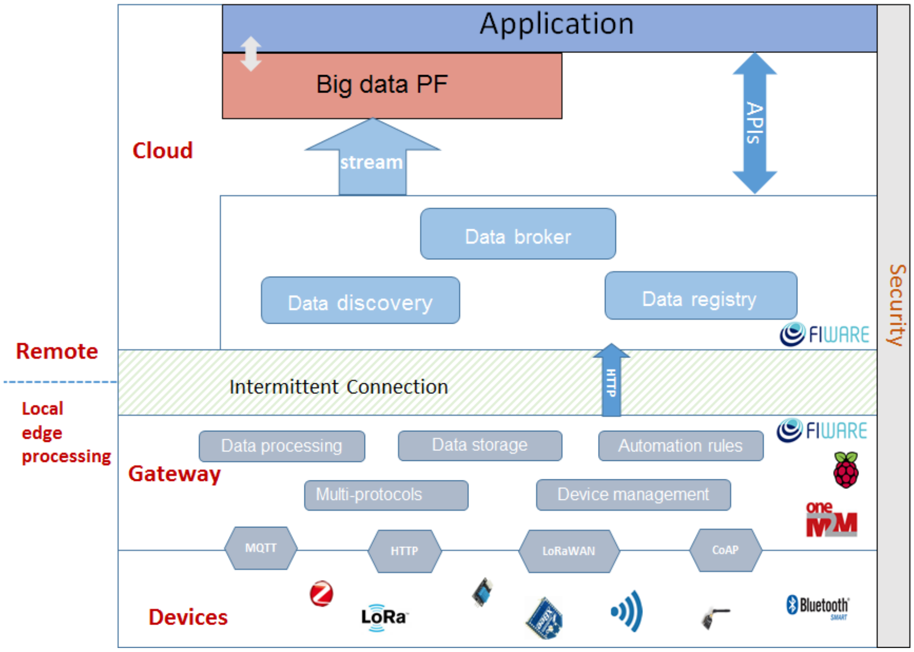

Fig. 2. IoT platform architecture

- *The device layer* It includes IoT sensors and actuator that are able to communicate each on its protocol (MQTT, CoAP, HTTP, LoRAWAN) with the gateway. It a constraint and autonomous device. Considering the project circumstances it has to be low cost and doesnt consume much power.
- *The gateway layer* The gateway is a key component in our architecture since it ensure a secure bidirectional communication between the Iot devices and the cloud services. It interfere with various kind of devices: getting the data from sensors and sending commands to actuators. It has to be multi-protocol in order to be able to support diverse devices and it need to have device management capabilities. Since the connection is intermittent in our project, it is mandatory that the gateway is able to manage automation rules (for example some event processing), to store data locally and to have low speed internet connectivity (2G: GPRS, EDGE).
- *The cloud layer* Our cloud main component are: the Data broker, the data registry and the data discovery.
  - *Data broker:* it is the key component of the IoT cloud services, it is based on publish/ subscribe mechanism to ensure data transfer from different producers to their respected consumers.
  - *Data registry:* it contains the virtual representation of devices (the data and the meta-data)
  - *Data discovery:* it is responsible on querying the data.

2.3 Big Data Platform

TODO ▶*Charlotte*◀

Big Data Platform overview

3 Implementation

In this section we will present WAZIUP platform components and the technology and tools selected for each component.

3.1 Platform components

Here is a detailed view of WAZIUP platform components:

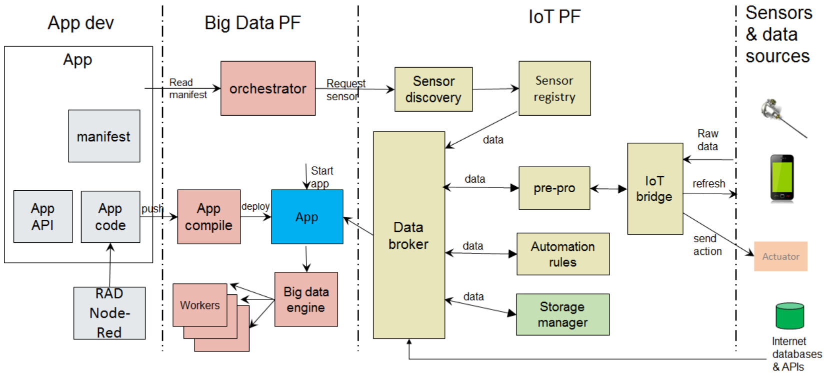

Fig. 3. WAZIUP IoT platform overview

The figure 3 presents the full WAZIUP architecture. There are 4 silos (from left to right): Application development, big data platform, IoT platform, Sensors and data sources. The first silo involves the development of the application itself. A rapid application development (RAD) tool can be used, such as Node-Red. The user provides the code source of the application, together with the manifest. The manifest describes the requirements of the application in terms of:

- *Compute needs (i.e RAM, CPU, disk)*
- *reference to data sources (i.e. sensors, internet - sources...)*
- *big data engines needed (i.e Flink, Hadoop...)*
- *configuration of sensors (i.e. sampling rate)*
- *local and global application deployment*

The application source code, together with the manifest, is pushed to the Waziup Cloud platform by the user. The orchestrator component will read the manifest and trigger the compilation of the application. It will then deploy the application in the Cloud execution environment. It will also instantiate the services needed by the application, as described in the manifest. The last task of the orchestrator is to request the sensor and data sources connections from the IoT components of the architecture. The sensor discovery module will be in charge of retrieving a list of sensors that matches the manifest description. On the left side of the diagram, the sensor owners can register their sensors with the platform. External data sources such as Internet APIs can also be connected directly to the data broker. The sensors selected for each application will deliver their data to the data broker, through the IoT bridge and preprocessor. The IoT bridge is responsible on the senors communication, it receives and transmits data to sensors using its specific protocol. The pre-process can apply transformation of the received data. Since internet connectivity is intermittenet, local storage and local automated rules are need. The automation rules component is responsible on executing local predefined rules and the storage manager is responsible on storing historic data.

3.2 Sequence diagrams

In this section the main two more important sequence diagrams are presented.

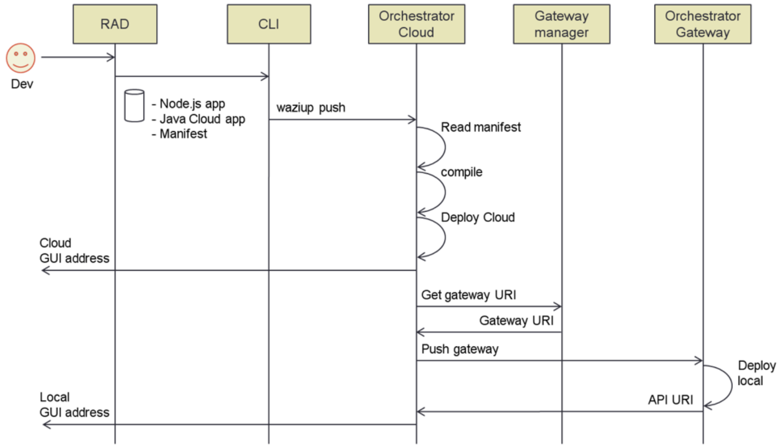

Fig. 4. Application deployment sequence diagram

The Figure 4 shows the sequence of an application deployment on Waziup platform. First, the developer uses the RAD to create a program source code. It is then pushed on Waziup platform using Waziup CLI. The Orchestrator of the Cloud instance read the manifest, compiles the application and deploys it in the Cloud execution environment. Using the information from the manifest and the gateway manager, it will also select the gateways where the application needs to be deployed. It will then contact the orchestrator from the corresponding gateways and ask them to deploy the application. If the gateway is not available at the moment, this action is delayed.

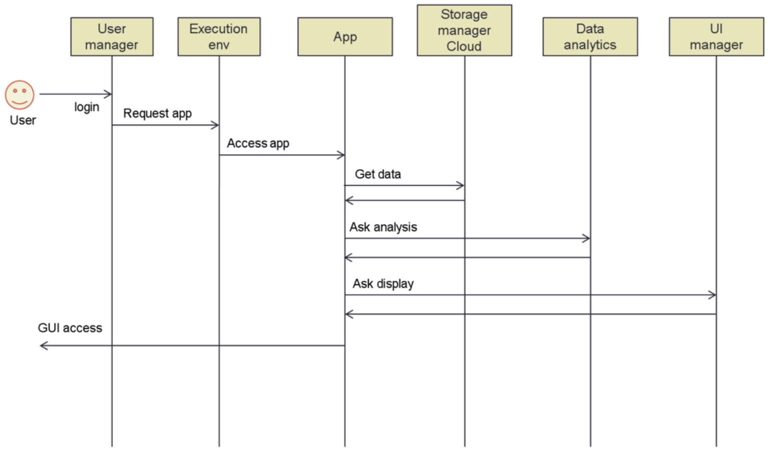

Fig. 5. Application run sequence diagram

The Figure 5 shows the sequence followed by a running application. First of all a user request the access to an application to the user manager, using his login and password. If granted, the User manager will forward the request to the execution environment which gives access to the app. The app is then getting its data from the Cloud Storage manager. It is further pushing the data to the data analytics component in order to get the elaboration. This elaborated data is then sent to the UI manager, in order to be displayed to the user.

3.3 Technology and tools

In this section the technology choices for each functional component stated above will be presented.

In the cloud level, functional components are the data broker, data registry and data discovery. To cover these functionalities, The FIWARE generic enabler Orion context broker is the all-in-one solution. It is based on the Next Generic Service Interfaces (NGSI) 9/10. NGSI specification defines a data model and interfaces to manage the whole cycle of virtual entities. other components can also be an option : the Nec IoT broker that covers the functionality of the data broker and uses the ConfMan component for the data discovery and registry.

In the gateway level, these functionalities have been identified:

- *Multiprotocol* The FIWARE IoT agent component: can be an option to ensure this characteristic. Each IoT agent can be responsible on the data transfer to and from the devices using the device specific protocol.
- *Device management* The LwM2M device management protocol is a good candidate, it is a light weight protocol specified by Oma .
- *Automation rules* FIWARE Cepheus enabler can be an option since it has a complex event processing engine.
- *Data processing* this component will be developed depending on use cases requirements.
- *Data storage* Mongo DB can be an option, other light data bases can be also considered such as SQLiteDB.

In the devices and networking level, LoRa technology captured our attention. It specifies LoRaWAN for Low Power Wide Area Network (LPWAN), the kind of technology needed in our project context (rural environment). Using LoRaWAN we can reach above 20 km in Line-Of-Sight (LOS) condition between devices and the gateway.

4 Deployment

4.1 Platform as a Service (PaaS)

Platform as a Service (PaaS) is a category of cloud computing service that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and deploying an application. Typically, a PaaS framework will compile an application from its source code, and then deploy it inside lightweight virtual machines, or containers.This compilation and deployment is done with the help of a file called the manifest, which allows the developer to describe the configuration and resource needs for his application. The manifest file will also describe the services that the application requires and that the platform will need to provision. Furthermore, PaaS environments usually offer an interface to scale up or down applications, or to schedule various tasks within the applications. The idea of WAZIUP is to extend the paradigm of the PaaS to IoT. Indeed, developing an IoT Big data application is a complex task. A lot of services need to be installed and configured, such as databases and complex event processing engines. Furthermore, it requires an advanced knowledge of:

- The various communication protocols.
- The programming of embedded devices.

– The storage the data in a distributed fashion.
– The programming of data stream processors.
– The programming of advanced data analytics.
– The programming of GUIs and user interactions.

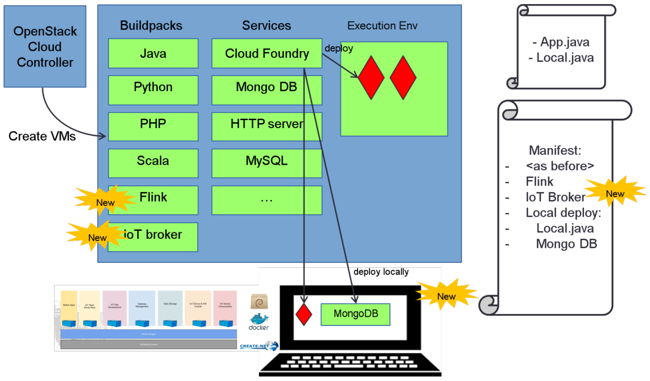

Fig. 6. PaaS deployment extended for IoT

The Figure 6 shows the PaaS deployment in WAZIUP. Traditional PaaS environment are usually installed on top of IaaS (in blue in the picture). The blue boxes are physical servers, respectively the Cloud Controller and one Compute node. The PaaS environment is then installed inside the IaaS VMs, in green in the picture. We use Cloud Foundry as a PaaS framework. It comes with a certain number of build packs, which and programming languages compilers and run time environments. It also provides a certain number of preinstalled services such as MongoDB or Apache Tomcat. The manifest file, showed on the right hand side, provide a high-level language that allows describing which services to instantiate. We propose to extend this language to IoT and big data services:

– Data stream and message broker
– CEP engines
– Batch processing engines
– Data visualization engines

Furthermore, we propose to include in the manifest a description of the IoT sensors that are required by the application. This query includes data such as the sensor type, location and owner. The manifest also includes the configuration of the sensors. The application will then be deployed both in the global Cloud and in the local Cloud.

4.2 Local and global Clouds

An important feature in WAZIUP platform deployment is the possibility to have a local cloud platform. It is an infrastructure able to deliver services to clients in a limited geographical area. The local Cloud replicates some of the features provided by the traditional Cloud. It is used for clients that doesn't have a good access to the traditional Cloud, or to provide additional processing power to local services. In order for such an infrastructure to be considered as a Local Cloud it must support a virtualization technology. In the case of WAZIUP, the local Cloud comprises the end user or service provider PC and IoT Gateway. The local Cloud characteristics are:

- *Existence of IoT devices attached*
- *Can have geographical characteristics*
- *Must support virtualization*
- *Must support local cloud components*
- *Has an identifiable administrator/owner*
- *Has certain regulations/privacy considerations for data access and treatment*

The Global Cloud, on the other end, is a backbone infrastructure which increases the business opportunities for service providers and allows services to access a virtually infinite amount of computing resources. In order for such an infrastructure to be considered as a Global Cloud it must support a virtualization technology and be able to host the global cloud components of the WAZIUP architecture. The figure below present our deployment vision of WAZIUP platform:

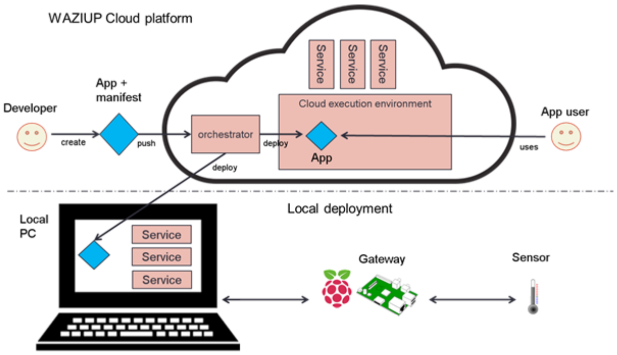

Fig. 7. WAZIUP deployment overview

The Figure 1 displays the WAZIUP deployment overview.

On the left hand side of the picture, the application is designed by the developer, together with the manifest file. It is pushed on the Waziup Cloud platform. The orchestrator then takes care of compiling and deploying the application in the various Cloud execution environments. Furthermore, the orchestrator drives the instantiation of the services in the Cloud, according to the manifest. The manifest is also describing which part of the application need to be installed locally, together with corresponding services. The local application can then connect to the gateway and collect data from the sensors.

Deployment on the local gateway An interested tool for deploying code on the gateway level is resin.io. It facilitate the deploy, update, and the code maintain on remote devices. It is based docker containers and it supports several devices (the Rpi 1, 2 and 3 are supported).

5 Use case applications

The project will build a strong community able to address innovation in the African rural ecosystem through EU ICT technology. Actors from various disciplines, including those having industrial backgrounds, those from application areas and target end-users will push towards the creation of an African-European innovation ecosystems based on the open sources for the IoT-Big data platform. Starting from the rural domain, the project technology will allow the African young application developer to produce new added value supporting multiple applications domains in the long run. The induced ecosystem renews the business paradigm: new benefits are engendered from providing and sharing devices, services, and data across stakeholders and applications.

5.1 Fish farming

Two different test bed scenarios will be used to effectively integrate the real world fish farming situations/solutions with the digital world of WAZIUP. The first test-bed is located in a small farm with 4 mini ponds of 90*90 metres and a large farm which operate several ponds of different sizes[3]. These two farms produces both tilapia and catfish. This use case validation will deploy sensor nodes responsible for sensing three parameters from the fish ponds: dissolved oxigen level, temperature and Ph. The sensor nodes will communicate with the gateways which stores the data locally. If the internet is available through Wifi of GPRS/UMTS, the data will be send to the Cloud instance of Waziup.

5.2 Cattle breeding

With the collaboration of the surrounding farmers we will put sensors around the cows neck in order to have measure related to location and speed. We can also work with CIMEL(Centre dImpulsion et de Modernisations de lELevage) which is a public structure for livestock impulsion and modernisation. With this center we can make test and simulate a cattle rustling case. To make this more interesting we imagine a smart food distribution system on CIMEL for a more optimal management of farmers ressources.

[3] http://www.kumahfarms.com/

5.3 Crop farming

Weather surely plays a crucial role during the crop growth seasons. Severe weather can impact crop growth by affecting different bio-physiological processes. The impact on crop growth depends on the nature of extreme weather event, the crop type and time when it occurs. Detrimental impact of extreme weather events may be alleviated by taking timely and appropriate preventive measures to cope with the nature and intensity of these weather events. The availability of local weather predictions is a key factor in the decision process of the farmers. For example, inputs that are added in the field (such as fertilizer of pesticides) can be lost if rain comes in the next few hours. This use case presents how the WAZIUP platform can be used to prevent and alleviate the risks due to weather in farms. To validate this use case, the WAZIUP project will build a low-cost weather station based on Arduino. The data collected will be processed by the WAZIUP platform and will serve to adjust weather predictions on a local level.

For the deployment of the test bed, two pilots will be conducted in Ghana and Burkina Faso. Both experimental farmer at the university and operational farm environment will be used as test-beds. They will provide the possibility to interact directly with farmers, professionals and students who work on these farms.

5.4 Supply chain management

Transport is a key point of development for Africa. It is very important, for food safety and delivery efficiency, to be able to track the origin and transportation of goods. The test bed will consist of the deployement of a good tracking system. For the sake of simplicity, the system will be based on SMS. SMS is the communication platform that is the most available across African countries, even in remote areas. The system will allow the operators to update the status of the delivery, sending SMS keywords such as "In transit", "Delivered". The client will be updated on the delivery date and location. the client will also be able to interact with the system, for example to change the delivery location.

6 Related works

In this section, we survey the related works. We first analyse similar works from the literature. Regarding the IoT big data aspect, the research have already been largely instanciated inside Open Source frameworks. We present a selection of the most interesting Open Source contributions.

6.1 Review of literature

IoT in Africa Their is very little penetration of IoT in Africa, as evidentiated in [1] and [2]. The authors of [1] provide a survey, country by country, of the undertaking of IoT. They also document some of the challenges affecting adoption of IoT in the continent. Africa has only 7% of her households on the Internet; this is far behind the worlds figure of 41%. Given this lag in the baseline technology needed to implement Internet of Things, the author of [2] advocate for a technological leap and an African-centric approaches to IoT. Taking the case of a drought early warning and assets tracking systems, the author demonstrates that by innovatively incorporating the realities such as the prevalence of African indigenous knowledge on weather, unreliable communication, low-end mobile phone handsets, among others, a home-grown Internet of Things flavour has higher chance of succeeding. An extensive report from Cisco [3] provides also many insights on the current use and potential of Internet of Things (IoT) technologies in tackling global development challenges, highlighting a number of specific instances where IoT interventions are helping to solve some of the worlds most pressing issues.

A deployment of a Wireless Sensor Network for precise irrigation in Malawi is presented in [4]. For the system to be self-sustained in terms of power, the study used solar photovoltaic and rechargeable batteries to power all electrical devices. The system incorporated a remote monitoring mechanism through a General Packet Radio Service modem to report soil temperature, soil moisture, WSN link performance, and photovoltaic power levels. Irrigation valves were activated to water the field. The paper give insights to develop a robust, fully automated, solar-powered, and low-cost irrigation system to suit the socioeconomic conditions of small scale farmers in developing countries.

The authors of [5] provides a survey of possible IoT applications in South Africa and Zambia. In particular, they identify examples of IoTs to mitigate the agricultural needs of these communities for the domains of crop farming, weather forecasting, wildlife management, forestry, livestock farming, market identification and rural financing.

IoT in Agriculture There is not a lot of literature on the specific topic of applying IoT in agriculture, and practically none went it comes to rural Africa. In [6], the author presents a work supporting the transition from traditional agriculture to modern agriculture in China. They propose an agriculture intelligent system based on IOT for organic melon and fruit production. A number of new technologies are used, such as RFID and sensors. They monitor temperature, humidity, light and CO_2 around the crops and use a small model on the fruits of growth process. Always in China, [7] uses Internet of things and RFID technologies to realize automatic control production of agriculture.

In [8], the authors perform temperature control in greenhouse using Zigbee. In [9], the authors elaborate a crop growth model. The model is then embbeded in their IOT application system. This allows them to make the system more intelligent and adaptive for the facility agriculture. The authors of [10] and [11] proposes remote agriculture monitoring and process automation. They are both based on gateway infrastructure and wireless connection. The work in [12] shows a semantically enhanced digital agriculture use case built with the OpenIoT platform.

In [13], the authors uses IoT to check electronically on the vital signs of the cattles. Their tool facilitates the day to day management of dairy activities. It also provides forecastings allowing to handle weather related issues, cattle health and emergencies.

IoT is also deployed within the product supply chain, another key area of agriculture. In [14], the authors builds a quantitative trust model to describe the trustworthiness of foods delivered in supply chains. The Internet of Agricultural Things (AIoT), where the technologies of the Internet of Things are widely used in all of the phases in the agriculture industry, is proposed to resolve the food safety problem. In order to provide a common model to describe and transmit the data in agriculture Internet of Things, [15] proposes a specific ontology, while [16] uses a naming service to identify products.

With difference with the literature surveyed in this section, the proposed Waziup platform is a full IoT platform taylored entierly for African need and constraints. In particular, it as application hosting capacities based on the PaaS paradigm, is resilient to disconnections and provides big data capacities.

6.2 Review of Big Data tools

Far from an exhaustive list, this paragraph describes the most used Open Source Big Data tools and compares them in order to give a better understanding of the Big Data ecosystem. Moreover, this review gives an indication on the best tools fitted for WAZIUP platform.

Databases and data warehouses HDFS, developed by Apache, is a distributed, scalable and portable file-system written in Java for the Hadoop Framework[4]. It has been designed for large dataset analysis and by its structure has high fault tolerance. It is the basis upon which everything works in the Hadoop Ecosystem. Build on top of HDFS, Apache HBase is a distributed, non-relational column oriented datastore[5]. HBase is designed to efficiently address random access and fast record lookup. It has the capability to handle extremely large tables of data with low latency. Though, this data storage tool should be used when random and real-time read/write access to data is needed and when many thousands of operation per seconds need to be performed on large datasets (up to petabytes).

Apache Hive[6] is a data warehouse infrastructure that can manage and query unstructured data as if it were structured. As a full component of Hadoop Ecosystem, it uses MapReduce for execution and HDFS for storage. It has its own language SQL-like (HiveQL) that brings expressiveness to the queries. This storage mode should be used for SQL-like queries and when higher language than MapReduce is needed. Used by big companies who cant afford to lose data (Apple, Netflix, Spotify), Apache Cassandra[7] is a column oriented database of

[4] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[5] https://hbase.apache.org/

[6] https://hive.apache.org/

[7] http://cassandra.apache.org/

structured data. The data are highly available thru column indexes and are automatically replicated thru multiple nodes for fault tolerance. Cassandra has a unique masterless ring design that is easy to setup and to maintain[8].This tools should be use when losing data is not the critical point and not affordable.

First considered as an outsider, getting rid of traditional table-based relational database, mongoDB[9] quickly became a must-have tool: a NoSQL, relational, document oriented database. Document are shared in JSON format with dynamic schemas (called BSON) and makes the integration of data sometimes easier. This database is very useful when you need to consume your data in many applications, as many connectors have been developed.

Data publication and subscription Apache Flume[10] is a distributed, reliable and available service for efficiently collecting, aggregating and moving large amounts of streaming event data. Flume should be used if the data is designed for Hadoop as it can move them to HDFS. It has many built-in sources and sinks and can process data in-flight using interceptors, which is useful for data masking or filtering. It is composed of agents and data collectors (and interceptors if needed).

More general purpose, Apache Kafka[11] is a high-throughput, distributed, publish-subscribe messaging system. It can replicate events, has low latency and is capable of data partitioning. Kafka is also easily scalable and this tool is very useful when the data is to be consumed by multiple applications. It is composed of producer, consumers and topics. Actually, we might not have to choose between Kafka and Flume, as both can work quite well together. If the workflow design requires streaming data from Kafka to Hadoop, using a Flume agent with Kafka source to read the data makes sense. This association is quite common and is called Flafka[12]. If a Data/Context Scenario is developed, we may need to use a context broker. Orion Context Broker[13] (FIWARE platform) is a publish/subscribe platform that is able to register context elements and manage them through updates and queries. It is possible to subscribe to context information when some conditions occurs (e.g. an interval of time passed or the context elements have changed). Orion is a C++ implementation of the NGSI9/10 REST API binding developed as a part of the FIWARE platform.

Data processing Obviously, choosing a data processing tool depends mostly on the outcome expected from the data. The most common tool for BigData analysis, and what we probably think at first, is Hadoop MapReduce[14]. It has proven its efficiency in many ways and is an incredible tool. But if we want to step a

[8] http://www.planetcassandra.org/what-is-apache-cassandra/

[9] https://www.mongodb.com/

[10] https://flume.apache.org

[11] http://kafka.apache.org

[12] http://blog.cloudera.com/blog/2014/11/flafka-apache-flume-meets-apache-kafka-for-event-processing/

[13] http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

[14] http://hadoop.apache.org

bit aside of Hadoop workflow or if we have specific needs, other tools exists and some are becoming more and more powerful.

But first, lets talk about this milestone Hadoop MapReduce. MapReduce programming model contributed to the amazing progress of BigData processing this past decade. By breaking down the work and recombining it in series of parallelizable operations, it is simple but incredibly efficient and scalable to ten thousands of machines if needed. It can run on inexpensive hardware, lowering the cost of a computing cluster. The latest version of MapReduce is YARN, called also MapReduce 2.0.

If a higher level of programming on top of MapReduce is needed, Apache Pig[15] is the one. Pig has its own language (PigLatin) similar to SQL and works on top of MapReduce. Pig Engine parses, optimizes and automatically executes PigLatin scripts as a series of MapReduce jobs on a Hadoop cluster. Its easy to learn and opens Hadoop to data professionals who may not be software engineers.

First designed to work with HDFS on top of YARN, Apache Spark[16] is a different system for processing data and can work out of Hadoop ecosystem with other data managements systems. It does not work with MapReduce and it can be up to a hundred times faster than MapReduce with its capacity to work in-memory, allowing to keep large working datasets in-memory between jobs, reducing considerably the latency. What makes it more and more attractive to many users worldwide, is its wide range of applications: batch and stream processing (micro-batch processing with 0.5s latency), machine learning (MLib), SQL (with Hive), graph Analytics (graphX). Language supported are Java, Python and Scala.

Demand for stream processing becoming more and more important in Big Data analysis, Apache Flink[17] has been recently developed and is growing very fast. Flink is a streaming dataflow engine that provides data distribution, communication and fault tolerance. It has almost no latency as the data are streamed in real-time (row by row). It runs on YARN and works with its own extended version of MapReduce. Language supported are Java and Scala.

Machine learning Machine learning is the union between statistics and artificial intelligence. It blends AI heuristics with advanced statistical analysis. We let the machine learn about the data, make decisions, and then apply statistics. Algorithms used for this tasks can be grouped in 3 domains of actions: Classification, association and clustering. To choose an algorithm, different parameters must be considered: scalability, robustness, transparency and proportionality. Overlearning (or overfitting) of the model must be carefully checked.

Without any math or programming requirement, KNIME[18] is an analytic platform that allow the user to proceed the data in a user-friendly graphical interface. It is a good tool to train your model and evaluate different machine

[15] http://pig.apache.org
[16] http://spark.apache.org
[17] http://flink.apache.org
[18] http://www.knime.org

learning algorithms rapidly. If the workflow is already deployed on Hadoop, a machine learning library exists and is called Mahout[19]. Spark also has his own machine learning library called MLib[20]. H20[21] is a software dedicated to machine-learning, which can be deployed on Hadoop or Spark (Flink in development). It has an easy to use Web interface, which makes possible to combine big data analytics easily with machine learning algorithm to train models.

Data visualisation and exploration To visualise the data in real time, Freeboard provides a simple, real-time dashboard, commonly used in IoT world[22]. There is a direct Orion Fiware connector. To connect with streaming engines, a JSON connector can be used. Design is simple and customisation is not possible, but it is a very good dashboard to visualise easily raw data coming from sensors, before data analysis.

Tableau Public[23] offers a good visualisation and exploration tool on batch data. Tableau is a software where you can upload your analysed data (previously extracted in .csv format). The visualisation tool is very powerful and allow a deep exploration the data. However it is not designed for really Big Data with large datasets and the open Source version of Tableau (Public) does not offer the data streaming capacities (e.g. Spark connectors). Nevertheless, Tableau Public is a highly customisable, user-friendly and intuitive exploration tool for data that have already been processed and analysed.

To visualise data in real-time, after analysis (filtering, aggregating, correlating), one of the best tool is probably Kibana[24]. It is the visualisation tool coming with ElasticSearch. Elasticsearch is a search server based on Apache Lucene that provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. It is really designed for real-time analytics, most commonly used with Flink or Spark Streaming.

7 Conclusion

References

1. Nashon Onyalo, Hosea Kandie, and Josiah Njuki. The internet of things, progress report for africa: A survey. *International Journal of Computer Science and Software Engineering*, 2015.
2. M. Masinde. Iot applications that work for the african continent: Innovation or adoption? In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pages 633–638, July 2014.
3. ITU. Harnessing the internet of things for global development. Technical report, ITU, 2015.

[19] http://mahout.apache.org

[20] http://spark.apache.org/mllib/

[21] http://www.h2o.ai/

[22] https://freeboard.io/

[23] https://public.tableau.com/s/

[24] https://www.elastic.co/fr/products/kibana

4. Million Mafuta, Marco Zennaro, Antoine Bagula, Graham Ault, Harry Gombachika, and Timothy Chadza. Successful deployment of a wireless sensor network for precision agriculture in malawi. *International Journal of Distributed Sensor Networks*, 2013.
5. N. Dlodlo and J. Kalezhi. The internet of things in agriculture for sustainable rural development. In *Emerging Trends in Networks and Computer Communications (ETNCC), 2015 International Conference on*, pages 13–18, May 2015.
6. Fu Bing. Research on the agriculture intelligent system based on iot. In *2012 International Conference on Image Analysis and Signal Processing*, pages 1–4, Nov 2012.
7. Fan TongKe. Smart agriculture based on cloud computing and iot. *Journal of Convergence Information Technology (JCIT)*, 2013.
8. L. Dan, C. Xin, H. Chongwei, and J. Liangliang. Intelligent agriculture greenhouse environment monitoring system based on iot technology. In *Intelligent Transportation, Big Data and Smart City (ICITBS), 2015 International Conference on*, pages 487–490, Dec 2015.
9. Xiangyu Hu and Songrong Qian. Iot application system with crop growth models in facility agriculture. In *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on*, pages 129–133, Nov 2011.
10. .. .. Nakutis, V. Deksnys, I. Jaruevicius, E. Marcinkevicius, A. Ronkainen, P. Soumi, J. Nikander, T. Blaszczyk, and B. Andersen. Remote agriculture automation using wireless link and iot gateway infrastructure. In *2015 26th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 99–103, Sept 2015.
11. Prosanjeet J. Sarkar and Satyanarayana Chanagala. A survey on iot based digital agriculture monitoring system and their impact on optimal utilization of resources. *Journal of Electronics and Communication Engineering (IOSR-JECE)*, 2016.
12. P. P. Jayaraman, D. Palmer, A. Zaslavsky, and D. Georgakopoulos. Do-it-yourself digital agriculture applications with semantically enhanced iot platform. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, pages 1–6, April 2015.
13. A. Ilapakurti and C. Vuppalapati. Building an iot framework for connected dairy. In *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on*, pages 275–285, March 2015.
14. W. Han, Y. Gu, Y. Zhang, and L. Zheng. Data driven quantitative trust model for the internet of agricultural things. In *Internet of Things (IOT), 2014 International Conference on the*, pages 31–36, Oct 2014.
15. Siquan Hu, Haiou Wang, Chundong She, and Junfeng Wang. *AgOnt: Ontology for Agriculture Internet of Things*, chapter AgOnt: Ontology for Agriculture Internet of Things, pages 131–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
16. Y. Liu, H. Wang, J. Wang, K. Qian, N. Kong, K. Wang, Y. Shi, and L. Zheng. Enterprise-oriented iot name service for agriculture product supply chain management. In *Identification, Information and Knowledge in the Internet of Things (IIKI), 2014 International Conference on*, pages 237–241, Oct 2014.