VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Aapo Koskela

SETTING UP LOGSET REMOTE SERVICE

Introducing Logset REDI and Logset Manager

Technology and Communication
2020

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Aapo Koskela
Title	Setting up Logset Remote Service
Year	2020
Language	English
Pages	30
Name of Supervisor	Gao, Chao

Logset had a need to develop remote support and control service, which could be used by service personnel and selected resellers. There was also a need for Fleet Management for handling big data, which could be developed with remote service. Logset started developing both services and as a result, they had to be documented. The purpose of this thesis was to strengthen Logset's internal knowledge of new upcoming services, and to generate work instructions for setting up, and the usage of Logset REDI Service. Thesis instructions were written based on interviews with the developers of the service and by personal experiences from the testing.

During writing this thesis, REDI Service and Logset Manager were being simultaneously developed. Logset had implemented objective and basic setup which was then followed by first testing in theory and then in practice, and before the thesis was finished, were both REDI Service and Fleet Management announced publicly.

The thesis presents features that are available now, and those that will probably be available in near future. The work instructions will be taken in use as soon as wide-scale installations in the factory and field will begin.

Keywords CAN bus, CANopen, Work instructions, Remote diagnostics, and control

VAASAN AMMATTIKORKEAKOULU
Koulutusohjelman nimi

TIIVISTELMÄ

Tekijä	Aapo Koskela
Opinnäytetyön nimi	Setting up Logset Remote Service
Vuosi	2020
Kieli	englanti
Sivumäärä	30
Ohjaaja	Gao, Chao

Logsetilla oli tarve kehittää etätuki- ja ohjauspalvelua, jota huolto ja valitut jälleenmyyjät voisivat käyttää. Tämän lisäksi Logset halusi myös kerätä dataa kaluston seuranta -palvelua varten. Tämä pystyttiin toteuttamaan samaan aikaan etähallinnan kanssa. Logset alkoi kehittää molempia palveluita, jotka täten pitää myös dokumentoida. Opinnäytetyön tarkoituksena oli vahvistaa Logsetin omaa sisäistä tietotaitoa tulevasta ominaisuudesta, sekä tehdä työohjeet Logset REDI palvelun käyttöönotosta ja käytöstä. Nämä ohjeet on kirjoitettu ohjelmistokehittäjien haastattelujen, sekä testauksessa tulleiden henkilökohtaisten havaintojen perusteella.

Opinnäytetyön tarkoituksena oli vahvistaa Logsetin sisäistä osaamista uusien palveluiden suhteen, ja kehittää työohjeet Logset REDI-palvelun käyttöönottoa ja käyttöä varten. Opinnäytetyön kirjoittamisen aikana REDI -palvelua kehitettiin samaan aikaan. Logset kehitti suunnitelman ja järjsetelmän perusasetukset, jota tutkittiin ensin teorissa ja sitten käytännön kokeilla. Lopulta, REDI- ja Logset Manager -palvelut julkaistiin ennen kuin opinnäytetyö saatiin päätökseen.

Opinnäytetyö esittelee REDIn ja Managerin toimintaperiaatteet ja ominaisuudet, jotka ovat tällä hetkellä käytettävissä, kuin myös ominaisuudet, jotka tulevat olettavasti käytettäväksi tulevaisuudessa. Työohje otetaan käyttöön heti, kun laajamittaiset asennukset tehtaalla ja kentällä alkavat.

Avainsanat CAN vaylä, CANopen, Työohje, etähallinta ja -ohjaus

CONTENTS

LIST OF FIGURES AND TABLES

TERMS AND ABBREVIATIONS

ARM	Advanced RISC Machine, processor architecture
CAN	Controller Area Network
CANopen	High-level communication protocol, using CAN implementation for lower level protocol
CAN_H	CAN-high cable in CAN bus
CAN_L	CAN-low cable in CAN bus
CLI	Command-line Interface
Client	Computer with hardware and/or software components that requests functionality form server
COB	Communication Object, part of CANopen protocol for time-critical processes, configuration, and network management data
Database	Organised collection of data
Demuxing	Demultiplexing, separating multiplexed data from single medium for individual signals
Device	Controller; valve, relay, switch
DH-GEX	Diffie-Hellman Group Exchange
Distribution	Linux-based operating system
ECU	Electronic Control Unit
Forwarder	Forestry machine for transporting felled logs from the forest to roadside landing
GPS	Global Positioning System
GUI	Graphical User Interface

Harvester Forestry machine for felling trees and cutting to specified length

Hash Output of hash function that generates fixed size value from data

HEX Hexadecimal, numerical representation of value commonly used in computing and mathematics

HTTP Hypertext Transfer Protocol, application layer protocol used for transferring hypertext

I/O Input/Output

ISOC Internet Society, non-profit organization providing Internet related standards, policies, etc.

JSON JavaScript Object Notation, file format

Kernel Program in OS that communicates between software and hardware

Linux Unix-like free operating system family based on Linux kernel

Logset Manager Fleet management service provided for Logset Harvesters. Referred as Fleet, Logset Manager, and Manager

Muxing Multiplexing, combining multiple signals for transmission over single medium

NMEA National Marine Electronics Association, GPS data

NMT Network Management, part of CANopen protocol for controlling and monitoring device states

OS Operating System

OSI model	Open Systems Interconnection model, standardised model of communication function layers
PDO	Process Data Object, CANopen protocol for broadcasting high-priority control and status information
REDI	Logset Remote Diagnostics, service. Referred as REDI and Redi
REST-API	Representational State Transfer – Application Programming Interface
RFC	Request for Comments, publication from ISOC for universal discussion of new innovations, methods research, and behaviours
RS232	Data standard
RTC	Real-time Clock
SAE J1939	Society of Automotive Engineers standard SAE J1939
SCP	Secure Copy Protocol, protocol used for securely transferring files between local and remote hosts
SDO	Service Data Object, CANopen protocol that enables access to all entries of a CANopen object library
Server	Computer with hardware and/or software components that provides other computers/software with external functionality
SQL	Structured Query Language
SSH	Secure Shell, cryptographic network protocol for use of secured remote connection over insecure network
TCP/IP	Transmission Control Protocol/Internet Protocol, used as end-to-end data communication protocol suite
Teltonika RMS	Remote Management System service for Teltonika routers

Third-party software Software that is not developed by Logset or its partners

TOC 2 Total Operation and Control 2, control system. Referred as both TOC 2 and TOC

1 INTRODUCTION

1.1 Background

Finland is a leading forest machinery producer and competition for the market is tough, and therefore it is necessary to keep developing innovations. Network communications have been developing rapidly over the years and this has made it possible for new kind of remote services to appear. Even though, remote service and data gathering over network are not new in the market, it has not yet been developed by Logset Oy (Logset) for their own machinery. For some countries, fleet management is required to have in machinery, which rises the need for release of this development.

1.2 Objectives

The major objective of this Bachelor's thesis is to deepen and generalise understanding of REDI (Logset Remote Diagnostics) for Logset's own use and for my own personal understanding for the company. While REDI is still under development, it is necessary to implement and set parameters for its stable usage. The thesis contains a structured look in the REDI service with expanded view to most relevant parts. The general objective is to make a full-scale description of usage and functions of REDI for Logset and consider how this service can be used in the future. Documentation done for the thesis work will be available for Logset internal use only. It may be requested from Logset directly and copy of it is given if seen as desirable.

1.3 Logset Oy

Logset oy was founded in 1992 and is now located in Koivulahti, Mustasaari, near the city of Vaasa. In 2018 Logset had 40.1 MEUR in revenue and made profit for 1.3 MEUR, while employing 75 workers in Finland. Logset also established a subsidiary Logset Inc in 2019 in Ottawa, Canada, for a better grasp of market share in America. The company manufactures range of forwarders, harvesters, and harvester heads. Logset invests in R&D and was the first to launch a hybrid harvester in 2016. Machines use second generation TOC 2 (Total Operation and Control 2), and TOC-MD 2 (Total Operation and Control Measuring Device 2) for head units, which have Linux based control system. This control system is based on Exertus Oy produced CAN (Controller Area Network) controllers or rather edge modules, which are used in all machines. The module software is custom made for Logset machines. With these modules and co-operation with Exertus, this REDI service is possible to achieve.

1.4 Remote Service in Forest Machinery

There are two main reasons for development of remote services for forest machines. The first is the customer focused fleet management possibility. The owner of the forest or the machine can use online fleet management to follow the phase and productivity of that machine. With this kind of management, the user can calculate much more precise results. The simplified remote connection structure can be seen in Figure 1 below.

The second, also important reason for remote services is troubleshooting and repairing service part. And under COVID-19 pandemic the need for remote support has increased exponentially. As forest machines are sold and used all over the world, travelling becomes inconvenience, as there is a need to repair them from time to time. When most of troubleshooting and section of these repairs can be done remotely, the need for travel across the world decreases dramatically, which is a winning situation for everyone, customer, vendor, and the climate. Of course, there are also repairs that need to be done locally, but when all necessary information is gathered beforehand, the repair itself can be done without problems.

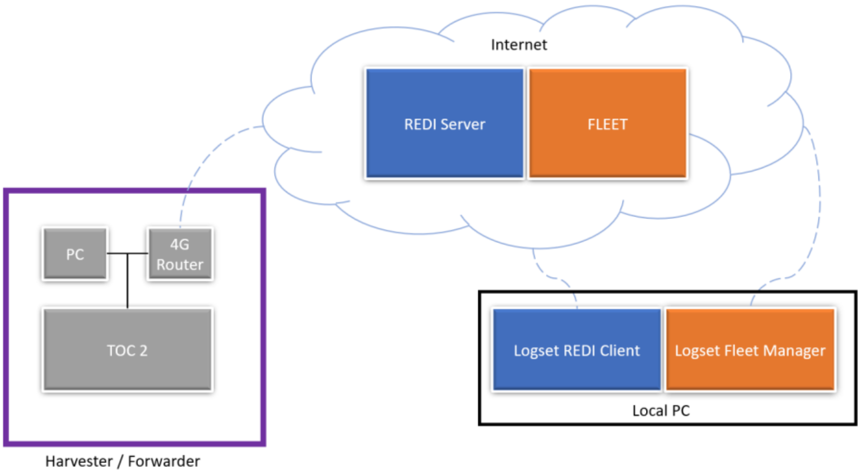

Figure 1. Communication structure

2 COMMUNICATION STRUCTURE

The whole system communication structure can be divided roughly to four different components: local data gathering, edge computing, data transfer, and server communication. Data gathering is done by using sensors connected to I/O controller that will use a CANopen bus to forward information to a Linux based edge computer for procession. Selected processed and raw data is sent to a cloud server using best available networking method (3G/4G) and is then used for personal client applications and further usage on other servers.

2.1 CAN Buses in Logset Forest Machines

Modern machines and vehicles have numerous sensors and devices that need to communicate with each other fast and reliably. CAN communication has been developed just for this purpose and is nowadays used in various fields, including Logset's forest machines. A CAN bus refers to the two first layers of the OSI model (Open Systems Interconnection model). The first layer is a Physical layer and the second is the Data Link layer. These are so called lower layers and in modern CAN communications, higher layers will be used by a higher layer protocol, such as CANopen and J1939, which are used in Logset machines. The CAN Data Link layer protocol was originally standardized in ISO 11898-1 and in 2011 CAN FD (Flexible data-rate) was developed and standardized in same ISO 11898-1.

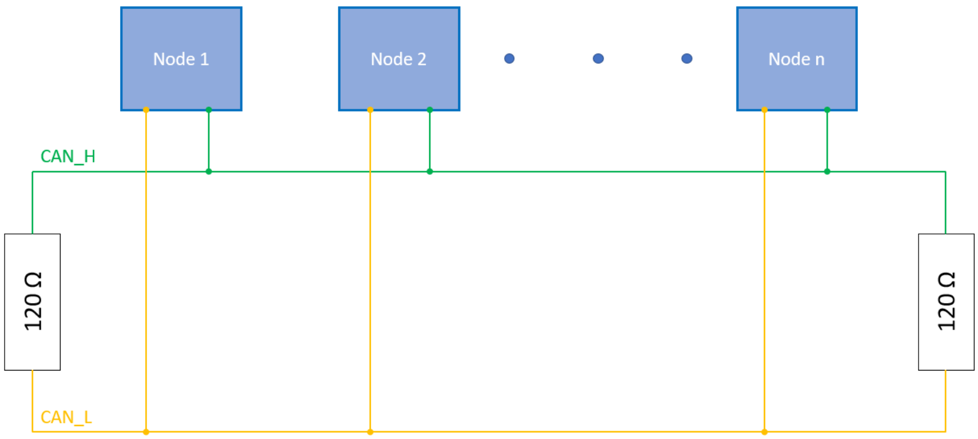

Figure 2. CAN bus connection

The CAN bus is robust network technology that can be used with different topologies but is most used as line-topology with twisted-pair of cables. The most common CAN transmission speed (bit-rate) is 1 Mbit/s (Logset uses 250 Kbit/s) but after CAN FD was introduced, even bit-rates of 2 Mbit/s and 5 Mbit/s are achievable with respectively suitable parameters. The CAN bus is based on two lines, CAN High (CAN_H) and CAN Low (CAN_L) which both are connected to specific nodes and terminated with 120 Ω resistors at each end, as seen in Figure 2. These nodes in Logset harvester are the CAN Controllers (modules) seen in Figure 3 and which will be discussed later in Chapter 2.2. The basic principle of physical CAN control is the Voltage difference of CAN_H and CAN_L, which will determine if the data bit will be read as 0 or 1. CAN signal is based on AND logic, so when CAN_H is 3,5 V and CAN_L is 1.5 V, they are driven toa dominant state and thus produces the data bit 0. On the other hand, when both CAN_H and CAN_L are 2.5V they are driven to a recessive state and thus produces data bit 1 (Figure 3). Basically, a voltage difference under 0.5 V is considered as a recessive state and difference of 2 V is a dominant state. (CiA, 2020)

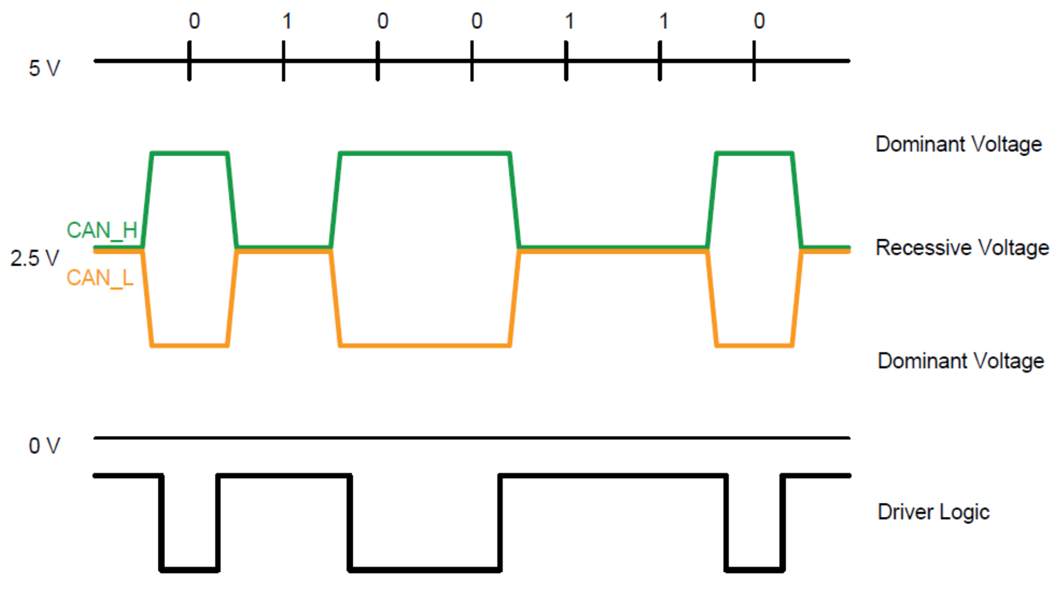

Figure 3. CAN signal

CANopen is a CAN-based communication system that works as a higher-layer protocol, whereas the CAN bus is the lower-layer. CANopen has been developed as flexible and “plug- and- play” type of system in mind. This means that system developers can rather easily implement new CANopen devices to the network, which will work with the existing software; this will then of course lower the production costs. The CANopen protocol consists of SDO (Service Data Object), PDO (Process Data Object), NMT (Network Management), Special function, and Error control protocols, which all are implemented with different COBs (Communication Object). , The CANopen communication is based on messages of different priority, so the channel will prioritise transferring the highest priority messages and will only send lower priority messages if the channel has enough band.

SDOs enable access to the CANopen object dictionary, which will make peer-to-peer client-server communication between two devices possible to establish. In initiation, client device will indicate information accessible on s the object directory of the server, type of SDO, and if the information will be written on or read from.

PDOs are used to broadcast high-priority control and status information. System devices are designed to have a set amount of receive and transmit PDOs based on the need. PDO needs transmission/reception parameters corresponding the usage, communication parameters, and mapping parameters. Four different ways are available for triggering PDO transmission. Event-/Timer-driven triggers when a set timer limit or set value has been exceeded. Remotely-requested needs PDO Consumer to request PDO. Cyclic Synchronous transmission triggers when synchronisation is requested (PDO might not be needed/used but is sent regards that). Acyclic Synchronous transmission is a combination of event and synchronous message, where PDO will be sent on the next synchronous message if the event has been triggered.

NMT tells the state of the machine, which can be Initialisation state, Pre-operational state, Operational state, or Stopped state. The device always begins from the Initialisation state when powered on, and then transits to the Pre-operational state. This automated process will also send a boot-up message to let other devices in network know it is ready. In the Pre-operational state, the device cannot communicate using PDO, but either use the SDO communication or transmit SYNC-Time Stamp, or Heartbeat messages. A Heartbeat message will be periodically sent, and thus it is the easiest way to know if the program has been installed in the device. The Stopped state will only react to NMT messages but has also support for the Error control protocol.

The functions of the Special function and Error control protocols are included in other protocols. The Special functions consist of Synchronisation, Emergency, and Time-stamp protocols. The Error control protocol consist of Heartbeat, Boot-up, and Node-/Life guarding protocols. They are mostly used by SDO, PDO, or NMT protocols. (CiA, 2020)

In addition to CANopen used in Logset machines, there is another higher-layer CAN protocol used, called SAE J1939. Most heavy-duty machines use J1939 communication standard between the engine ECU (Electronic Control Unit) and other control units. J1939 was introduced in 1994 and was formally included part of CAN standard in 2000. J1939 has a slower bit-rate than CANopen, rates being of 250 Kbit/s or 500Kbit/s regards parameters. Similarly to CANopen, J1939 messages can be broadcasted or requested, but the difference is that J1939 uses PGN (Parameter Group Number) as a message identifier, and SPN (Suspect Parameter Number) as a signal identifier. The only place where Logset uses J1939 is the communication between Diesel Engine ECU, Hybrid System ECU (if installed), and Control modules. The internal communication structure can be seen in Figure 4 below. (CSS Electronics, 2020)

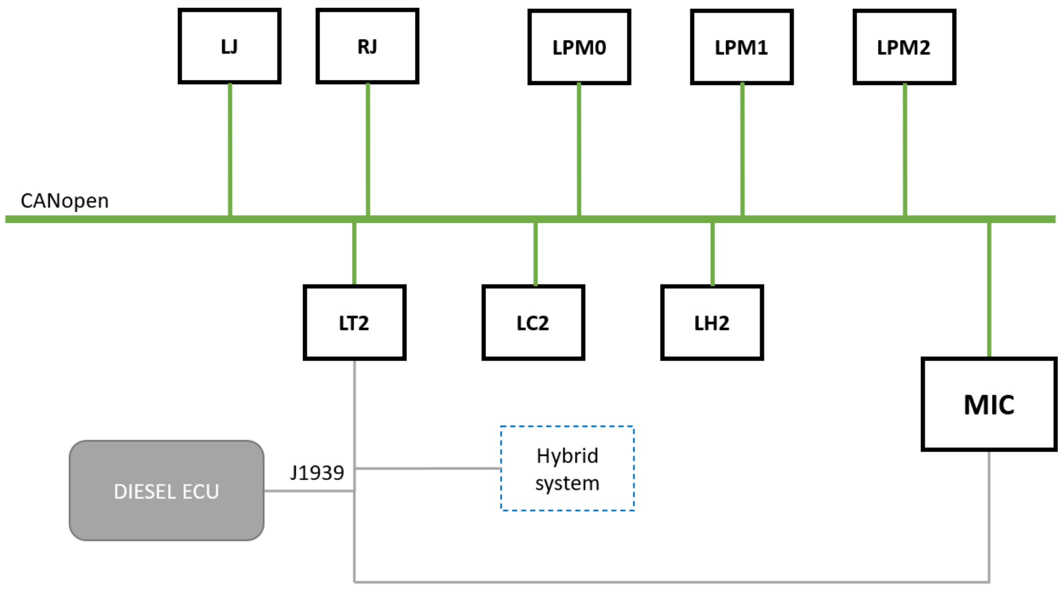

Figure 4. Logset harvester CAN bus layout

There is also sometimes a need to read the data transmission of the CAN bus, and that is where Canto2 is used. Canto2 is Exertus made PC software to transform messages and requests transferred in the CAN bus to a human readable format. Canto2 is suitable for both CANopen and SAE J1939 CAN protocols used in Logset machines. A PC running Canto2 needs to have a built- in CAN controller, or then use a USB-CAN converter to access the CAN bus. When connected to bus, the user can monitor transferred values, read values in modules, and write values in possible locations in module. After REDI was introduced in Logset machines, it has also been possible to connect to the CAN bus remotely and therefore use Canto2 remotely. (Exertus, 2015)

2.2 Software in Modules

Three different types of machine CAN controllers (modules) are used in a basic Logset harvester: MIC2000S, HCM2010S, and HCM2110S. All the modules have different hardware and software configurations (parameters), the differences of which can be seen in specification Table 1. HCM2010S and HCM2110S are for sensors and devices as HCM2010S connects low power sensors/devices to the CAN bus when HCM2110S controls devices needing more power such as lights, and electric motors, in similar fashion to smart relays. The modules have a runtime system written in C language that controls I/O behaviour. The runtime system converts gathered data to the CANopen protocol and transmits it using the CAN bus. The runtime also has software to convert incoming messages to commands for controlling devices. In harvesters there are three HCM2010S modules; harvester head (LH2), crane (LC2), and transmission (LT2) controller. Forwarders do not have LH2 as they are not equipped with harvesting head. Three HCM2110S (LPM0, LPM1, LPM2) are also used in the machines, except forwarder does not have LPM2. All these modules use ARM (Advanced RISC Machine) architecture CPU (Central Processing Unit), which is common architecture in embedded systems. For all CAN connections to be made, a CH8SH CAN Hub is necessary. The CAN hub has 8 CAN ports in a star topology bus, where MIC2000S (MIC), Joysticks (Left [LJ] & Right [RJ]), among others are connected. The module connection layout can be seen in Figure 4.

MIC2000S is used for processing gathered data and controlling devices, using TOC 2 (Total Operation and Control 2). TOC 2 is a control system running on private Linux distribution. There are three major software components where Linux is used: TOC, REDI, and TOC Office. TOC 2 runs Totalview, GUI (Graphical User Interface), which is used for displaying and adjusting variables for device control. Totalview is the only way the basic user can control the whole system, service can use Linux CLI (Command Line Interface) to control the system more widely and even parts of system that Totalview cannot. Linux is used to generate a communication key pair between the machine and the server. The machine – server key is generated using cryptographic authentication. The key is then sent to the server where the machine and key pair is made for further REDI usage. TOC Office is report generating software that communicates with a local PC (if installed). TOC Office will generate standardised reports that can be printed out or sent forward. More of TOC Office will be discussed in Chapter 3.1.3.

Table 1. CAN controller specification (Exertus, 2020a, 2020b, 2018)

Module	CPU	RAM	Flash Memory	Other
MIC2000S (MIC)	ARM Cortex A9 Dual Core, 800 MHz main CPU ARM Cortex M4, 168 MHz co-processor	2 GB, shared	2 GB, flash 1 MB, secondary	2 x Graphics controllers 3 x CAN Interface 2.0 2 x RS232 Interface RTC 40 configurable I/O-lines
HCM2010S (LT2, LC2, LH2 [Harvesters])	ARM Cortex M4, 168 MHZ CPU	256 kB SRAM 4 kB RAM, battery secured	2 MB	RTC Z-axis gyro 60 configurable I/O-lines Max. load 40 A
HCM2110S (LPM0, LPM1, LPM2)	ARM Cortex M4, 168 MHZ CPU	192 kB RAM	1 MB	RTC 52 configurable I/O-lines Max. load 120 A

2.3 Communication Channel from Machine to Server

Data generated in the machines needs to be transferred to the REDI server that will then be used in numerous ways. The usage of this data will be discussed later. In addition to transferring data from the machine, data is also transferred from the server to the machine. Data needs to be transferred using a channel that is secure (usable in insecure networks), light (avoids swelling from unnecessary header data), stable, and usable even with slow connection speeds. The data transfer connection protocol is integrated in Exertus's REDI, which is called ExMeBus. This ExMeBus channel is made by Exertus's ExMeBusServer, which makes the connection between participants. As ExMeBus channel is a private protocol, its structure and work principle cannot be revealed here in detail but according to Arttu Pulli (2020): "ExMeBus traffic is just basic TCP/IP packet traffic". Pulli also explained that data transfers are divided in two separate channels, the CAN traffic channel and the signal data channel, which both use the ExMeBus protocol. The application layer of OSI model has many different TCP/IP (Transmission Control Protocol/Internet Protocol) packet traffic protocols, where SSH (Secure Shell) is one example that will be discussed more in detail on behalf of ExMeBus. (Pulli, 2020)

The SSH-2 (presented as SSH) Protocol Architecture was introduced in ISOC (Internet Society) RFC 4251 (Request for Comments) publication in 2006 by Tatu Ylönen and Chris Lonvick. SSH Connection Protocol is part of this said Architecture and is designed to provide login sessions, remote execution of commands, forwarded TCP/IP connections, and forwarded X11 connections.

SSH is based on Client and Server communication, where either end can send a specified request. An example of a request is a request to start TCP/IP forwarding for a specific port, if the request is accepted, it will be started in an independent channel between Client and Server. If more that one connection channel between Client and Server is established, all channels will be Muxed (Multiplexed) into a single connection. These channels have an identifying number which can be different on each end. Only requests to the open channel contain the sender's channel number, whereas any other channel related message contains the recipient's cannel number. The opened channel can transfer data under specified size limits agreed by both parties. The limits can be requested to be adjusted by either end, and if the sent data does not fit in the allowed limit window, it can be discarded. When no more data is wished to be transferred to the channel, it should be terminated by both parties. Either party can start the channel termination process by sending a closing message to other party, which will reply with the same closing message and so terminate the channel. After the channel termination, the channel opens its channel number to be used for other instances. (Ylönen, 2006a, 2006b)

REDI Service is a flexible solution and has been built to be extendable. It supports bridging to other protocols e.g. SSH. As long as the Exertus ExMeBus protocol stays the same regarding security, flexibility and reliability, Logset has no need to change.

2.4 Signal Data between Servers (REST-API)

The Logset Remote Service is based on two consisting services or rather servers. The first server is for machine - server communication, and another server for Logset Manager, which is dependable on prior communication. Machines will send data as signals to the server where it will be stored in the SQL-database. A machine can also send files which will be saved unmodified. The same signal data is formatted to the JSON (JavaScript Object Notation) format before forwarding to the fleet. This server is used for storing data and files but is also responsible running web-based fleet management with various functions, such as value modification to readable format and graphical display that will improve productive data usage.

The fleet server communicates with the client (usually a browser), using private REST-API (Representational State Transfer – Application Programming Interface). The server has REST-API that the client will use when requesting data. REST is a software architectural style, which is used display and manipulate data within six defined constraints.

Client-Server: The first constraint is to separate client and server from each other. This is to improve and simplify both the server and the client, as both have only functions concerning themselves. (Fielding, 2000)

Stateless: This constraint handles communication between the client(s) and the server. Model of stateless constraint is to keep all necessary information on the client and not use any stored data from the server. All connections are separate (Figure 5), and the client will keep the session state and force requests. By requiring authenticated access from the client, requests will be scalable, reliable, and secure. (Fielding, 2000)

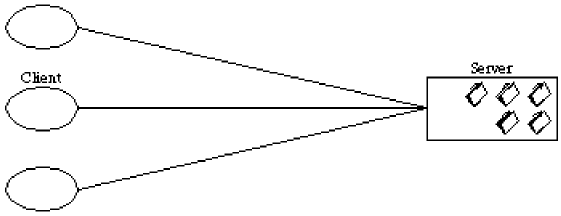

Figure 5. Multiple client connections to Server (Fielding, 2000)

Cache: Efficiency and performance can be improved by caching some or all responses, so that there is no need for multiple repeating requests. Data can be cacheable or non-cacheable, and Logset Manager information is cacheable for aforementioned reasons. (Fielding, 2000)

Uniform Interface: REST has uniform architectural style, where all components follow for simpler communication. This has a positive effect on evolvability on every client but a negative impact on efficiency since connections are not optimized for end-to-end connection. (Fielding, 2000)

Layered System: This constraint allows components to be more flexible in layered system. The component is unable to "see" beyond the immediate level it is interacting with, this adds security but also generates latency and overhead. Figure 6 below shows that between the client and the server there can be different or even multiple components. (Fielding, 2000)

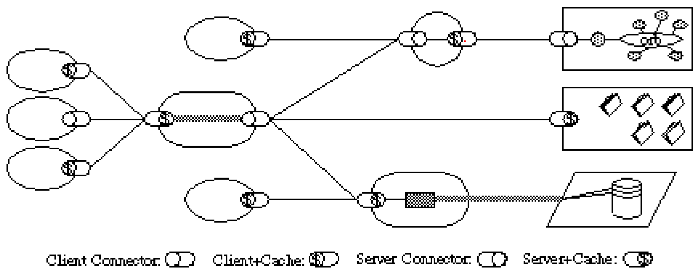

Figure 6. Layered REST system (Fielding, 2000)

Code-on-Demand: The final constraint is extending client functionality by allowing the code to be executed as scripts. The code is first downloaded from the server, but as a firewall or other inhibition may prevent this for security reasons, therefore it is the only constraint that is optional in REST. (Fielding, 2000)

API is a programmatic interface that is used for REST interactions between the client and the server database (Figure 7). The Logset Manager uses a customised Web API interface that receives JSON data from the server and represents it in user readable and visually sensible fashion. If desired, other API solutions can be made in the future.

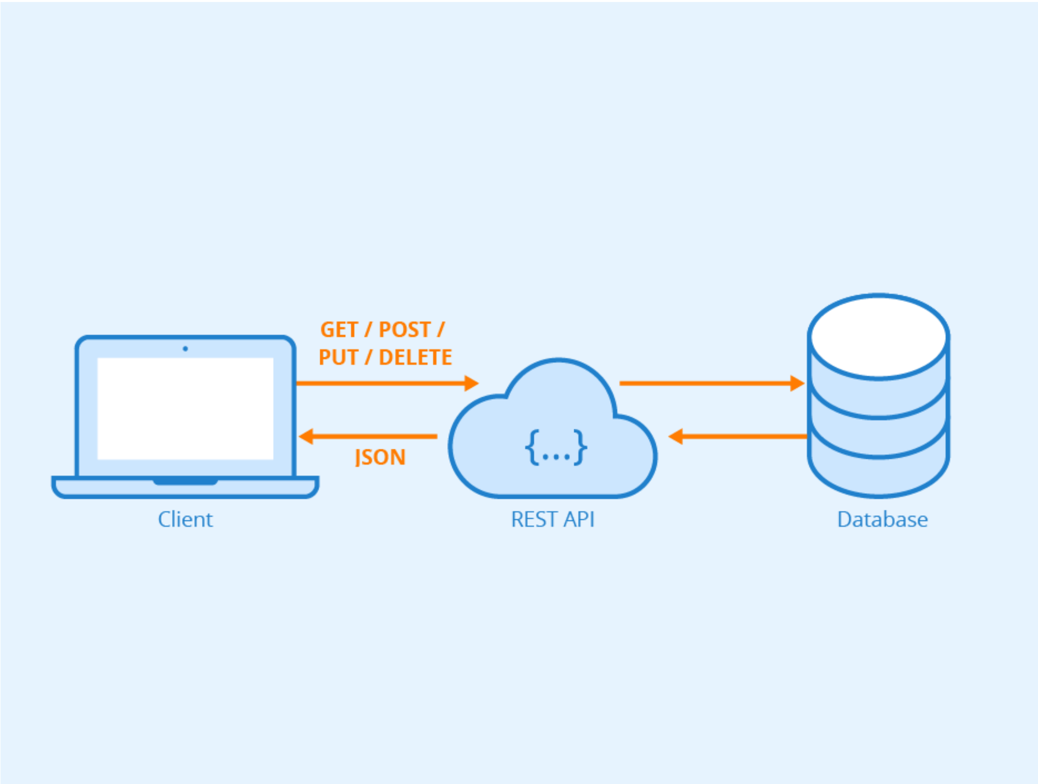

Figure 7. REST-API communication structure (Seobility, 2020)

3 MACHINE – USER COMMUNICATION

The REDI service consists mostly of “non-physical” elements that cannot be accessed by the basic user. What is part of the users’ field is everything inside the actual machine. Harvesters and forwarders consist mostly of the same base equipment and accessories, where, at the time, harvesters have more REDI focused functions.

3.1 REDI Harvester Accessory

A REDI harvester differs from a basic harvester only slightly, as it is sold as an option and deliberately made possible to upgrade a basic harvester model to have the REDI service. Externally both machines have the same CAN modules, Machine-PC, and such but the difference between them is the restrictions in the software of TOC 2 (MIC2000S module).

3.1.1 TOC 2

TOC has a few tangible features that are available for the user i.e. the driver. There are also some features only the owner of the machine can enable/disable, such as allowing signal data to be sent to the Logset Manager and allowing Logset to make connection to the machine TOC. This way important data that the owner could use in the Fleet will not be prevented by the driver. Of course, the owner can give driver the same access level to TOC, so they are able to control same settings.

Driver usable features include File transfer, Chat, and allowing remote control, where both the file transfer and the chat work between the machine and the Manager, and remote control between the machine and the REDI PC-client. File transfer is used to send production reports or files for diagnostic if there is a problem. Chat is unambiguous, it is meant for communication between the driver and the Logset Manager user, without the need of driver personal contact information for example, service contacting driver. Remote control is mostly used by service as they can remotely take control of TOC and check necessary settings or visually see what the problem is. Because harvesters are machines with potential risk of harm, the driver needs to allow remote control before service can connect to the machine, preventing possibilities of the risk. The driver also always needs to have control of the machine when remote control is used so that no accident can happen. In addition, remote control can be only allowed 12 h at a time, and it will automatically disable control if TOC is powered off.

3.1.2 Router

The REDI service needs to have a network connection to function, therefore it is necessary to have cellular communication for data transfer. This is carried out using a WLINK R520-LF 4G router, which allows the usage of fast 4G connection, but also has support for 3G/2G, which are still vastly used in less technologically developed countries. The REDI service is made as lightweight as possible so even 128 kbps connection speed is sufficient for remote connection and control, this will allow remote usage anywhere a cellular connection is available. So far, no satellite connection is used as the price of satellite modem is over thousand euros and the price of data transfer varies from three to five euros per MB. This WLINK router is also used for its GPS (Global Positioning System) capability, as for some cases, GPS coordinates for felled tree needs to be registered. TOC will get the coordinates from NMEA (National Marine Electronics Association) data that WLINK produces.

In future, WLINK may be replaced with a Teltonika RUT955 router, which has the same functionality (except Teltonika only works as GPS Client, where WLINK can work as Client or Server). This Teltonika router also has more customisable functionality for possible future usage, and Teltonika also offers RMS (Remote Management System) that Logset could also utilise. Whether a WLINK R520 or Teltonika RUT955 router is used, there is still no simple way to access either one, as REDI would need functioning internet connection that the router is supposed to give, and therefore only way to access the routers is locally with the Machine-PC.

3.1.3 Machine-PC

Nowadays forestry companies send harvesting instructions using dedicated software, and therefore most Logset harvesters are equipped with an external Machine-PC. The Machine-PC will receive an instruction file from the forestry company and transfer it to TOC. After harvesting is complete, TOC will generate a production file that is then sent to the Machine-PC, which will then forward it to the forest company. This kind of communication between the Machine-PC and TOC is necessary, as it is not possible to install forest companies' software directly to TOC as they are not ARM compatible or they are Windows based software. What makes matters worse is that forestry companies can use different software (even in Finland) for the instruction transfer, which means the software itself needs to work in the Machine-PC, but also that the communication between TOC and Machine-PC must work at all times. The communication between TOC and the PC is made with ExMeBus, which is also used in remote connections from TOC. But nonetheless, this has caused some problems as REDI is not yet capable controlling the Machine-PC, and therefore remote troubleshoot/repair needs to be done using multiple software. Software used the most are REDI for TOC and TeamViewer for Machine-PC. TeamViewer could be replaced by any other Windows compatible desktop sharing software.

3.2 REDI Server

As stated, the REDI Service is based on an Exertus made platform and big part of that Service is the server it is running on. The server itself is controlled and handled by Exertus, but Logset has a possibility to add its own variants for the functionality. At the time, Logset's REDI is version 2.0 compared to version 1.0 is more stable and has more processing power necessary for running multiple connections simultaneously. Processing power has not been an issue so far, as only harvesters have been equipped with REDI, but when forwarders are also equipped, the connections will rise exponentially, as forwarders are produced more than harvesters, and therefore cause more load for the server.

When a new harvester is produced in a factory, or an older machine is desired to be equipped with REDI, it must be added to the REDI Server. The first step is to create a machine specific REDI-key, which will be created in TOC. If the machine later changes an owner, the generated key can be deleted, and a new key generated for the new owner. The key consists of a machine serial number and a securely generated cryptographic data string. In the future REDI-keys will be sent directly from the machine to the REDI server, but as for now, the generated REDI-key needs to be written to external USB drive and manually sent to Exertus, where the key will be manually added to the servers database.

There are multiple machines on the server and therefore each connection needs to be handled with extreme care. There will not be case where data from one machine is forwarded to a different end user, which is one reason why the connections are closed, of course excluding connections made after strict authentication and user management by the server. The most important aspect of the server is the authentication, as life threatening harm can be done if someone with malicious intent gets remote access to the machine. The server will forward data to the RediClient and Logset Manager using muxing (multiplexing), which combines multiple data sources to a single packet and transfers it over a shared medium. The program at the other end will use the data as desired, in each situation.

3.3 REDI Client for PC (Service)

Logset uses the Redi service on every aspect possible and developed it to also for service support. The service can connect remotely to the machine TOC and the whole system with RediClient. RediClient is Exertus made PC software tailored for the Logset use, where all machines with the Redi service can be accessed. RediClient communicates with the REDI server by demuxing (demultiplexing) data from the server and displaying individual machines, in the fashion of Figure 8 below.

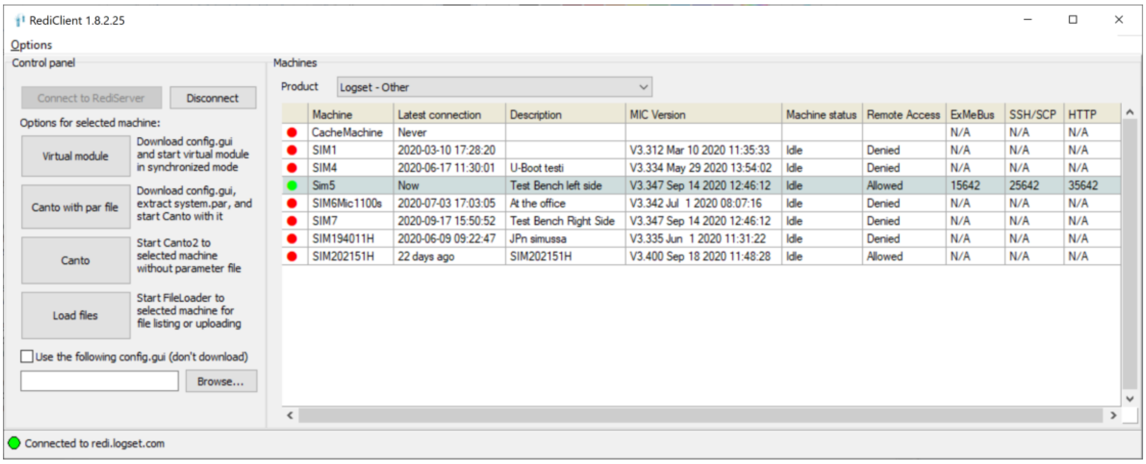

Figure 8. RediClient view of Logset Simulators

The service can use the RediClient in various ways, the most important features are Virtual Module, Canto2, ExMeBusManager, and Secure Port opening. Virtual Module is a virtual copy of machines TOC 2 view. This feature can be used for checking basic settings and instructing driver remotely. As TOC supports a touch display, a Virtual Module user can use the same navigation, with the help of side panel buttons that represent those in the joystick platform. It can also be used the other way around as a driver can show the problem area to the service. There is no voice- or text-chat available, so the service and the driver must have another communication way established for comprehensive support.

Canto2 is the same software as introduced in Chapter 2.1, it is normally used locally with a physical connection to the CAN bus of the machine. As modern laptops have no CAN interface, a CAN-USB adapter is necessary in the local connection but with RediClient a remote CAN bus connection can be done and Canto2 used. Remote Canto2 is not as reliable in slow connection speeds so it is mostly used just checking and modifying the system parameters, whereas with fast connections, all Canto2 features can be used.

ExMeBusManager is the most barebone feature RediClient has. It basically forwards selected machines data from the REDI server to the client in HEX (Hexadecimal) representation. This is used to check validity of the data before it is being altered to more a human friendly fashion, such as Logset Manager or Virtual Module. It is possible to select what data to gather from a given option. This feature is seldom used other than testing purposes but still exists as a backup if nothing else works.

Opening Secure Ports as a feature has the most possibilities to develop as third-party software can be used with it. RediClient opens a port between the local PC and the remote TOC, which allows the local machine to then establish a connection to the TOC's program. As can be seen in Figure 8, RediClient can open a secure port directly for ExMeBus, SSH/SCP (Secure shell / Secure Copy Protocol), and HTTP (Hypertext Transfer Protocol). ExMeBus port is used in Canto2 and Virtual Module connections. SSH/SCP ports can be used with PuTTY and WinSCP as an example. PuTTY is a terminal emulator that opens a CLI connection directly to TOC. Like Canto2, PuTTY can also be used locally. WinSCP is a secure file transfer tool made for Windows PC, which can be used for transferring files between the local PC and the connected TOC. Logset does not use HTTP connection in TOC at the time, but it could be utilized in the future. Other ports can also be securely opened for TOC, so the possibilities this feature is almost limitless.

3.4 Logset Manager

Logset Manager is a fleet management platform that Logset has been developing together with Remion. This fleet service is based on Remion's Regatta solution and tailored for the Logset use case. The Logset Manager utilizes signal data gathered from the harvester that the REDI server tunnels and converts it to a Regatta suitable format. This formatted data is then displayed in a user suitable fashion, for example, GPS data is displayed in a map, with backtracking possibility. Manager users mostly consist of the owners of one or multiple machines that want to monitor and plan the use of the machines. The data in the Manager can also be utilized for future planning and calculations for new offers to be made. As mentioned before, the user can send and receive files and chat messages in the respective tabs.

Figure 9 below shows the layout of a Productivity tab, with daily summary of gathered test data from span of a week. At the time, this is the default harvester layout, where the user can only change the time span to view. It is also possible for Logset to handcraft custom layouts that will be more suitable for the customer, and when Forwarders start to use the Redi service, Productivity view must be upgraded to be more suitable.

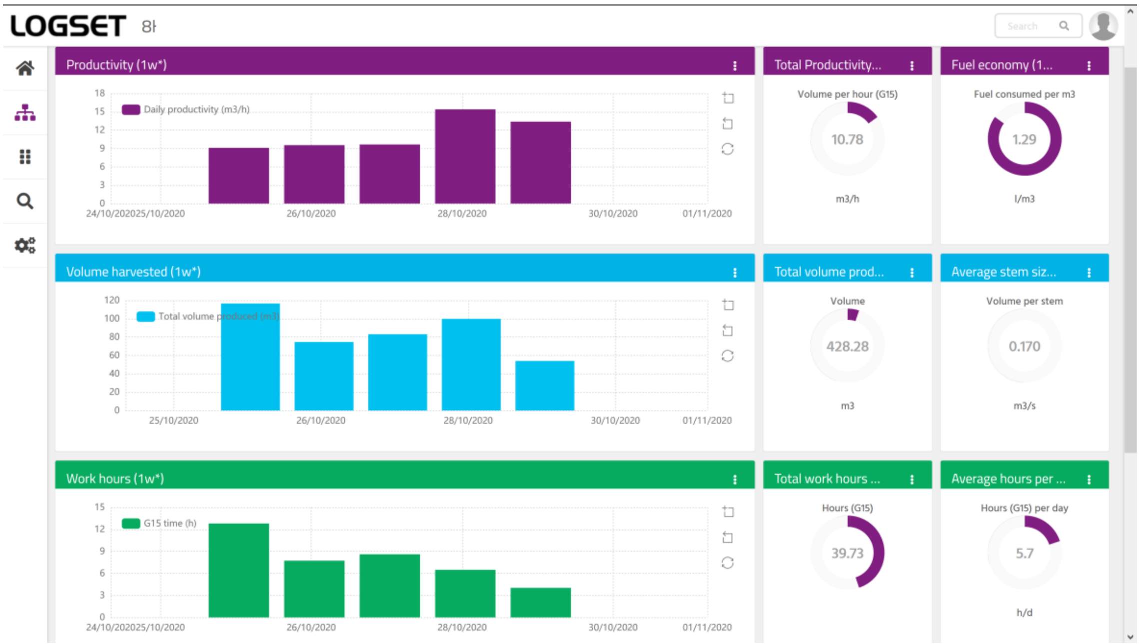

Figure 9. Logset Manager Productivity view

Logset has implemented the Manager to have tools for service to use. The service can use these tools to remotely inspect the condition of the machine, and thereafter prevent unnecessary repairs, or plan the most suitable time for the repair. These tools include more detailed information gathered directly from sensors and devices, that can be used in different ways. A good example of this would be service exporting all temperature data from a two-week period and examining it for any threatening signs. In the future, an automated process could follow the condition of the machine and report service when something alarming is noticed.

4 CONCLUSION

The idea of the thesis was first introduced in the beginning of 2020, and the thesis work that was done was meant to both strengthen Logset’s internal knowledge and generate usable work instructions. Writing easy to understand and simultaneously comprehensive instructions turned out to be more difficult as REDI and Manager were both being developed during the thesis work, and there were a few features that were removed or changed completely and thus had to be removed or re-written respectively. The base of the system structure was fortunately stable as changing from CANopen to another higher-layer CAN protocol would not be done even in a year.

The development was done with the close partner Exertus that also provides Logset with necessary CAN control modules and other software components. What turned out to be surprisingly challenging was to manage all network related connections stable. One problem that was not accounted for was the Machines Router firmware that had a design flaw, but which the manufacturer fixed by developing Logset specified firmware.

The work instructions written with the thesis work has not yet been widely used as REDI and Manager are still in a small -scale, kind of applied testing, phase. After REDI will be part of Logset service and customer day-to-day life, the instructions will become necessity. Before that happens, will there probably be a need to review and rewrite the instructions as REDI and Manager are constantly being developed, and changes to both hardware and software are certain.

REFERENCES

CAN in Automation (CiA). 2020. CAN Protocol. Web page. Accessed 20.08.2020. https://www.can-cia.org/can-knowledge/

CAN in Automation (CiA). 2020. CANopen Protocol. Web page. Accessed 20.08.2020. https://www.can-cia.org/canopen/

CSS Electronics. 2020a. CAN Protocol. Web page. Accessed 20.08.2020. https://www.csselectronics.com/screen/page/canopen-tutorial-simple-intro/language/en

CSS Electronics. 2020b. SAE J1939 communication. Web page. Accessed 22.08.2020. https://www.csselectronics.com/screen/page/simple-intro-j1939-explained

Exertus Oy. 2015. Canto2 Monitoring and commission tool introduction. Online resource. Accessed 18.10.2020. https://exertusoy.sharepoint.com/:b:/s/exefiles/EbAxsb9dS41Hk9yXxJg-Mz8BxDJpR_8bweTYQn36qNy_zQ

Exertus Oy. 2014. CH8SH CAN Hub Module. Technical Data Sheet. Online resource. Accessed 06.11.2020. https://exertusoy.sharepoint.com/:b:/s/exefiles/EWGVXi4kzixPlgRdOlZIt3MB1hXU0WlVflCsT8A72qPt3Q

Exertus Oy. 2020a. HCM2010S Technical Data Sheet. Online resource. Accessed 06.11.2020. https://exertusoy.sharepoint.com/:b:/s/exefiles/EdHpX-i_WpZHp3Kho9R12WYByyobGYMt1duz-hagGvVsMg

Exertus Oy. 2020b. HCM2110S Technical Data Sheet. Online resource. Accessed 06.11.2020. https://exertusoy.sharepoint.com/:b:/s/exefiles/EWTt4QN2eGVHjzaAuagLG9sBHFAuKrNJqYVqRESfDwMNRw

Exertus Oy. 2018. MIC200S Technical Data Sheet. Online resource. Accessed 06.11.2020. https://assets.website-files.com/5c545c15041cbc3e59a8ff56/5c5986f702a2979a9e4e2c60_TechDoc_MIC2000S.pdf

Exertus Oy. 2016. ReDiService Overview. Online resource. Accessed 02.11.2020. https://exertusoy.sharepoint.com/:b:/s/exefiles/EUlszpV_NWZMsPK9YqpiLnMBfZPsrb8qjyejOVa_VD2s5Q

Fielding, Roy Thomas. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine. Web page. Accessed 28.06.2020. https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Kytövaara J. 2020. Logset Oy. Interview 23. April 2020.

Pulli, A. 2020. Exertus Oy. Interview 11. May 2020

Seobility. 2020. REST API. Web page. Accessed 29.08.2020. https://www.seobility.net/en/wiki/REST_API

Ylönen, T. 2006a. The Secure Shell (SSH) Protocol Architecture. Online resource. Accessed 02.05.2020. https://tools.ietf.org/pdf/rfc4251.pdf

Ylönen, T. 2006b. The Secure Shell (SSH) Connection Protocol. Online resource. Accessed 02.05.2020. https://tools.ietf.org/pdf/rfc4254.pdf