SDVPv2

API Overview

Mapping between

IEC 61850-7-2 ACSI

and

XMPP stanzas

D R A F T

Revision: 2013-05-22

EURISCO
RESEARCH & DEVELOPMENT

Content

1 Introduction 4
2 System overview 4
3 API example walkthrough 5
4 Library API 6
4.1 Java library API 6
4.1.1 Session 6
4.1.1.1 Associate 6
4.1.1.2 Abort 6
4.1.1.3 Release 6
4.1.2 Directory 6
4.1.2.1 GetServerDirectory 6
4.1.2.2 GetLogicalDeviceDirectory 6
4.1.2.3 GetLogicalNodeDirectory 6
4.1.2.4 GetDataDirectory 6
4.1.3 Data 6
4.1.3.1 GetDataValue 6
4.1.3.2 SetDataValue 7
4.1.3.3 GetAllDataValues 7
4.1.4 Data set 7
4.1.4.1 GetDataSetValue 7
4.1.4.2 SetDataSetValue 7
4.1.4.3 CreateDataSet 7
4.1.4.4 DeleteDataSet 7
4.1.4.5 GetDataSetDirectory 7
4.1.5 Reporting 7
4.1.5.1 Report 7
4.1.5.2 GetBRCBValues 7
4.1.5.3 SetBRCBValues 7
4.1.6 Logging 8
4.1.6.1 QueryLogByTime 8
4.1.6.2 QueryLogAfter 8
4.1.6.3 GetLogStatusValues 8
4.1.6.4 GetLCBValues 8
4.1.6.5 SetLCBValues 8
4.2 C library API 9
4.2.1 Session 9
4.2.1.1 Associate 9
4.2.1.2 Abort 9
4.2.1.3 Release 9
4.2.2 Directory 9
4.2.2.1 GetServerDirectory 9
4.2.2.2 GetLogicalDeviceDirectory 9
4.2.2.3 GetLogicalNodeDirectory 9
4.2.2.4 GetDataDirectory 9

4.2.3 Data 9
4.2.3.1 GetDataValue 9
4.2.3.2 SetDataValue 9
4.2.3.3 GetAllDataValues 10
4.2.4 Data set 10
4.2.4.1 GetDataSetValue 10
4.2.4.2 SetDataSetValue 10
4.2.4.3 CreateDataSet 10
4.2.4.4 DeleteDataSet 10
4.2.4.5 GetDataSetDirectory 10
4.2.5 Reporting 10
4.2.5.1 Report 10
4.2.5.2 GetBRCBValues 10
4.2.5.3 SetBRCBValues 10
4.2.6 Logging 10
4.2.6.1 QueryLogByTime 10
4.2.6.2 QueryLogAfter 11
4.2.6.3 GetLogStatusValues 11
4.2.6.4 GetLCBValues 11
4.2.6.5 SetLCBValues 11
4.3 API versus XMPP data model 12
5 XMPP data model 13
5.1 Underlying basis 13
5.2 Data model 13
5.2.1 XMPP native 13
5.2.2 GetNameList 13
5.2.3 Read 13
5.2.4 Write 13
6 References 13

1 Introduction

This document will give an overview of the two APIs used in “Styr Din Varme Pumpe version 2” (SDVPv2) and the XMPP message format.

The two APIs are for the Java library and the C library and will described separately.

The XMPP messages format part will describe which technologies there are used for XMPP and is showing how the XMPP messages is formatted for use in this project.

2 System overview

The system overview in Figure 1 shows how the communication layers are split in the project. The red parts are the applications which communicate to each other through the SDVPv2 system. The blue parts are third party libraries. The green parts are developed in the project.

Going into details about the system is out of scope for this document.

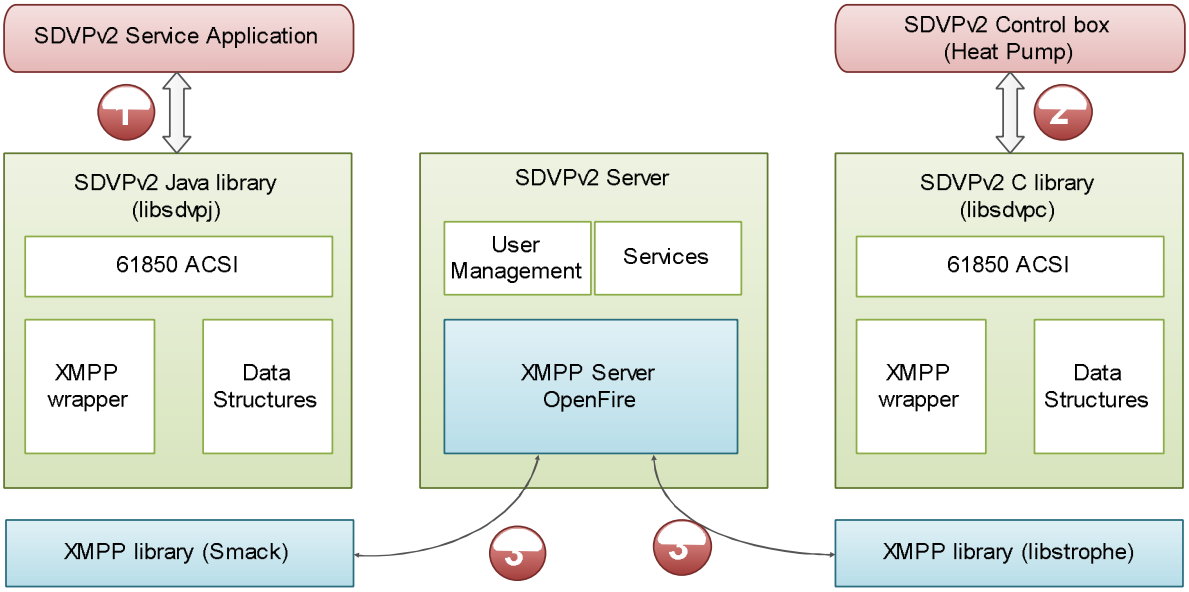

Figure 1: SDVPv2 system overview

As seen in Figure 1 there are three point of interest regarding interfaces.

1. The API for interfacing the Java library
2. The API for interfacing the C library
3. The XMPP message format between the XMPP clients, this is called ‘XMPP data model’.

The APIs for interfacing the libraries is mostly the same but have some minor differences in naming.

3 API example walkthrough

Given below is an example of how the API is looking. The example is showing how a Service Application makes a request for a specific data value on a Control Box and how the reply is send back. The numbers in bold references to Figure 1.

1. **1** – The service application call the Java library method for reading out a data value, this could be the in door temperature.
    - GetDataValue (HP_LD/STMP_1.Tmp)
2. **3** – The SDVPv2 Java library then creates a XMPP IQ stanza (a specific looking XMPP message) which is to be send by the XMPP library.
    - <iq type='get'
      from='service_app@sdvpv2.dk/java-client'
      to='heatpump@sdvpv2.dk/controlbox'
      id='request1'>
      <query xmlns='jabber:iq:rpc'>
      <methodCall>
      <methodName>read</methodName>
      <params>
      <param>
      <value><string> HP_LD/STMP_1.Tmp </string></value>
      </param>
      </params>
      </methodCall>
      </query>
      </iq>
3. **2** – The previous XMPP stanza is received by the libsdvpc library on the Control box and the messages is translated into a request which is send to the Control box application using a callback. The return value from the callback could be:
    - 21.0
4. **3** – The libsdvpc then creates a XMPP stanza to send back.
    - <iq type='result'
      from='heatpump@company-a.com/controlbox'
      to=service_app@company-b.com/java-client'
      id='request1'>
      <query xmlns='jabber:iq:rpc'>
      <methodResponse>
      <params>
      <param>
      <value><double>21.0</ double></value>
      </param>
      </params>
      </methodResponse>
      </query>
      </iq>

5 **1** – The Java library then converts the XMPP reply stanza back to a local data structure and returns it when exiting the GetDataValue method.

4 Library API

4.1 Java library API

Constructor

4.1.1 Session

4.1.1.1 *Associate*

See [IEC 61850-7-2 Ed. 2 – section 8.3.2.2]

4.1.1.2 *Abort*

See [IEC 61850-7-2 Ed. 2 – section 8.3.2.3]

4.1.1.3 *Release*

See [IEC 61850-7-2 Ed. 2 – section 8.3.2.4]

4.1.2 Directory

4.1.2.1 *GetServerDirectory*

See [IEC 61850-7-2 Ed. 2 – section 7.2.2]

4.1.2.2 *GetLogicalDeviceDirectory*

See [IEC 61850-7-2 Ed. 2 – section 9.2.1]

4.1.2.3 *GetLogicalNodeDirectory*

See [IEC 61850-7-2 Ed. 2 – section 10.2.2]

4.1.2.4 *GetDataDirectory*

See [IEC 61850-7-2 Ed. 2 – section 11.4.4]

4.1.3 Data

4.1.3.1 *GetDataValue*

See [IEC 61850-7-2 Ed. 2 – section 11.4.2]

4.1.3.2 *SetDataValue*

See [IEC 61850-7-2 Ed. 2 – section 11.4.3]

4.1.3.3 *GetAllDataValues*

See [IEC 61850-7-2 Ed. 2 – section 10.2.3]

Get all data attribute values of all data objects made visible to the requesting client by the referenced logical node.

4.1.4 Data set

4.1.4.1 *GetDataSetValue*

See [IEC 61850-7-2 Ed. 2 – section 13.3.2]

4.1.4.2 *SetDataSetValue*

See [IEC 61850-7-2 Ed. 2 – section 13.3.3]

4.1.4.3 *CreateDataSet*

See [IEC 61850-7-2 Ed. 2 – section 13.3.4]

4.1.4.4 *DeleteDataSet*

See [IEC 61850-7-2 Ed. 2 – section 13.3.5]

4.1.4.5 *GetDataSetDirectory*

See [IEC 61850-7-2 Ed. 2 – section 13.3.6]

4.1.5 Reporting

4.1.5.1 *Report*

See [IEC 61850-7-2 Ed. 2 – section 17.2.3.2]

4.1.5.2 *GetBRCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.2.3.3]

4.1.5.3 *SetBRCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.2.3.4]

4.1.6 Logging

4.1.6.1 *QueryLogByTime*

See [IEC 61850-7-2 Ed. 2 – section 17.3.5.2]

4.1.6.2 *QueryLogAfter*

See [IEC 61850-7-2 Ed. 2 – section 17.3.5.3]

4.1.6.3 *GetLogStatusValues*

See [IEC 61850-7-2 Ed. 2 – section 17.3.5.4]

4.1.6.4 *GetLCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.3.2.5]

4.1.6.5 *SetLCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.3.2.6]

4.2 C library API

4.2.1 Session

4.2.1.1 *Associate*

See [IEC 61850-7-2 Ed. 2 – section 8.3.2.2]

4.2.1.2 *Abort*

See [IEC 61850-7-2 Ed. 2 – section 8.3.2.3]

4.2.1.3 *Release*

See [IEC 61850-7-2 Ed. 2 – section 8.3.2.4]

4.2.2 Directory

4.2.2.1 *GetServerDirectory*

See [IEC 61850-7-2 Ed. 2 – section 7.2.2]

4.2.2.2 *GetLogicalDeviceDirectory*

See [IEC 61850-7-2 Ed. 2 – section 9.2.1]

4.2.2.3 *GetLogicalNodeDirectory*

See [IEC 61850-7-2 Ed. 2 – section 10.2.2]

4.2.2.4 *GetDataDirectory*

See [IEC 61850-7-2 Ed. 2 – section 11.4.4]

4.2.3 Data

4.2.3.1 *GetDataValue*

See [IEC 61850-7-2 Ed. 2 – section 11.4.2]

4.2.3.2 *SetDataValue*

See [IEC 61850-7-2 Ed. 2 – section 11.4.3]

4.2.3.3 *GetAllDataValues*

See [IEC 61850-7-2 Ed. 2 – section 10.2.3]

Get all data attribute values of all data objects made visible to the requesting client by the referenced logical node.

4.2.4 Data set

4.2.4.1 *GetDataSetValue*

See [IEC 61850-7-2 Ed. 2 – section 13.3.2]

4.2.4.2 *SetDataSetValue*

See [IEC 61850-7-2 Ed. 2 – section 13.3.3]

4.2.4.3 *CreateDataSet*

See [IEC 61850-7-2 Ed. 2 – section 13.3.4]

4.2.4.4 *DeleteDataSet*

See [IEC 61850-7-2 Ed. 2 – section 13.3.5]

4.2.4.5 *GetDataSetDirectory*

See [IEC 61850-7-2 Ed. 2 – section 13.3.6]

4.2.5 Reporting

4.2.5.1 *Report*

See [IEC 61850-7-2 Ed. 2 – section 17.2.3.2]

4.2.5.2 *GetBRCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.2.3.3]

4.2.5.3 *SetBRCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.2.3.4]

4.2.6 Logging

4.2.6.1 *QueryLogByTime*

See [IEC 61850-7-2 Ed. 2 – section 17.3.5.2]

4.2.6.2 *QueryLogAfter*

See [IEC 61850-7-2 Ed. 2 – section 17.3.5.3]

4.2.6.3 *GetLogStatusValues*

See [IEC 61850-7-2 Ed. 2 – section 17.3.5.4]

4.2.6.4 *GetLCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.3.2.5]

4.2.6.5 *SetLCBValues*

See [IEC 61850-7-2 Ed. 2 – section 17.3.2.6]

4.3 API versus XMPP data model

61850-7-2 ACSI Naming	XMPP data model
Session	
Associate	XMPP native
Abort	XMPP native
Release	XMPP native
Directory	
GetServerDirectory	GetNameList
GetLogicalDeviceDirectory	GetNameList
GetLogicalNodeDirectory	GetNameList
GetDataDirectory	GetNameList
Data	
GetDataValue	Read
SetDataValue	Write
GetAllDataValues	
Data set	
GetDataSetValue	Read
SetDataSetValue	Write
CreateDataSet	Write
GetDataSetDirectory	GetNameList
Reporting	
Report	Write
GetBRCBValues	Read
SetBRCBValues	Write
Logging	
QueryLogByTime	Read
QueryLogAfter	Read
GetLogStatusValues	Read
GetLCBValues	Read
SetLCBValues	Write

5 XMPP data model

This data model is for describing how the XMPP stanzas are formatted in order to send commands between XMPP clients.

5.1 Underlying basis

The XMPP data model is based on XML-RPC like specified in "XEP-0009: Jabber-RPC"[1].

Using XML-RPC has the benefits of being more human readable than many others protocols and is easily to debug due to its clear text nature.

The downside of XML-RPC is that no description language exists, like WSDL files for SOAP web services. This makes is hard to maintain code when changes are performed at the model.

5.2 Data model

5.2.1 XMPP native

5.2.2 GetNameList

5.2.3 Read

5.2.4 Write

6 References

[IEC 61850-7-2 Ed. 2] IEC 61850-7-2 Ed. 2: Communication networks and systems for power utility automation – Part 7-2: Basic information and communication structure – Abstract communication service interface (ACSI)

[1] http://xmpp.org/extensions/xep-0009.html