# User-Friendly and Scalable Platform for the Design of Intelligent IoT Services: a Smart Office Use Case

Pieter Bonte, Femke Ongenae, Jelle Nelis, Thomas Vanhove, and Filip De Turck

Ghent University - iMinds, iGent-Toren, Technologiepark-Zwijnaarde 15, 9052 Gent , Belgium
Pieter.Bonte@intec.ugent.be

**Abstract** The Internet of Things (IoT) is starting to take a prevalent role in our daily lives. Smart offices that automatically adapt their environment to make life at the office as pleasant as possible, are slowly becoming reality.
In this paper we present a user-friendly semantic-based smart office platform that allows, through easy configuration, a personalized and comfortable experience at the office.

## 1 Introduction

In the Internet of Things (IoT) paradigm, multiple sensor and devices are connected to the Internet. The captured data of these sensors is used to facilitate intelligent decision making regarding their environment. The IoT aims at creating intelligent systems that support people as much as possible during their daily activities. Semantic technologies are the preferred mechanism for modeling the IoT data [4]. In a smart office setting, the goal of the IoT is to make the employees life at the office as pleasant as possible. The attached IoT sensors in the office sense the environment and transmit the captured data to a back-end for further processing. When decisions have been made regarding the captured data, the back-end can decide to interact with the sensors to change the environment in a sense that is optimized for the employee's office experience.

In this paper we present a smart office use case, implemented on a semantic-based, user-friendly and adaptable IoT platform. The platform simplifies the development of IoT services by limiting the amount of code the user has to write. Through the use of a visual interface, the logic of the IoT services can be defined.

## 2 Use Case

In the presented use case, employees are provided with a wristband, such as the Mi Band[1], enabling the office to detect their presence. When the employees are detected near their office, the door automatically unlocks, the lights turn on and the windows open based on the preferred office temperature of the employee.

When a meeting is planned with external visitors, they can also enjoy the optimized experience provided by the IoT. At arrival at the reception, the visitors are asked for the name of their contact person. The visitors receive a temporary batch for identification and are guided towards to the elevator that will bring them to the correct floor. In the

[1] http://www.mi.com/en/miband/

meantime, the contact person has been notified about the arrival of his visitors. Once the visitors arrive at the correct floor, their presence is detected and lightning strips attached to the floor guide them to the correct meeting room where their contact person is already waiting for them. Through the use of semantic technologies, the roles of the employees and the visitors can be modeled and use for automated decision making. It also allows each employee to model their personalized office environment and to easily integrate external sources.

## 3 Used platform

To implement the described scenario, a generic user-friendly platform that simplifies the creation of IoT services is utilized. The generic platform itself is composed of three existing frameworks, each specialized in its field. Fig 1 depicts how the three frameworks cooperate in the described use case. In the right bottom corner, the dashboard is shown that gives a real-time overview of the deployed service, the active sensors and their state. The following paragraphs describe the used platforms.

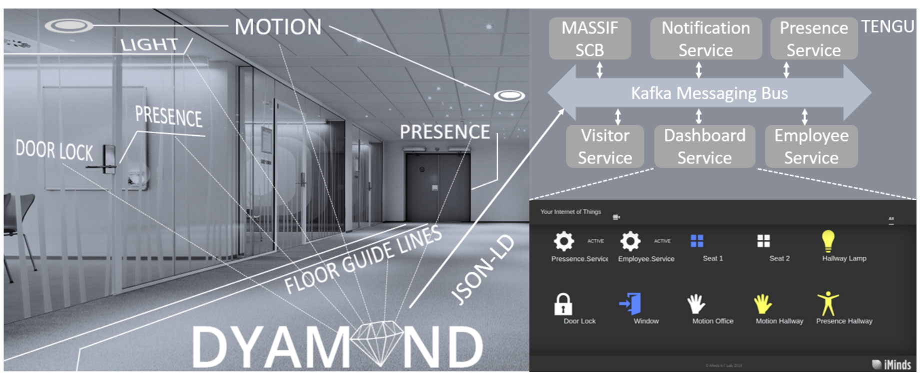



**Figure 1.** Architecture of the used platform, mapped to the smart office use case and an overview of the dashboard.

**DYAMAND**[2] [3] is an IoT sensor integration platform that enables easy integration, interaction and discovery of multiple sensors using various technologies. It facilitates the easy integration and usage of any kind of sensor. The internal model of the platform is mapped to the Semantic Sensor Network (SSN) [2] ontology allowing a standardize representation of sensor data. The captured sensor readings are transmitted to the MASSIF platform, that runs on top of Tengu, in JSON-LD format.

**MASSIF** [1] is a platform that facilitates the development and deployment of semantic services that can collaborate to tackle complex reasoning tasks. These services each fulfill a distinct reasoning task and share their conclusions over a Semantic Communication Bus (SCB). They can subscribe to the data on the SCB by passing an OWL Axiom

[2] https://dyamand.ilabt.iminds.be/

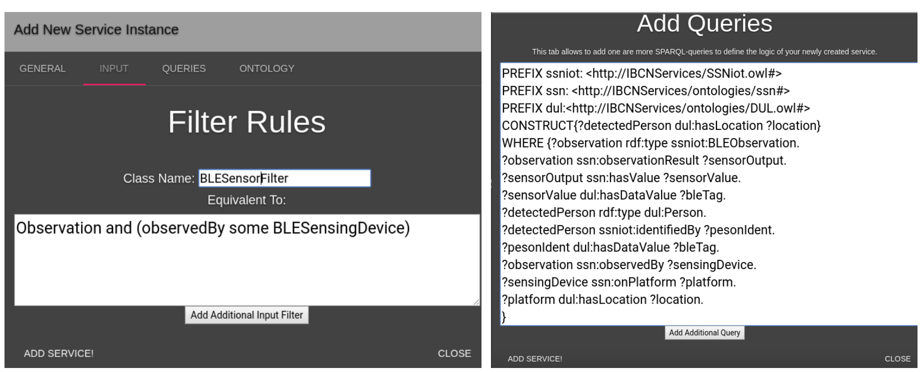
describing their data of interest on a high-level. Since each of these services are loosely coupled over the SCB, these service can easily be distributed and duplicated to achieve scalability. To cope with generic sensor data, the platform is able to annotate raw sensor data to the semantic model.

A visual interface is provided to simplify the definition of these services. One can launch a new service by i) providing the ontology the service should operate on, ii) defining the input data the service should subscribe to and iii) provide one or more SPARQL-queries to define the logic of the service. By using the SPARQL CONSTRUCT query form, the query results is a RDF graph that models the conclusions of the services. The results of the CONSTRUCT query serve as the results that can be shared with the other services, that are interested in this kind of data, over the SCB.

**Tengu**[3] [5] is a Big Data framework that allows the easy deployment of Big Data frameworks. It takes reliability into account by logging messages and recovering of failed instances. Scalability is taken into account by providing duplication and load balancing protocols. Tengu uses juju charms[4] to deploy the various components in the cloud. To allow easy set up of MASSIF, various charms were written to set up the MASSIF platform and a Kafka[5] message bus that takes the communication across various nodes into account.

## 4 Implementation

To implement the described scenario, various MASSIF services were created. We assume that all used sensors are already supported by DYAMAND and are thus automatically detected and ready to be used.




(a) Input data example

(b) Query example

**Figure 2.** Example service definition through the visual interface.

The following paragraph describes the created services to support the use case:

[3] http://tengu.intec.ugent.be/

[4] https://jujucharms.com/

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

- The Presence Service subscribes to all sensor location observations that indicate the localization of a device. The loaded ontology contains the modeling of the floor plan, sensor readings, etc. Inside the loaded ontology, a mapping is provided that links the employees to their device. When a location update is captured, the service links it to the correct person and shares the knowledge that the detected person is at the specific location with the other services over the SCB. An example of the input and the executed query definition through the visual interface can be found in Fig 2.
- The Employee Service subscribes to all location updates from the Presence Service and sensor updates. The loaded ontology models the offices in a generic way and defines the employee's preferences. Through the use of predefined actuators in the SPARQL-queries, the windows can be opened if the temperature is to high, the doors unlocked when an employee is in the presence of the office and the lights turned on and off.
- The Visitor Service describes the visitors logic. This service also subscribes to the location updates and checks if the detected person is a visitor. This service integrates the calendar of the employees, to know where the meetings take place. The service can activate the floor guidelines towards the correct meeting room.
- The Notification Service is a predefined services that captures the data from the other services that have indicated the need to interact with the environment, e.g. turning on the light. It will communicate with DYAMAND to execute the interactions.
- The Dashboard Service is also a predefined service and takes care of the visual overview of the active sensor and allows the deployment and configuration of new services, as visualized in Fig 1.

## 5 Demonstrator

The demonstrator will focus on the easy definition of new IoT services and the sensor interaction within the smart office use case. The dashboard, as depicted in Fig 1, will be used to provide an overview of whats going on and to easily configure the IoT services. The demonstrator visualizes the full use case in a simplified form for easy demonstration. The used sensors and devices consist of Mi Bands, motion sensors, pressure sensors and lights. For simplicity, the floor guidelines will be mocked by lights.

**Acknowledgment:** This research was made possible by the DiSSeCt Strategic Fundamental Research (SBO), funded by VLAIO.

## References

1. Bonte, P., et. al: The MASSIF Platform: a Modular & Semantic Platform for the Development of Flexible IoT Services . Knowledge and Information Systems (2016)
2. Compton, M., et. al: The ssn ontology of the w3c semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web 17, 25–32 (2012)
3. Nelis, J., et. al: Dyamand: dynamic, adaptive management of networks and devices. In: Local Computer Networks (LCN), 2012 IEEE 37th Conference on. pp. 192–195. IEEE (2012)
4. Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. In: Workshop Proceedings (2004)
5. Vanhove, T., et. al: Tengu: An experimentation platform for big data applications. In: 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops. pp. 42–47 (2015)