

Master thesis : MQTT broker with in-line, real-time data visualiser for the Internet of Things (IoT)

Auteur : Detienne, Martin

Promoteur(s) : Leduc, Guy; 12788

Faculté : Faculté des Sciences appliquées

Diplôme : Master en sciences informatiques, à finalité spécialisée en "computer systems security"

Année académique : 2021-2022

URI/URL : <http://hdl.handle.net/2268.2/14496>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

University of Liège
Master in Computer Science
Academic year 2021-2022

MQTT broker with in-line, real-time data visualiser for the Internet of Things (IoT)

by **Martin Detienne**

Supervisors: **Guy Leduc** (Uliège) and **Emmanuel Tychon** (Cisco)

Internet of Things (IoT) devices are more and more used in our today's life. However, as the amount of connected devices increases, the communications between them are more and more difficult to scale. The MQTT communication protocol has come to solve this issue. The MQTT protocol is a communication protocol between IoT devices that aims to centralise the communications inside one broker. However, a typical MQTT communication involves hundreds of messages per minute. This is then hard to analyse an MQTT communication with packet sniffer tools like Wireshark. The goal of this master thesis is to provide an appropriate and publicly available tool to visualise and analyse MQTT communications for debugging purposes. This report reviews the goals of this project: create several MQTT broker environments on the same system, display the set of published messages, the state of a broker, filter the published messages, and display telemetry on the number of published messages. This thesis report then goes through the MQTT protocol and its various types of packets. The high-level architecture is discussed with the first decisions for the role of each component that will be developed: a broker to record the MQTT data and a viewer to visualise it. We then decide on the type of database to store the recorded data persistently, the kind of data to record which is interesting to display to the user, and how this data should be stored. The architecture of the broker is reviewed as well as the technique used to manage the several independent MQTT broker environments. With this, we discuss how to save the events that happen in an MQTT broker and the techniques used to scale the project with a big amount of MQTT messages. To conclude the discussion about the architecture, we discuss the plugin system that allows extending the viewer, how the MQTT data should be stored and represented in the viewer, and how the user should have access to the required features. We then review how complex filters can be built into the viewer. We then discuss the development methodology of this project. The implementation details are then reviewed with the different technologies used in this project. We will talk about the difficulties encountered during the deployment and explanations on how to deploy this project easily. The results of this project are finally reviewed with the latencies encountered when the broker faces a lot of traffic and some screenshots of the viewer. A discussion is made about the further possible improvements on this project. Additional documents and documentation are provided at the end of this report.