Master Thesis of Science HES-SO in Engineering

Low-complexity scalable loT framework

Lucas Bonvin

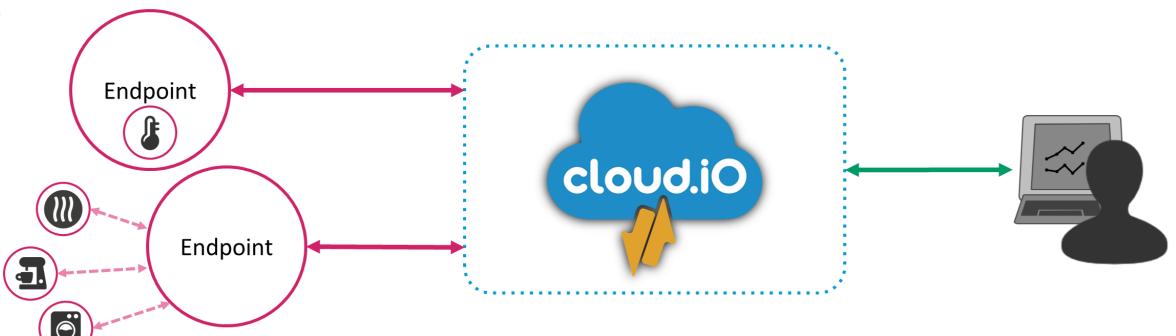
Supervising professor: Dominique Gabioud

HES-SO Valais

DESCRIPTION

Internet of Things (IoT) is a continuously growing field. According to Cisco's predictions, we can expect data exchange generated by IoT reaching 20.6 Zettabytes by 2021. IoT frameworks manage this enormous amount of data, and the leading solution is cloud-based architecture. The main goal of those IoT clouds is to connect different Things of an IoT environment to the web.

The Institute of Systems Engineering at HES-SO Valais Wallis did develop its cloud base solution called cloud.iO. cloud.iO works with a simple architecture composed of Users and Endpoints. And Endpoint can be considered as an IoT gateway for multiple things or a thing supporting cloud.iO. This Endpoint sends its state to the cloud and is managed by Users. The structure of a cloud.iO Endpoint is the following; it starts with the Endpoint, which can contain one or multiple nodes. These nodes can be compared to a Thing or functionality of a Thing in the IoT world. Such a node contains objects whose can have either other objects or attributes as its children.



Simple cloud.iO Architecture with Endpoints and User

This framework has been developed through research projects and reached a mature version, the so-called v0.1. The development of cloud.iO still presents a problem; it is developed for research project need and not according to the IoT landscape need. The primary motivation of this thesis is: making cloud.iO evolves according to the IoT landscape and create a v0.2.

OBJECTIVES

This thesis is based on 3 main objectives:

- Analysis of the open-source IoT landscape and positioning of the new cloud.iO v0.2 in this landscape.
- Collection of requirements for cloud.iO v0.2 (from potential users) and elaboration of a corresponding specification.
- Development, documentation, test, and validation of cloud.iO v0.2.

The proposed project aims to develop a new version "v0.2" for cloud.iO. While keeping the strength of the first version, the new design should simplify the commissioning of new things, ease the development of things firmware, and facilitate the integration into legacy IoT environment.

From the analysis of cloud.iO v0.1 and the IoT landscape, the following features are taken back and are implemented for cloud.iO v0.2:

- Redesigning the Endpoint and User Access Control
- Allowing Remote Job execution on the Endpoint
- Logging system from the Endpoint to the cloud
- Certificate Management by the cloud
- RESTful API covering all the features of cloud.iO
- Making cloud.iO compliant to Web Of Thing candidate recommendation about IoT

RESULTS

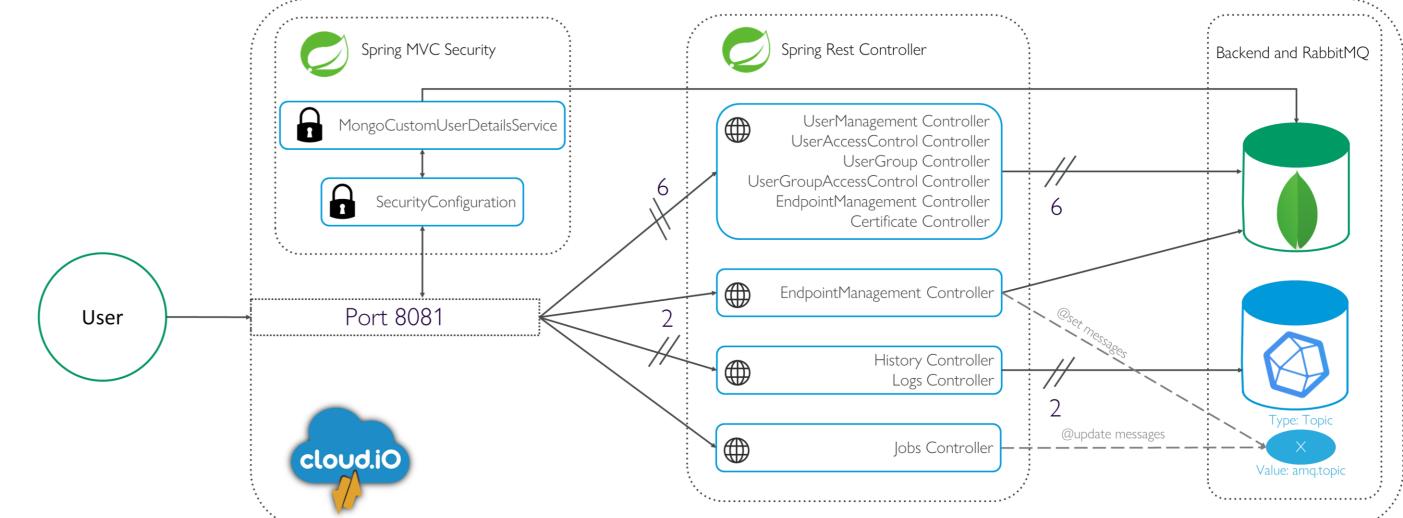
Access control: In cloud.iO v0.1, the access control was limited to permissions on Endpoints. In cloud.iO v0.2, we introduced the possibility to give permissions up to the Attribute. To simplify the access control, the concept of User Group was also added, giving the possibility to manage permissions of multiple Users.

Remote Jobs Execution: When an Endpoint is deployed, its access may be restricted. To allow the execution of simple task on it without the need of specific network infrastructure, cloud.iO v0.2 introduce the concept of Remote Jobs Execution. A user can specify through a specific MQTT message or through an HTTP request the name of a local script to be executed on the Endpoint. The User can then retrieve the outputs of the scripts.

Logging: cloud.iO v0.2 gives the possibility for developers to use Log4j to create logs and send them directly to the cloud via MQTT. Each Endpoint has a log level that filters which logs are sent to the cloud. It is possible to retrieve the logs easily with the RESTful API.

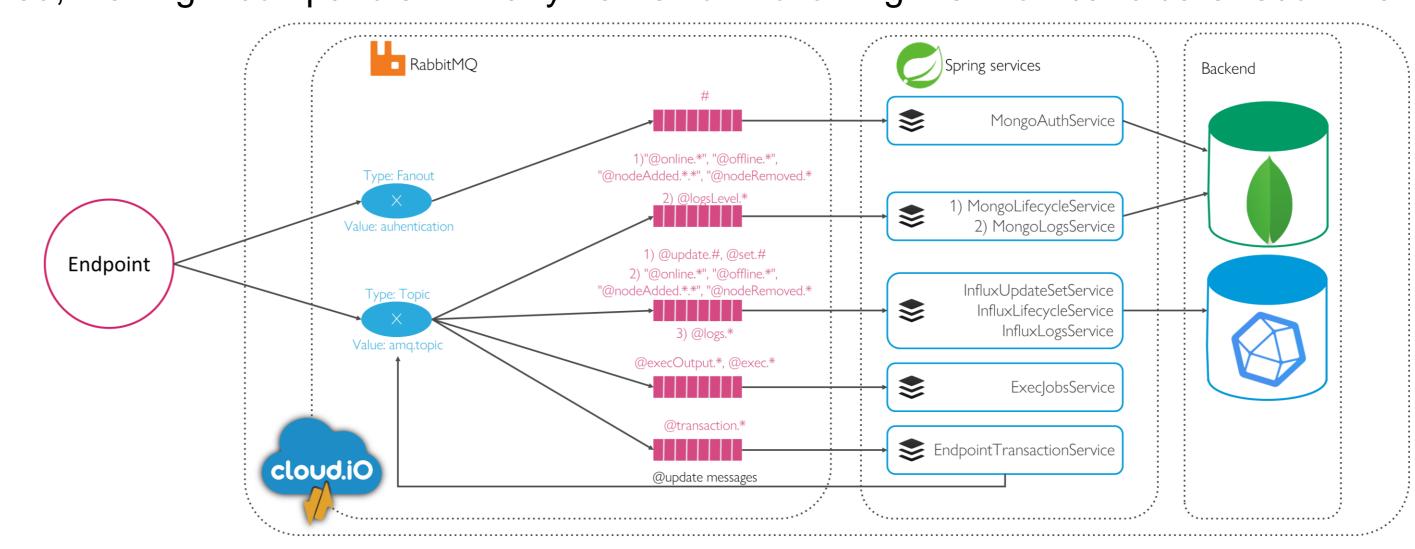
Certificate Management: Endpoint communication is using MQTTs (TLS) and authentication using an x509 certificate. In cloud.iO v0.1, the generation of certificates was made manually using a make file. In cloud.iO v0.2, certificates can be generated by the cloud using the RESTful API.

RESTful API: In cloud.iO v0.1, no tools were existing to do User or Endpoint management; everything was done manually with database tools. Also, the only way to communicate with cloud.iO was by using M2M protocols. In cloud.iO v0.2, we introduced a complete RESTful API covering all those themes: User Management, User Access Control, User Group, User Group Access Control, Endpoint Management, Certificate, History, Logs, and Remote Job Execution.



Architecture of cloud.iO v0.2 core linked to RESTful API

Web Of Things Compliance: cloud.iO v0.2 has been made compliant with the Web of Thing (WoT) candidate recommendation by W3C. WoT has for main goal to enhance interoperability across different IoT applications, frameworks, devices, etc. This interoperability is achieved by the introduction of the Thing Description (TD). A TD, in JSON-LD, is a device twin containing a self-describing API to access the Things resources. Any Things connected to cloud.iO can have its TD retrieved, making it compatible with any framework following the WoT candidate recommendation.



Architecture of cloud.iO v0.2 core linked to messaging

CONCLUSION

The features composing the specifications of cloud.iO v2.0 have been successfully implemented. Access control has been enhanced with permission based on topics and the introduction of user groups. Features like Remote Jobs Execution and Logging has been implemented in the cloud.iO core and Java library. The certificate management has been implemented directly in the cloud.iO core, removing the need for manual certificate management, as seen in cloud.iO v0.1. The major part of the cloud.iO enhancement part was the RESTful API. It has been implemented successfully, adding tools for User, User Group, access control and endpoint management, certificate creation, history, and log access, and remote job execution. Finally, cloud.iO has been made compliant to the W3C candidate recommendation aiming to the standardization of the IoT called Web of Things.

To go further with cloud.iO v0.2, we can first evoke enhancement on authentication and security. Indeed, to simplify authentication, JSON Web Token could be used. They provide a simple authentication and also secure data transfer. Finally, the major needed improvement of cloud.iO is the implementation of different programming languages for the Endpoint library. The current version includes only Python and Java. It is planned to develop a C/C++ version and implement language wrappers to adapt the library.



