Embedded Intelligence in Smart Cities through Multi-core Smart Building Architectures: research achievements and challenges

Basil Nikolopoulos, George Dimitrakopoulos, George Bravos, Alexandros Dimopoulos, Mara Nikolaidou and Dimosthenis Anagnostopoulos

Harokopio University of Athens, department of Informatics and Telematics

9, Omirou str., 17778, Athens, GREECE, e-mail: basil@hua.gr

***Abstract*— Economic growth in Europe has been, strongly associated with urbanization, overwhelming cities with vehicles. This renders mobility inside cities problematic, since it is often associated with large waste of time in traffic congestions, environmental pollution and accidents. Cities struggle to invent and deploy "smart" solutions in the domain of urban mobility, so as to offer innovative services to citizens and visitors and improve the overall quality of life. In this context, the paper discusses on the fundamental challenges that cities face when trying to become smarter, focusing on the particular area of smart buildings and the management of multi-core, smart building architectures and presents some key research achievements and relevant research challenges.**

***Index Terms*—smart cities, smart city operations, smart buildings, reconfigurable, management**

I. Introduction

It is widely accepted that citizens inside large cities at a worldwide level are "bombed" by large amounts of uncorrelated and non-synchronized data, from innumerable sources and through various devices in a complex manner. Citizens are thus not in position to efficiently handle them, this resulting in severe inefficiencies associated with their mobility, such as (i) fragmented travel solutions / lack of door-to-door solutions, especially when dealing with multimodal transportation, as well as (ii) inadequateness in providing real-time, whilst individualized services. Those drawbacks often result in losses of time, decrease in the level of safety in mobility, pollution, degradation of life quality, and huge waste of non renewable fossil energy. Moreover, they affect not only citizens, but all relevant stakeholders, such as also public authorities and businesses.

The contribution of this paper is that it defines a representative use case for the management of multi-core reconfigurable architectures in the context of smart buildings, which in turn can help with the stated issues.

II. The Multi-core Context-aware Smart Building Management Architecture

A. High level architecture view

Our envisioned architectural framework is depicted in Figure 1. It must be noted that our architecture supports plug and play connectivity and is designed to be scalable, self-optimized, and multi-layered.

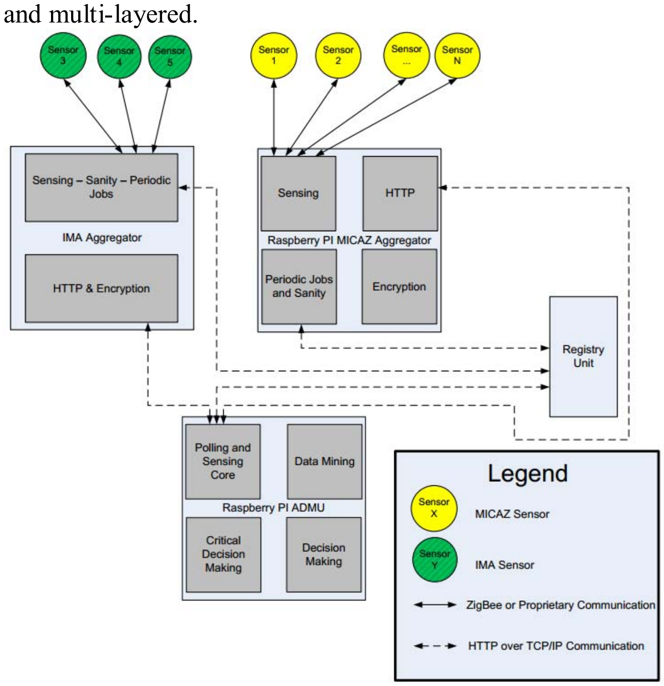

Figure 1: The Overview of the envisioned architecture

Starting from the lowest layer, the sensing units are found which report to an authority (aggregator) that is responsible for providing web based RESTful services. Each different sensor most likely uses different technology for communicating, and hence for each different technology one different aggregator is assigned. Therefore, each set of same technology sensors is responsible for reporting its own data to a designated data aggregator. At the next layer, that of the aggregators, various different procedures are serviced, e.g. keeping an up to date list of available sensors, making sensors' data available through the RESTful services and communicating with the sensors. The importance of the aggregators is apparent as it adds a layer of abstraction over the underlying WSN. It converts the obfuscated communication patterns of the WSN into a more reliable and easy to use format, over standard TCP/IP HTTP. Each aggregator unit must provide RESTful services in the standard way. A very important aspect of the communication is the secure transmission of data from the aggregators to the next layer. It must be clarified that the encryption concerns the communication between the aggregators and the next layer, where aggregated data are transmitted; the communication between the sensors and the aggregator is usually over a proprietary communication protocol and hence cannot be changed by a third party.

One layer above the aggregators is the middleware of the Registry Unit. Since there are many aggregators with different IP addresses, which all employ the TCP/IP stack to provide their RESTful services over HTTP, one should be able to discover available aggregators without much trade-off. Hence, that need is covered by the Registry Unit. The registry unit is responsible for keeping an up to date list of the services available from the aggregators and is contacted by the aggregators in order to register their services. Moreover the aggregators can update and delete any service they own. Additionally, the registry unit is responsible for polling said registered services to check on their availability and if not remove them from its list.

The top layer is that of the autonomic decision making unit (ADMU). This component adds the required intelligence on the proposed system, by deciding on actions to be carried out from smart agents, through their services. The decisions are based on policies set by third parties, like users or energy efficiency policies for instance, by taking into consideration user feedback/preferences, the context information from the sensing units and information from the pattern repository.
Based on the above description, it is clear that in the case of either the aggregator or the ADMU, the system must run more than one task, namely sensing, sanity checks, and encrypted communication with the higher layer for the aggregator and polling/sensing, data mining, encrypted communication with the lower layer, and decision making for the ADMU. All the above tasks and especially the encryption/decryption phase of the communication are so computationally intense for a simple uniprocessor embedded system that would definitely lead to lags and latencies. Luckily, nowadays there are many options for multi-core embedded systems, which allow the smooth concurrent execution of all tasks.

Regarding the registry unit, it constitutes a simplified DNS-like service, which acts as discovery service for the users currently entering the smart building and a registry service for the autonomous components currently available.

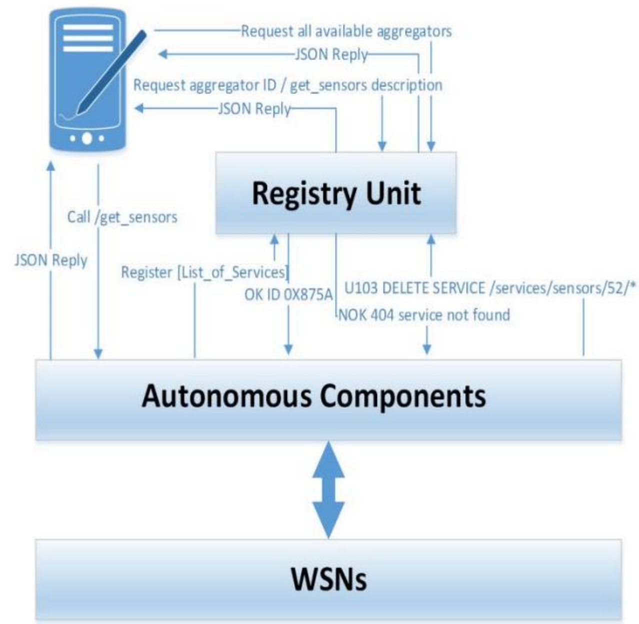

Figure 2: The role of the registry unit as seen from a service oriented point of view

B. The ADMU unit

The main idea behind ADMU is the development of the system-level schemes fostering autonomous functionalities as the technological solutions for mixed-criticality problems. The solutions are based on the autonomic networking concept, utilizing the newest results from several research areas, including dynamic distributed systems, self-management, self-healing, self-organizing and other self-x related management concepts, artificial intelligence, and machine learning.
In particular, having as inputs: contextual information (i.e. number of sensors), profile information (capabilities of sensors, profiles), as well as policies information (restrictions to the solution space invoked by the user or a regulator), autonomic decision making algorithms will be able to decide on prioritizing a service against another service, or changing the parameters of a service provided, so as to provide the user with the maximum QoS levels possible, whilst satisfying a set of pre-defined criteria.

III. ACKNOWLEDGMENT

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement n° 621429.

IV. REFERENCES

[1] Walravens, N., Ballon, P., “Platform business models for smart cities: from control and value to governance and public value”, Communications Magazine, IEEE, Volume: 51 , Issue: 6 ,, 2013, pps. 72 - 79

[2] Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., & Williams, P., “Foundations for Smarter Cities”, IBM Journal of Research and Development, 54(4).

[3] http://www.hitachi.com/products/smartcity/download/pdf/whitepaper.pdf , accessed March 16th, 2015

[4] http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/ , accessed February 26th, 2015