SAFE - Sensor & Actuator for Contextual Services and Predictive Maintenance
A Low Cost Industrial IoT Implementation for Industry 4.0

Tuan Dang[1] - Pascale Minet[2] - Patrick Bellot[3] - Christophe Mozzati[4]
Erwan Livolant[2] - Dragutin Brezak[3] – Mohamed T. Hammi[3]

Abstract. In this paper, we present a complete IoT (Internet of Things) system, starting from a network of smart sensors and actuators named OCARI to applications ready to use such as the KASEM predictive maintenance system or DISPLAY, a simple web application for consulting sensors results. To ensure interoperability and a high level of standardization, a customized but compatible OPC-UA industrial middleware implementation, named OPC-UA/ROSA, allows secured communications. It also acts as a distributed contextual services discovery mechanism.

1. Introduction

Industry 4.0 paradigm is the next industrial revolution in which smart sensors and actuators with wireless connectivity capability will deeply change the way industrial installations work. Process monitoring will be more smart and predictive thanks to low cost and reliable measurement capability that can be quickly implemented and deployed. Contextual services may then be easily implemented using the information enriched by pervasive sensors. In this scenario, implementing a suitable middleware that helps to securely orchestrate the information flow is essential.

In this paper, we present a low cost Industrial IoT system that implements sensors and actuators for contextual services and predictive maintenance [4]. This work has been done for the Cluster CONNEXION project headed by EDF from 2012 to 2016. See http://www.cluster-connexion.fr for more information. The complete IoT system for industry 4.0 is depicted in Figure 1. This system is built using:

- OCARI wireless sensors network protocol that is used to transform existing sensors and actuators into smart wireless instrumentation.
- OPC-UA/ROSA, an extended OPC-UA industrial standard middleware that is used to manage and to orchestrate contextual services.
- KASEM, a commercial software framework for predictive maintenance.
- Open source AJAX API chart library from Google Developers for displaying sensors measurement in Web browser/Smart phone.

---

[1] EDF R&D Lab, STEP Department, 78401 Chatou.
[2] INRIA, 2 rue Simone Iff, 75012 Paris.
[3] CNRS LTCI & Télécom Paristech, Université Paris-Saclay, 46 rue Barrault, 75013 Paris.
[4] PREDICT, 19 av. De la Forêt de Haye, 54519 Vandoeuvre-les-Nancy.

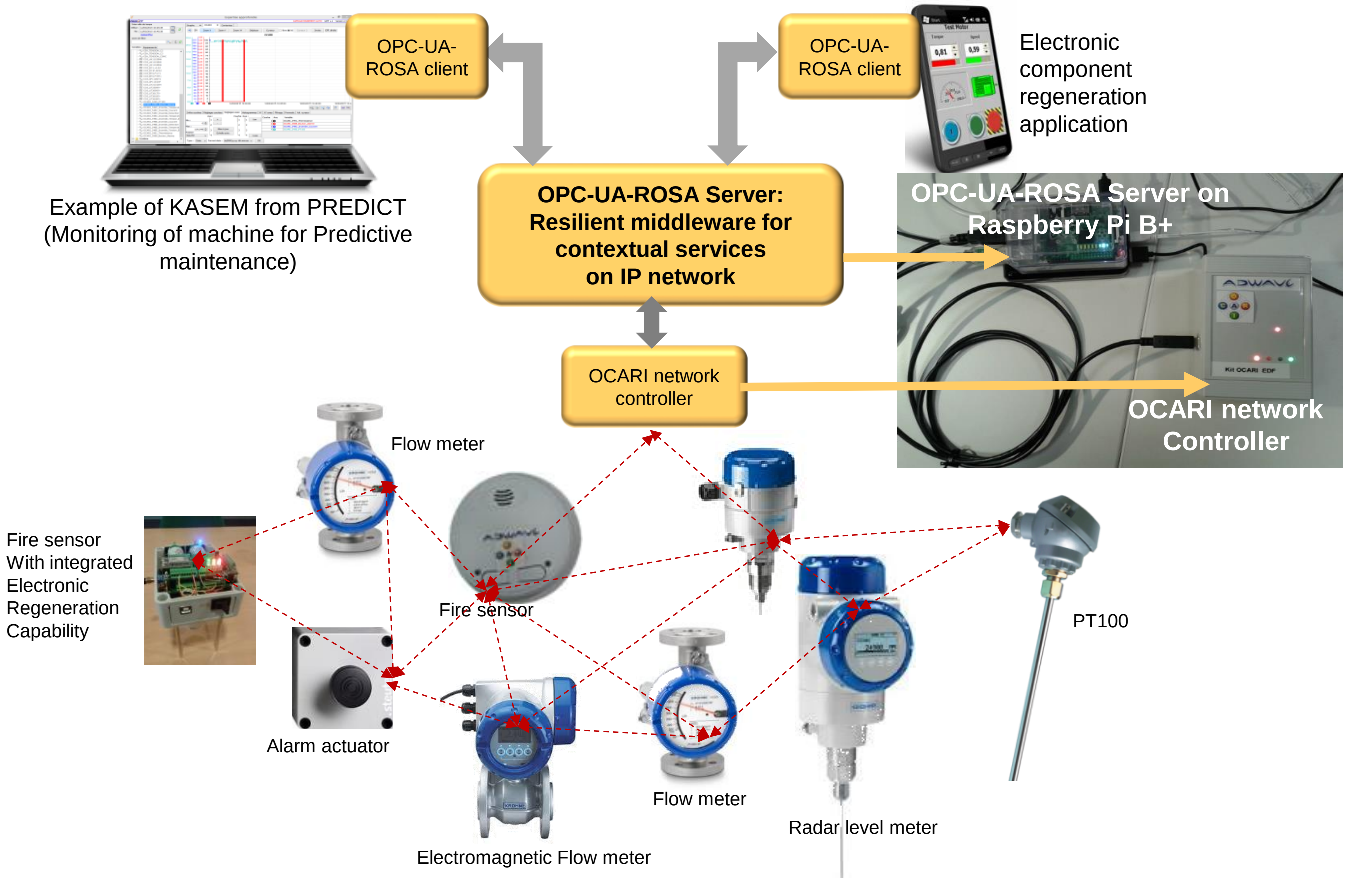

Figure 1: A complete IoT system

We detail the different system components of this IoT system in the following sections.

2. The OCARI wireless sensors network

OCARI (**O**pen wireless **C**ommunication protocol for **A**d hoc **R**eliable industrial **I**nstrumentation) [1] is a low-rate wireless personal area network (LR-WPAN) communication protocol that derives from the IEEE 802.15.4 standard. It is developed to address wireless sensor and actuator application in industrial processes with the following requirements:

- Reliable communication protocol for process monitoring and non-critical control as classified by ISA (International Society of Automation) as in Figure 2 below.
- "Plug-and-play" features so that people can deploy themselves the instrumentation network without the need of training.
- Minimum maintenance of the field transmitter (only battery replacement).
- Durability of the transmitter technology during the lifetime of industrial installation (typically up to 30 years).
- Reuse/retrofit of existing instrumentation to Industrial IoT (Internet of Things) in the context of Industry 4.0 paradigm.
- Ability to use the same transmission technology for a wide range of application beyond the process monitoring and control to achieve the scale effect.

OCARI was then designed to satisfy the following technical requirements:

- Running on standard IEEE 802.15.4 platforms (so low cost may be achieved; currently the following modules are supported: Dresden Electronik deRFsam3-23T09-3/23M09-3, Adwave Adwrf24-LRS and HiKoB Fox).
- Providing mesh networking with self-configuring and the power saving capability for operation on battery.
- Ability to support a large number of instrumentations per application with the same hardware platform (no specific transmitter so that interchangeability is possible).

Category	Class / Application	Description
Safety	Class 0: Emergency action	Always critical
Control	Class 1: Closed loop regulatory control	Often critical
	Class 2: Closed loop supervisory control	Usually non-critical
	Class 3: Open loop control	Human in the loop
		NOTE: Batch levels* 3 & 4 could be class 2, class 1 or even class 0, depending on function *Batch levels as defined by ISA S88; where L3 = unit and L4 = process cell
Monitoring	Class 4: Alerting	Short-term operational consequence (e.g., event-based maintenance)
	Class 5: Logging and downloading / uploading	No immediate operational consequence (e.g., history collection, sequence-of-events, preventive maintenance)

Importance of message timeliness increases

OCARI

Figure 2: Application classes addressed by OCARI

With the above requirements in mind, the OCARI communication protocol was developed with the following features:

- Energy-efficient proactive and adaptive routing (the path to reach the sink has a minimum energy cost, and new paths are automatically created when the existing ones are broken, and only paths built from symmetric links are retained) and load balancing of router nodes (the node that has the highest residual energy among the one-hop neighbors that are closer to the sink than the node considered is dynamically selected as its next hop to the sink).
- Distributed synchronization of the operating cycle based on multi-hop deterministic synchronization of nodes using cascaded beacons. It allows to determine the sleeping period of all network nodes for energy saving.
- An activity scheduling mechanism based on a distributed three-hop coloring algorithm that minimizes the number of colors (pre-reserved slots). Thanks to this mechanism, extra energy saving can be obtained because of no collision and

a node wakes up in its slot if it has data to transmit and in the slots of its 1-hop neighbors if it has data to receive, and sleeps the rest of the time.

- Spatial reuse of the time slots (the 4-hop nodes can reuse the same color and therefore transmit at the same time). This facilitates the scalability of network per application.

More precisely, the OCARI operating cycle is organized into 5 periods as illustrated by the following figure:

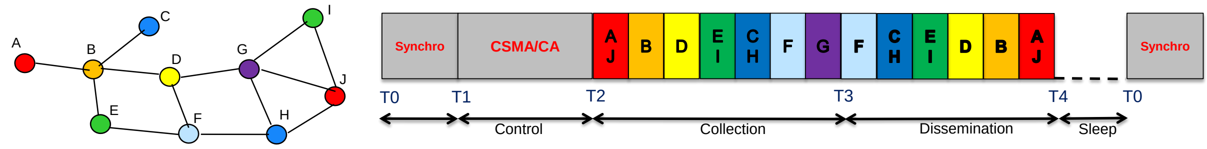

Figure 3: OCARI operating cycle

- [T0-T1]: Multi-hop deterministic synchronization of nodes using cascaded beacons.
- [T1-T2]: Transmission of messages and signaling data by competition (CSMA/CA).
- [T2-T3]: Transmission of data messages without collision (collection) in colored slots (optimized TDMA).
- [T3-T4]: Transmission of data messages without collision (dissemination/order) in colored slots.
- [T4-T0]: Sleep.

3. The OPC-UA/ROSA industrial middleware

OPC-UA is an industrial middleware protocol for M2M communications ensuring security and interoperability [2]. OPC-UA servers provide a lot of services amongst which Data Access (OPC-DA), Historical Data Access (OPC-HDA), Alarms and Events (OPC-A&E) are the most distinctive.

From an architectural point of view, there is a clear separation between the communication stack and the application itself. Applications communicate using asynchronous messages. The stack is in charge of serializing the messages, handling the security and transporting them. OPC-UA proposes different stacks. If you serialize in binary (called UA-BINARY), you handle the security with X.509 certificates and transport the message using UA-TCP, and you get the most efficient stack. UA-TCP maintains open a TCP socket between the two participants meanwhile with security you can sign and/or encrypt the message with different key lengths. Another type of stacks uses SOAP and XML with WS* security and HTTP or HTTPS transport to provide a service oriented architecture.

The specification of OPC-UA is so dense that it is difficult to read, however two implementations with compatible stacks are totally interoperable. An OPC-UA server implementation may provide different stacks to meet different clients.

We mixed OPC-UA with ROSA [3] to provide a resilient, distributed and contextual discovery service. Of course, this kind of service is not part of OPC-UA. OPC-UA has a Service Discovery service but it is much more static than what we provide. The current contextual information for services allows retrieving them. This contextual information system is adapted for nuclear plant industry but could be re-engineered for another kind of industry. ROSA is an overlay network in which there exists Distributed Hash Tables (DHT). These DHTs are resilient. They are used for the distributed implementation of our contextual discovery services. Contextual discovery services are provided as an extension to OPC-UA services.

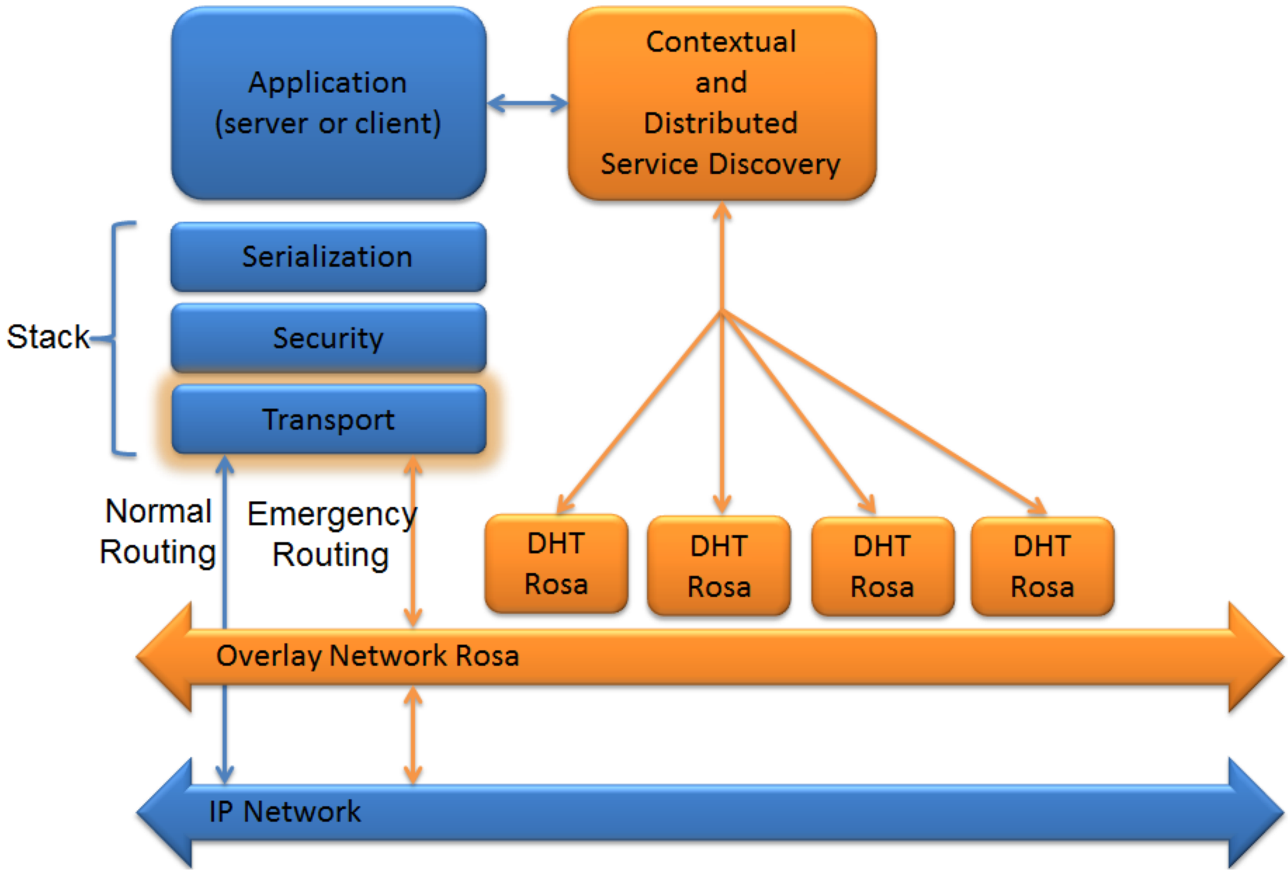

Figure 4: Architectures of OPC-UA and ROSA

4. The predictive maintenance system KASEM

We now present the KASEM software from PREDICT. KASEM is a predictive maintenance tool that computes health assessment indicators and detects drifts or abnormal behavior. It contains a web portal, an acquisition service, a computation engine. To do so, it needs the widest possible set of sensor data, using any communication technology available, wired or wireless. Therefore, we decided to use OPC UA to connect to various data sources, and ROSA for dynamic discovery of wireless network sensors.

An OPC-UA/ROSA client is developed and integrated inside the KASEM acquisition architecture. This client subscribes to the variables tags exposed by the OPC-UA/ROSA server. The embedded OCARI gateway updates the tags each time a new value is read from the wireless sensor network. New values are retrieved every second in KASEM. A data visualization tool is embedded inside the KASEM web portal as shown in Figure 5. The service discovery mechanism of ROSA is used to dynamically update the available variable list when a new OCARI sensor joins the network.

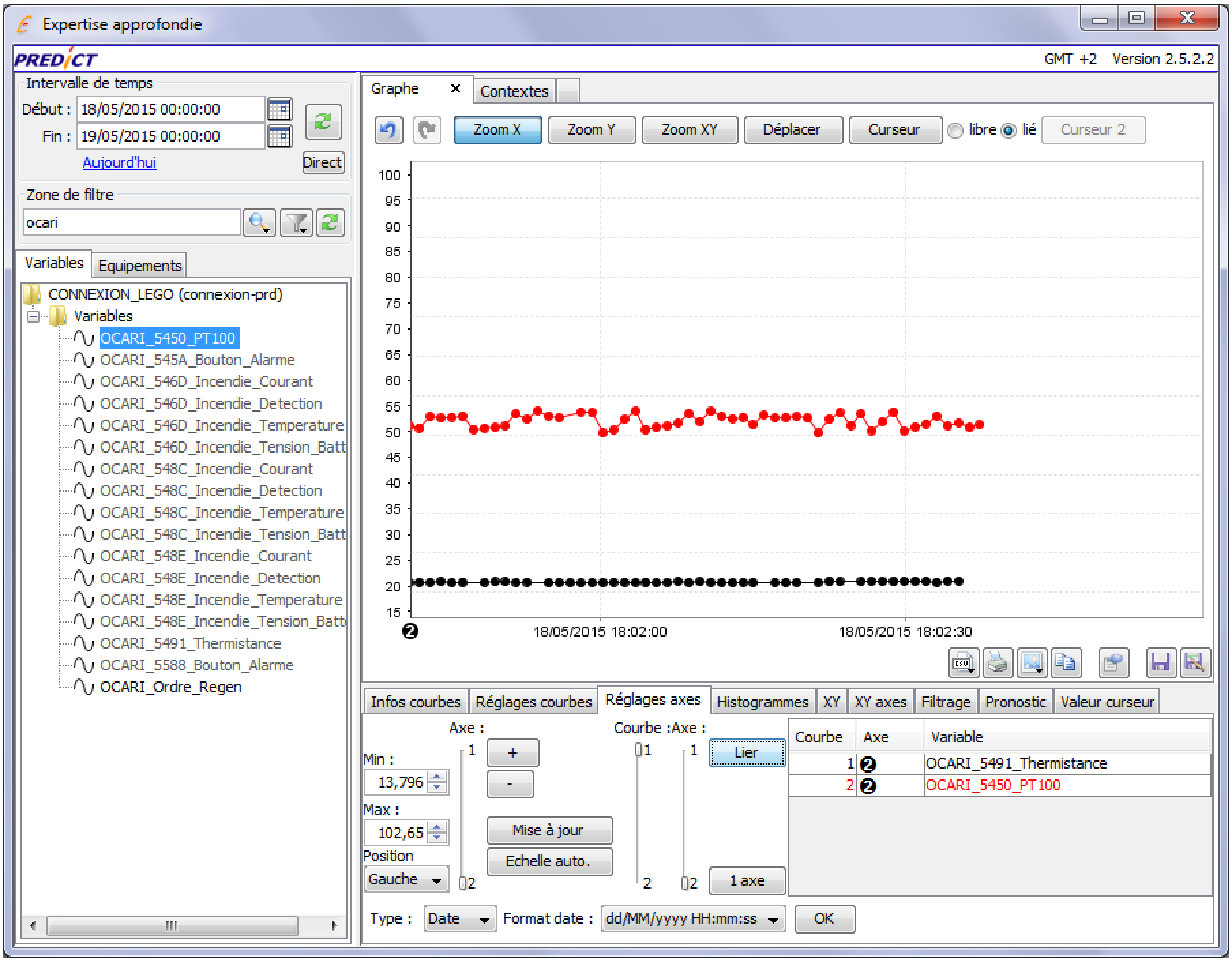

Figure 5: KASEM data visualization tool

A built-in computation engine in C# is integrated for KPI computation, event triggering and report generation. KASEM provides a library of mathematical models dedicated to health assessment computation and drift detection. A typical computation includes signal inputs, parameters, reports and signal outputs. Computation can be arranged in sequences, to divide a complex algorithm into several simple steps (see Figure 6).

The KASEM knowledge base stores every data point received by KASEM from its OPC UA client and every output from the computation engine. KASEM uses a SQL database as a back-end to store this data. Each data point is stored with its timestamp, its status (good, bad value, error, overflow...) and its value.

Finally, an OPC UA server has been built to let other clients subscribe to outputs and indicators and read them. This server also exposes reports files generated by KASEM directly as byte array OPC nodes.

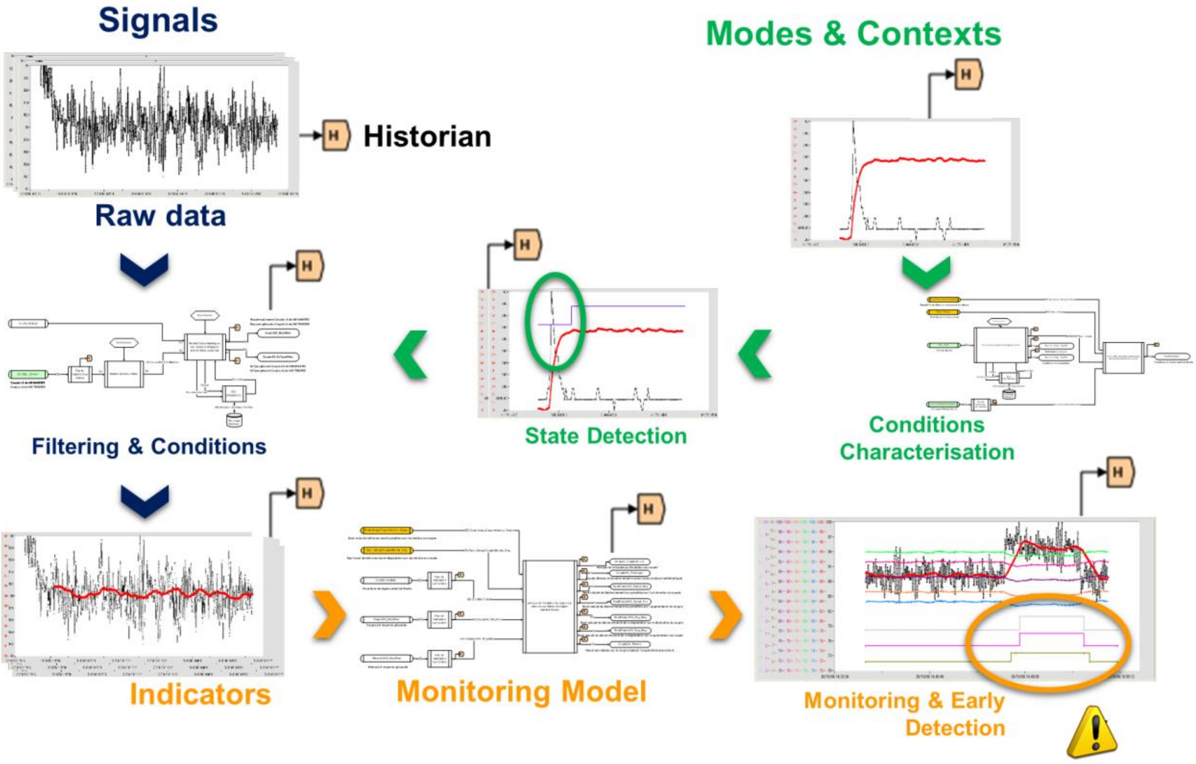

Figure 6: KASEM data processing steps: from raw data to early detection

5. The web application for displaying sensors results DISPLAY

We now present DISPLAY, a web application for displaying the results of the OCARI sensors. On the Raspberry, we have added a USB Wifi dongle so that any smartphone can access the Apache server. We have designed a simple OPC-UA client that feeds the Apache server and a JavaScript program that allows the animated display of curves. With all that, a smartphone can connect to the Raspberry and dynamically display the measurements of a sensor, as illustrated in Figure 7.

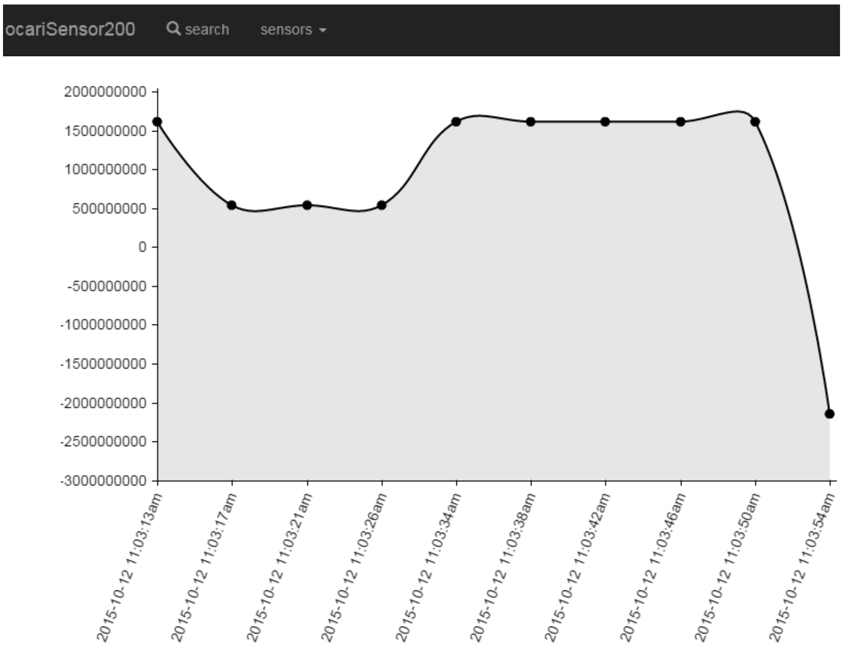

Figure 7: Screen copy of a smartphone displaying an OCARI sensor

6. Conclusion

In this paper we have presented a complete IoT system for Contextual Services and Predictive Maintenance. It is based on Open Source software components as well as commercial off-the-shelf software framework for predictive maintenance, and on low cost/standard hardware modules. Through this experience, we have shown that how easy it is to implement and deploy an industrial IoT using the existing instrumentation. We plan to release OCARI and OPC-UA/ROSA to the Open Source community in the end of 2016. We call for participation and contribution to the development of these technologies in order to accelerate the implementation of Industry 4.0 paradigm.

7. Bibliography

[1] Dang, T., Minet, P., Livolant, E., *OCARI: A wireless sensor network for industrial environments*, ERCIM News 2015(101) (2015). http://ercim-news.ercim.eu/en101/special/ocari-a-wireless-sensor-network-for-industrial-environments.

[2] OPC-UA Foundation, *OPC-UA Specification, Part 1 to Part 12. Release 1.01*, (2009).

[3] Baud, L., Pham, N., Bellot, P., *Robust overlay network with self-adaptive topology: Protocol description*. In: 2008 IEEE Int. Conf. RIVF, Ho Chi Minh City, Vietnam, July 2008, pp. 154–160 (2008).

[4] T. Dang, P. Bellot, P. Minet, E. Livolant et C. Mozzati: *Architecture orientée services : une clé pour faciliter l'intégration de nouveaux points de mesure pour la performance des architectures de maintenance.* Génie Logiciel, n° 112, pp. 25-31, mars 2015.