Advanced Engineering Informatics 33 (2017) 243–257

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

Platform as a service gateway for the Fog of Things

Nandor Verba [a,*], Kuo-Ming Chao [a], Anne James [a], Daniel Goldsmith [a], Xiang Fei [a], Sergiu-Dan Stan [b]

[a] Faculty of Engineering, Environment and Computing, Coventry University, Coventry. United Kingdom
[b] Faculty of Mechanical Engineering, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

ARTICLE INFO

Article history:
Received 1 July 2016
Received in revised form 8 November 2016
Accepted 21 November 2016
Available online 30 November 2016

Keywords:
Internet of Things
Gateway
Fog Computing
Horizontal integration
MQTT
OSGI
Cloud manufacturing

ABSTRACT

Internet of Things (IoT), one of the key research topics in recent years, together with concepts from Fog Computing, brings rapid advancements in Smart City, Monitoring Systems, industrial control, transportation and other fields. These applications require a reconfigurable sensor architecture that can span multiple scenarios, devices and use cases that allow storage, networking and computational resources to be efficiently used on the edge of the network. There are a number of platforms and gateway architectures that have been proposed to manage these components and enable application deployment. These approaches lack horizontal integration between multiple providers as well as higher order functionalities like load balancing and clustering. This is partly due to the strongly coupled nature of the deployed applications, a lack of abstraction of device communication layers as well as a lock-in for communication protocols. This limitation is a major obstacle for the development of a protocol agnostic application environment that allows for single application to be migrated and to work with multiple peripheral devices with varying protocols from different local gateways. This research looks at existing platforms and their shortcomings as well as proposes a messaging based modular gateway platform that enables clustering of gateways and the abstraction of peripheral communication protocol details. These novelties allow applications to send and receive messages regardless of their deployment location and destination device protocol, creating a more uniform development environment. Furthermore, it results in a more streamlined application development and testing while providing more efficient use of the gateway's resources. Our evaluation of a prototype for the system shows the need for the migration of resources and the QoS advantages of such a system. The examined use case scenarios show that clustering proves to be an advantage in certain use cases as well as presenting the deployment of a larger testing and control environment through the platform.

1. Introduction

The concepts from the Internet of Things (IoT) are of key interest to researchers and industry leaders [1]. New initiatives like Germany's Industry 4.0 [2], as well as concepts from [3] consider the interconnection of devices as the fourth industrial revolution which is estimated to result in over 21 billion devices connected to the Internet by 2020 [4]. Such systems would require homogeneous and interoperable Machine to Machine (M2M) networks that can be accessed from the Internet while abstracting unwanted details and enabling higher level application development and better use of resources.

There are multiple problems and directions that can be taken to create a fully Automated Manufacturing Environment. The IoT oriented components focus on the orchestration of resources that are needed to make products, reducing time to market, manufacturing times and idle devices and resources. Previous research has proposed various solutions. In [5] model-based task and deployment method is suggested while in [6] a Service Oriented Manufacturing (SOM) solution is presented for mapping and access. Furthermore, the platforms presented in [7,8] suggests the collaboration of a number of higher-level systems to meet these requirements.

When discussing the IoT we can consider three distinct approaches for the architectures of such systems [9]. An IPv6 based network where devices are uniquely accessible through Constrained Application Protocol (CoAP) or other lightweight protocols has been suggested in [10], while in the cloud oriented approach devices are accessed through API's [11] or using Message Queue Telemetry Transport (MQTT) protocol. The middleware approach is based on gateways or brokers that communicate with devices

* Corresponding author.
E-mail addresses: verban@coventry.ac.uk (N. Verba), csx240@coventry.ac.uk (K.-M. Chao), csx118@coventry.ac.uk (A. James), aa9863@coventry.ac.uk (D. Goldsmith), aa5861@coventry.ac.uk (X. Fei), sergiu.stan@mdm.utcluj.ro (S.-D. Stan).

http://dx.doi.org/10.1016/j.aei.2016.11.003
1474-0346/

through more lightweight communication protocols such as 6LoWPAN, nRF24L01 or ZigBee and forward these messages to the cloud or other clients such as in [12].

With the introduction of Cloud and Fog Computing paradigms, the use of resources available at the edge of the network is considered as well as the deployment of application and processing tasks on the edge devices [13]. Proposals like MADCAT in [14,15] suggest large applications be decomposed into components and deployed onto devices while [16] suggests a MapReduce like approach with IoT application development. Together with proposals from [17] which looks at reconfigurable components and [18] which looks at agent based cooperative smart objects, these suggest a need for Software Defined Networking as well as the need of decomposing applications into components and running them on the gateway.

The idea for the use of resources on the edge of the network was first introduced by Cisco [19]. Advances in Networking as a Service (NaaS) and the increased processing power of gateway devices have led to the development of edge computing platforms like Docker [20]. Edge computing includes solutions based on Virtual Machines [21,22] as well as container based application deployment solutions.

Open Service Gateway Interface (OSGI) is a modular service platform for Java that has been the focus of research towards modular IoT Gateways. OSGI can be used for multi-tenant cloud connection architecture as in [23]. Platforms like HEPA [24] propose the use of Zookeeper to control a set of OSGI Gateways that would facilitate the transmission and translation of device information. One of the drawbacks of the OSGI core platform is that it lacks solutions for asynchronous communication between components. To address this issue [25,26] proposed an a messaging based solution that maps messages to either internal services or to an event administration system component.

The requirements presented in [2] for the industrial use of IoT Devices as well as innovations in supply chain management presented in [27] show that the existing gateways and solutions require extensive research on the horizontal integration of devices and gateways to allow for multiple devices to send and receive messages between each other regardless of the underlying protocol. Existing gateways focus on a more vertical approach where systems are built upon one single platform or language, which results in system lock-in and reduces the number of use cases and overall functionality of the gateway. Vertical integration is concerned with integrating devices and protocols from a single provider and expanding their capabilities and scope, while horizontal integration focuses on providing inter-platform support and cross-communications solutions to users so that resources from various suppliers can interoperate.

When discussing message passing the scenarios that need to be considered are Machine to Machine (M2M), Machine to Gateway (M2G), Machine to Cloud (M2C) and their combinations as described in [28]. The translation and forwarding of these messages is difficult, due to the existing protocol fragmentation in the industry as well as due to each protocol having their own advantages and specific use cases. There is a need for translation of messages from different protocols as suggested in [29] but the bigger issue is to allow for applications to be deployed in a protocol agnostic environment where all the layers of communication are abstracted away from applications. This is a feature that is missing from existing gateway architectures that merely provide means to directly address physical resources but fail to provide higher level abstraction of device communication.

The exploitation of smart gateways for Wireless Sensor Network (WSN) use cases such as smart-office and wide area monitoring requires that devices, and their events and controls be available from a wide variety of locations to allow for more complex applications to be deployed without the need to redesign them. This can only be done if we consider a local cluster of gateways that share information and devices, which allows for redundancy, load balancing and high availability. Current gateways allow for a horizontal implementation of similar features that would enable applications components to communicate with each other. While this solution is sufficient for certain implementations, they lack an implicit implementation of clustering which would allow messages and events to be passed without the need to program migrations and the deployment of applications to multiple gateways.

We propose a messaging oriented gateway architecture that allows for multiple application containers and drivers to be deployed and connected on the same device while enabling application migration. This architecture would allow multiple connections to different providers with different privileges as well as a regional clustering and the use of local resources such as storage and location services.

This gateway architecture would allow the horizontal integration of multiple platform devices through the connection of the messaging service and the abstraction of protocol specific information. Messages would be passed from multiple protocols and drivers from different gateways to one application deployed to one gateway, reducing the impact of protocol fragmentation as well as creating a coherent application environment inside the cluster allowing for message routing from one application to another without these having to be configured for regional communication. These additional functionalities reduce the complexity of singular applications and allow larger systems to be deployed without individual gateway (re)configuration. Such an environment is needed in order to be able to deploy applications to devices without platform lock-in or other prohibitive factors.

The rest of the paper is organized into a state of the art chapter, a chapter dedicated to describing the proposed architecture, one for implementation and performance evaluation and a conclusion and future work chapter. Section 2 evaluates the state of the art and investigates the requirements of horizontal integration of gateways. Section 3 presents the proposed architecture, showing how it meets the requirements identified in Section 2, as well as explaining its construction. The final two sections look at evaluating the performance of the Gateway, together with use case scenarios that highlight the added functionalities, followed by the conclusions drawn from these as well as future work.

2. State of the art

IoT gateways have become increasingly configurable and their functionalities have expanded. The horizontal integration directives aim at allowing platforms and devices from different providers using different protocols to interact. This would increase reusability and reduce application complexity. To allow the connection of multiple devices we need multi M2M protocol support, as well as registration, management and an enhanced configurability for these devices. The increased number of resources available on the gateway has led to the need to be able to virtualize these and move a portion of the resource use from the cloud to the edge of the network.

There are a number of different approaches to the design and implementation of IoT gateways. Most of the initial approaches as well as some of the latest ones like Eclipse Scada, Krikkit, SmartHome and HePA [24] concentrate on semantic interpretation of data and configuration based routing or event creation. Other approaches like that of Kura and Eliot look at fully reconfigurable systems where applications configure and define everything, a fully modular system. These approaches cause platform and provider