Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 337–342

Proposal of Component Based Architecture for Internet of Things: online laboratory case study

Miroslav Gula and Katarína Žáková

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava, Slovakia
(e-mail: miroslav.gula@stuba.sk, katarina.zakova@stuba.sk)

Abstract: The paper introduces new generic architecture for Internet of Things (IoT) applications with focus on online experimentation. The proposed architecture was inspired by many common features shared between IoT applications and online laboratories. The architecture follows principles of component based design of software systems, and reusability of components. The main goal of the proposed architecture and its implementations is to simplify and accelerate future development and deployment of new online laboratories and IoT applications as well.

Keywords: Internet of Things, Online Laboratories, Web Services, REST, WebSocket protocol.

1. INTRODUCTION

In the Oxford English Dictionary phrase Internet of Things is defined as: "The interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and receive data". By this definition every experiment device connected to the Internet and therefore every experiment running via the Internet can be classified as IoT application. Such experiments can include measurement of various variables such as temperature, force, position, pressure, etc. (via sensors) and the control of these variables (via actuators). Online experiments can be part of scientific research and development and also part of technical education, as well.

In the past online laboratories (remote and virtual) were usually created as monolithic and rigid software for performing remote experiments on specific equipment. Each newly added feature or equipment usually required extensive software changes in existing application. Sometimes required changes were so complex that it was easier to develop completely new system from scratch. Current trends in development of online laboratories tend to shift from creating specific, one purpose applications towards designing more generic and reusable architectures for online laboratories, as can be seen in Muros-Cobos and Holgado-Terriza (2012), Odeh (2010), Olmi et al. (2011a,b), Richter et al. (2011), Salzmann et al. (2015). In this way, it is much easier and faster to develop and deploy new laboratories, and it is also much easier to extend or modify already existing laboratories based on such architectures.

The proposed architecture, described in this paper, is inspired by component based architecture and software design (Heineman and Councill, 2001; Lüders, 2003; Szyperski, 2002). A component based architecture can be characterized as set of loosely coupled, often distributed, software components. Software systems built with component based architecture in mind are much more flexible in terms of extensibility and modifiability. Loose coupling of components also largely simplifies swapping of one component implementation to another. Mentioned features are highly desirable in the context of online laboratories, because these features can greatly simplify development, deployment and management.

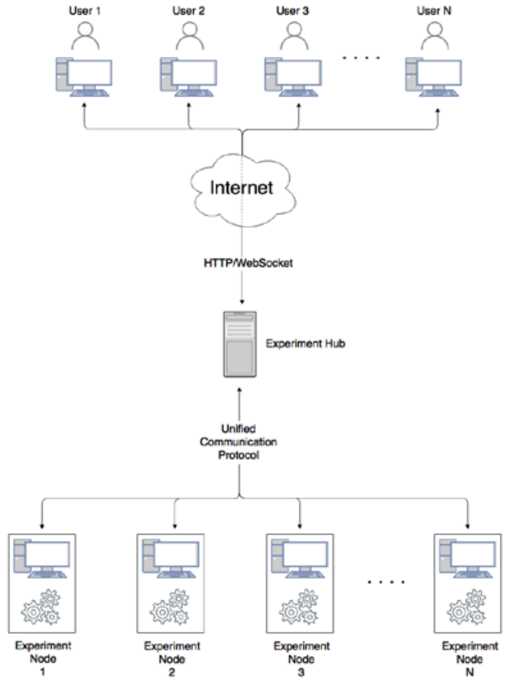

Fig. 1. Diagram of proposed architecture.

2. OVERVIEW OF PROPOSED ARCHITECTURE

The proposed architecture can be seen in Fig. 1. It consists of several separate and distributed components which are loosely coupled together using standard web technologies.

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.153

The architecture provides two main components: Experiment Hub and Experiment Node. Experiment Node encapsulates all software and hardware needed to run the experiment, and provides access to the experiment as to one whole unit. Experiment Hub serves as glue to the whole architecture. It contains information about connected experiments (Experiment Nodes). It also serves as access point and provides environment for an end user to access and execute experiments. Furthermore, Unified Communication Protocol is used to abstract communication between Experiment Hub and Experiment Nodes. This way any kind of experiment can be connected to Experiment Hub, as long as it adheres to communication standards introduced by Unified Communication Protocol. In the following sections, we will describe each individual component in more details.

3. EXPERIMENT HUB

Experiment Hub serves as central access and management point of the whole online laboratory. For the end user, it is the only visible part. The whole functionality of online laboratory is provided in the form of standard web application, so the end users do not need to install any additional software on their computers. Only software they need is the standard web browser. Communication between end users and Experiment Hub is carried out using two application-level protocols: HTTP (HyperText Transfer Protocol) (Fielding et al., 2014a,b,c,d,e,f) and WebSocket protocol (Fette and Melnikov, 2011). HTTP protocol is used for standard communication (e.g. fetching web pages from Experiment Hub), and WebSocket protocol is used for indirect communication with Experiment Node during the experiment execution. WebSocket protocol was used because it provides some advanced features compared to HTTP, which are suitable for environment of remote experiments. These features will be described in more details later in this article.

Main responsibilities of Experiment Hub can be divided into three categories:

- to provide end users with user interface, so they can interact with online laboratory,
- to maintain database of all managed experiments,
- to relay commands and data between end user and the experiment (Experiment Node).

Mentioned responsibilities are fulfilled by following key services:

1) *Experiment management*, i.e. adding, removing, and editing of remote experiments by administrators of online laboratory.

2) *Experiment browsing*, i.e. providing list of currently present experiments in online laboratory and providing experiment details to end users.

3) *Experiment access policy* – Experiment Hub has to provide policy which determines method of access to one experiment by multiple users at the same time. One way is to provide scheduling system, where each user has to make reservation of time slot when he or she is allowed to access experiment. Another way is to make experiment shareable amongst multiple users. In this case, concurrent modification of experiment by multiple users has to be properly handled.

4) *Security* – Experiment Hub is only component directly accessible from the Internet, so all security concerns are focused on it. It has to provide standard set of authentication and authorization mechanisms, so each experiment is accessible only by permitted users. Experiment Hub should also provide mechanism for communication encryption.

5) *Experiment execution* – Experiment execution is the most essential functionality provided by online laboratory. During experiment execution Experiment Hub servers as proxy, so all data sent between end user and experiment (Experiment Node) go through Experiment Hub. This means that Experiment Hub is able to check and intercept invalid values of parameters sent from users to an experiment. Experiment Hub is also able to reduce amount of sampled data, which are streamed from experiment to the end user. For example, if an experiment produces too much data, Experiment Hub can reduce this amount by sending only each n-th sample.

6) *Data recording* – Experiment Hub is able to log and archive all data sent to an experiment by the end user and produced by the experiment during its execution. In this way, it is possible to record the whole experiment session and play it back step by step in “offline” mode after experiment has ended.

4. EXPERIMENT NODE

Experiment Node encapsulates all hardware and software of remote experiment. By implementing Unified Communication Protocol, Experiment Node provides unified access point to the whole experiment as to one independent unit. Communication abstraction introduced by Unified Communication Protocol provides opportunity to integrate wide variety of remote experiments into online laboratory. For better understanding of different needs of different types of remote experiments, we split remote experiments into three separate classes: remote experiments based on standard PC (personal computer), remote experiments based on single board computer, and remote experiments based on microcontroller.

4.1 Remote experiments based on standard PC

Using personal computer in conjunction with appropriate hardware and software is nowadays probably the most preferable solution for creating remotely accessible experiments. This was also, historically, the first solution ever used. General architecture of remote experiment based on personal computer is depicted in Fig. 2.

Since personal computers do not contain specialized I/O (input/output) hardware required for direct communication with sensors and actuators of an experiment, it is necessary to use additional, specialized I/O hardware. More precisely, this specialized hardware is responsible for translation of signals produced by sensors into digital data, which can be processed by personal computer, and also vice-versa, for translation of digital data generated by personal computer into signals for actuators of the experiment. This hardware can be implemented in the form of DAQ (Data Acquisition) card,