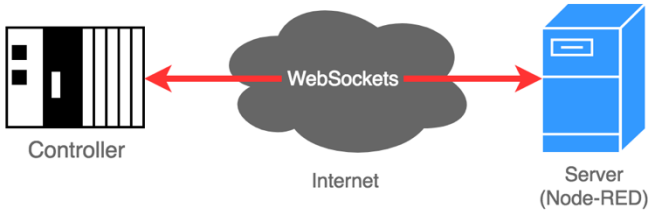
Generic server infrastructure in an industrial environment

ACRONYM	WAGO2
PROPOSED BY	WAGO SA
STUDENT-S	Marc Bürgy
PROFESSOR-S	Daniel Gachet, Rudolf Scheurer
EXPERT-S	Jérôme Page
No	B18T06
TYPE	Bachelor Thesis
CONTACT	marc.buergy@edu.hefr.ch

Context

The company WAGO SA (www.wago.ch) builds powerful programmable controllers based on Linux. For new industrial applications WAGO wants to use web technologies to forward critical messages (alarms, acquisitions, etc.) to centralized servers. The messages should easily cross firewalls in industrial environments.

In its current state, the connection between controller and server is implemented using the WebSockets transfer protocol. This protocol allows bidirectional communication over the standard ports of http and https.



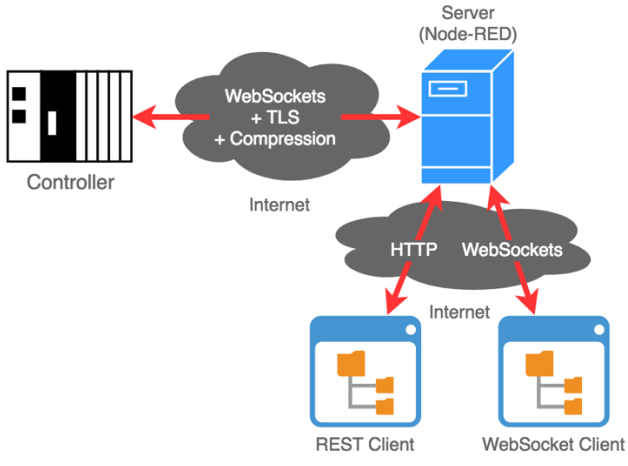
Goals

Encryption of the connection and authentication of all participants with TLS.

Compressing the payload of messages exchanged between the controller and the server.

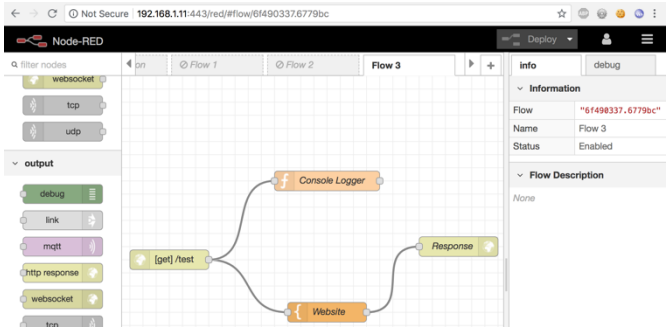
Implementation of HTTP and WebSockets interfaces on the server for external clients.

Measurements and comparison of the delay and payload size with the new implementation.



Node-RED

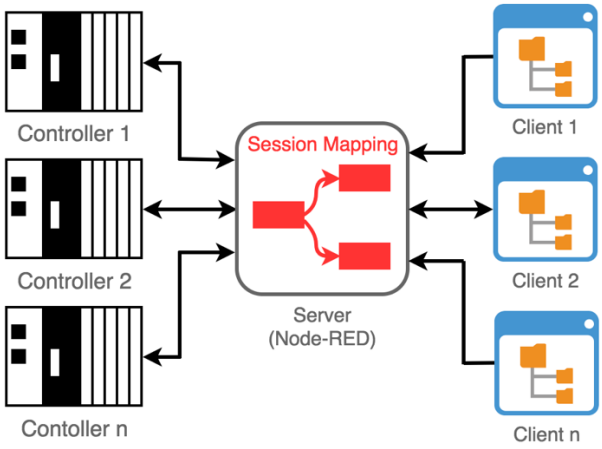
Node-RED is a visual Framework based on node.js. As a server application, Node-RED can be programmed in the browser:



By default, the Framework offers a variety of different input, output and function modules called *nodes*. These nodes are linked together to form *flows*, isolated applications which allow the server to receive, process and send data. This allows for surprisingly complex applications with just a few nodes and is partially suited for users with little to no experience in programming.

Results

Node-RED has been used to create a Session Mapping for this infrastructure:



The Session Mapping allows to manage the exchange of messages between multiple controllers and clients.

The communication between controllers, clients an server is encryption with TLS and the participants are authenticated by certificates.

The compression of WebSockets messages allowed a reduction of payload size by up to **30%**. This is especially useful when the controller is deployed in a mobile network.

Clients ca connect to the server through HTTP or WebSockets to receive updates from selected controllers as well as send commands to controllers.

The measurements have shown that encryption, compression and Session Mapping have little impact on the performance of the infrastructure.

Perspective

During the project, several ideas for expansion of the infrastructure were proposed, such as:

Using a database to store user credentials or the controller states on the server.

An infrastructure that allows automatic certificate creation and distribution to clients/controllers.

Advanced features of the Session Mapping, for instance a user access control function.