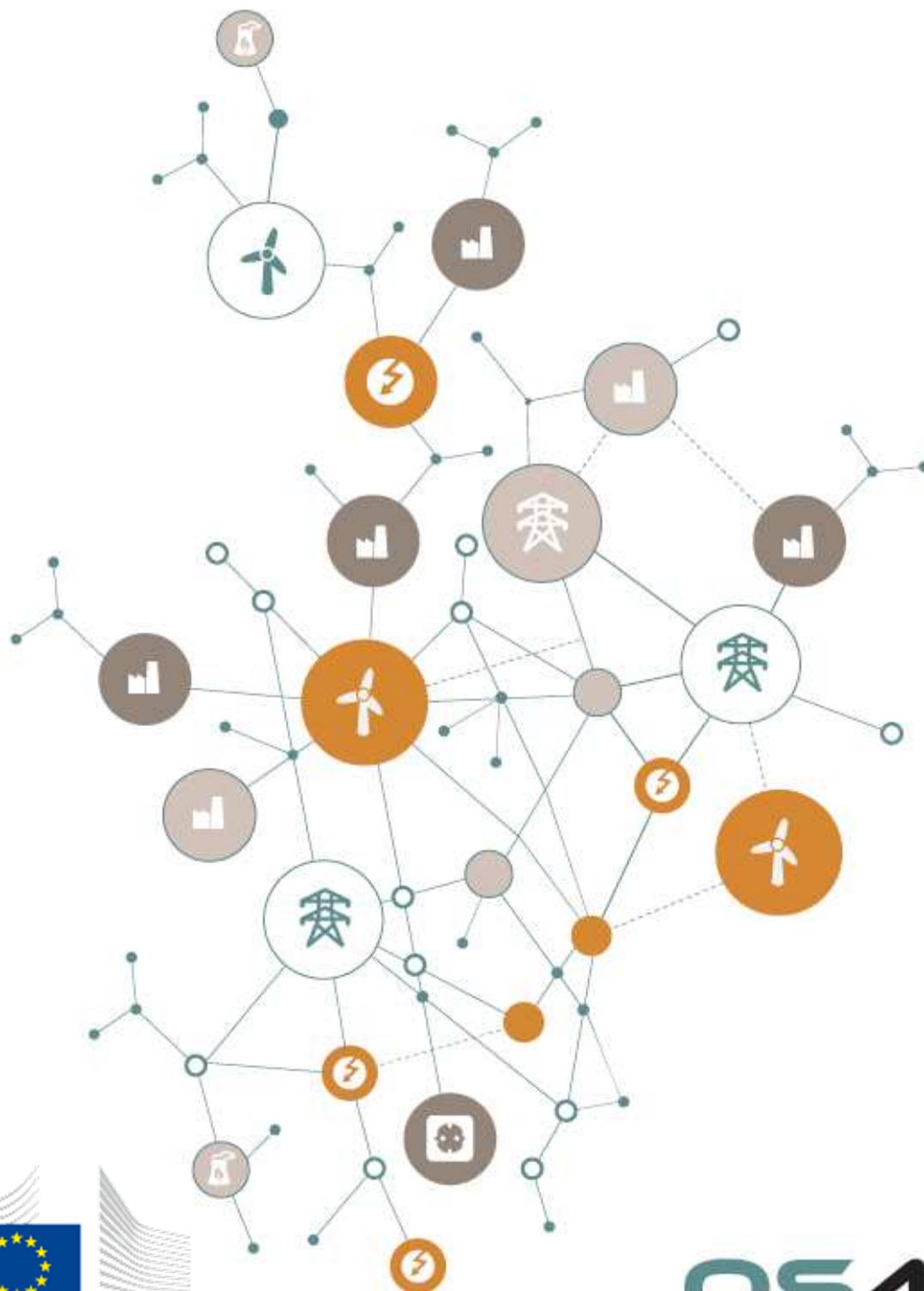


Deliverable 5.4

Application prototypes



D5.4 Application prototypes

Document Information

Programme	FP7-ICT-2013-11
Project acronym	OS4ES
Grant agreement number	619302
Number of the Deliverable	D5.4
WP/Task related	WP5
Type (distribution level)	RE
Start date of project	01/07/2014
Date of delivery	15/06/2016
Status and Version	Final version 1.0
Number of pages	37
Document Responsible	Mikel Fernández – Tecnalia
Author(s)	Idoia Murua –Tecnalia Mikel Fernández – Tecnalia
Reviewers	Michiel van den Berge - Stedin

Revision History

Version	Date	Author/Reviewer	Notes
0.1	27.06.2016	Idoia Murua, Tecnalía	ToC
0.2	28.06.2016	Idoia Murua, Tecnalía	First version
0.3	01.07.2016	Mikel Fernández, Tecnalía	Internal revision
1.0	14.07.2016	Michiel van den Berge, Stedin	Final revision

Executive Summary

This deliverable describes the application prototypes that will be executed by the smart grid actors to validate the OS4ES system in the scope of the WP7 laboratory and field trials.

This document is structured in two main chapters:

1. An introductory part describes the purpose and scope of the document.
2. Chapter 2 explains how to install, configure and use the application prototypes

Table of contents

- Document Information 1
- Revision History..... 2
- Executive Summary..... 3
- Table of contents 4
- List of Tables..... 6
- List of Figures 7
- 1 Introduction 8
 - 1.1 Purpose..... 8
 - 1.2 Lead partner 8
 - 1.3 Scope 8
 - 1.4 Notations, abbreviations and acronyms 10
- 2 UsefOS4ES Installation 11
 - 2.1 Source code repository 11
 - 2.2 Code structure..... 11
 - 2.2.1 Environment variables 12
 - 2.1 Installation prerequisites..... 12
 - 2.1.1 Oracle Java SE Development Kit 8 or OpenJDK JDK 8:..... 12
 - 2.1.2 Apache Maven 12
 - 2.1.3 JBoss Wildfly..... 13
 - 2.1.4 H2 13
 - 2.1.5 Libsodium 13
 - 2.1.6 Root certificate (optional, for demo purposes)..... 14
 - 2.1.7 ISC BIND..... 15
 - 2.1.8 Hat-open-dist stack and JavaPythonBridge 16
 - 2.2 Environment utilization 17
 - 2.2.1 Starting the USEF environment..... 18
 - 2.2.2 Stopping the USEF environment..... 18
 - 2.2.3 Accessing the USEF database..... 18
 - 2.2.4 Sending messages 19
 - 2.2.5 Aggregator GUI..... 20
 - 2.2.6 JBoss Management Console 26
 - 2.2.7 Nodes folder..... 26
 - 2.3 Configuration..... 27

2.3.1	Adjusting the usef-environment.yaml configuration	27
2.3.2	Adjusting iec61850_client.yml configuration	28
2.3.3	Zones, contract templates and DER addressing related data in USEF database	28
2.3.4	Logging	30
2.3.5	Reserved TCP ports	30
2.3.6	Configure resolver entries.....	31
2.3.7	Configuring a proxy server (optional)	31
2.3.8	Resolving participant information	32
Annex	34
Annex A. Configuration of scheduled processes	34
References.....	36

List of Tables

Table 1: Extraday triggers	35
Table 2: Intraday triggers	35

List of Figures

Figure 1: BIND installation window.....	15
Figure 2: H2 Database connection dialog	19
Figure 3: Web page to contract new DER system services (Market-wise contract type) –part 1	21
Figure 4: Web page to contract new DER system services (Market-wise contract type) –part 2	22
Figure 5: Web page showing the SearchResponse results	23
Figure 6: Web page showing detailed info of a specific service – Part1	24
Figure 7: Web page showing detailed info of a specific service – Part2	25
Figure 8: Web page showing the ContractResponse results	25
Figure 9: WildFly mangement console.....	26

1 Introduction

1.1 Purpose

This deliverable presents the application prototypes developed by the OS4ES project consortium in order to validate the OS4ES system, understanding the OS4ES system as a black-box which includes the middleware developed in WP6, the registry developed in WP4 and the DER resources gateways/communication protocol developed in WP3.

These applications have been developed in both tasks T5.3 *Prototyping DER management applications that integrate network operation algorithms and middleware services*, and T5.4 *Prototyping DER management applications that integrate market participation algorithms and middleware services*.

As a concretization of the description on how the scope of both tasks was envisaged in the DoW, it was decided that T5.4 involves the application executed by the aggregator and T5.3 the applications executed by the other smart grid actors.

1.2 Lead partner

Although in the DoW the lead partner was TNO, an agreement within the consortium has been arranged, to shift that responsibility to TECNALIA, which has been the partner leading the specification, design and implementation of the prototypes.

1.3 Scope

The application prototypes have been developed to validate the OS4ES system in the frame of the OS4ES use cases defined in the deliverable D1.1 [1]. They have been developed based on the design defined in deliverable D5.3 [2] The current implementation has been successfully tested in the scope of the functionalities inherent to the OS4ES V1 prototype described in deliverable D7.1 [3].

As described in D5.3 [2], the OS4ES applications developed will be used by the smart grid actors: the AGR, TSO, DSO and BRP. The OS4ES application system is composed of four software applications:

- OS4ES Aggregator Application (**OAA**): Application executed by the aggregators, providing the functionalities required by their two associated roles, the flexibility provider and the flexibility aggregator (see D1.1 [1], section 3 Roles, for role definition).
- OS4ES TSO Application (**OTA**): Application executed by the TSO.
- OS4ES DSO Application (**ODA**): Application executed by the DSO.
- OS4ES BRP Application (**OBA**): Application executed by the BRP.

These applications exchange data by means of a business framework. This framework implements the interactions between the flexibility users (TSO, DSOs and BRPs) and the aggregators defined in the OS4ES application use cases. The implementation of the business

framework used in OS4ES is based on the Universal Smart Energy Framework (USEF)¹, which provides a base software reference implementation providing a standardized way of exchanging information by means of flexibility-oriented messages. However, USEF does not cover all the interactions defined in the OS4ES application use cases, so it has been retrofitted with the features demanded by OS4ES. All information exchange, use case workflows and business intelligence not considered in USEF have been developed in T5.3 and T5.4. The main objective of this development has been the functionality provided by the aggregator, which closes the gap between the USEF framework (which provides the market interaction) and the OS4ES system (which provides the DER management capabilities).

We have named the USEF framework modified during tasks T5.3 and T5.4 as usefOS4ES. UsefOS4ES gathers the OS4ES application prototypes developed according to the design specified in D5.3 [2].

¹ <http://www.usef.info/Home.aspx>

1.4 Notations, abbreviations and acronyms

The following table list all notations, abbreviations and acronyms, used in this document.

Acronym	Description
AFC	Autonomous Frequency Control
AGR	Aggregator
ASP	Aggregated Service Provider
AVC	Autonomous Voltage Control
CRO	Common Reference Operator
DER	Distributed Energy Resources
DSO	Distribution System Operator
DVPP	Dynamic Virtual Power Plant
EAN	European Article Number
GSRN	Global Service Related Number
OAA	OS4ES Aggregator Application
ODA	OS4ES DSO Application
OTA	OS4ES TSO Application
PCC	Point of Common Coupling
PTU	Program Time Unit
TSO	Transmission System Operator
UC	Use Case
USEF	Universal Smart Energy Framework
VPP	Virtual Power Plant

2 UsefOS4ES Installation

As explained in the introduction section, the application prototypes have been developed using USEF framework, so the usefOS4ES installation is very similar to the USEF framework. The following sections have been extracted from [4] and adapted to the actual usefOS4ES framework.

2.1 Source code repository

The prototypes are located at GitLab, <https://gitlab.com/idoiamurua/usefOS4ES>. A GitLab account has been created to let the EuropeanCommission download them:

- User: OS4ESEuropeanCommission
- Password: Fg%5x3.8

An assoa cited Gmail account has also been created to manage this GitLab user:

- User: OS4ESEuropeanCommission@gmail.com
- Password: Fg%5x3.8

There are several branches at the repository:

- Master: Is the current working implementation which implements increasingly the functionalities defined in the design document [2].
- Versions: Versioned prototypes that follow the OS4ES integrated system continuous integration functionalities described in [5]. Currently the only prototype version is V1, <https://gitlab.com/idoiamurua/usefOS4ES/tree/1.0>

2.2 Code structure

The source code consists of the following folders:

- usef-build: Which contains the USEF source code to build all participant components.
- usef-environment: Which contains scripts and configuration to run USEF.
- usef-environment-tool: Which contains source code to build tooling to generate the USEF environment.

Folder usef-build is important for the development phase, and it contains pom-files to build the source code with Maven.

Folder usef-environment is important when starting using USEF. This deliverable contains several folders:

- bin: Which contains Windows batch scripts and Linux scripts the generate the environment and start the environment.
- config: Which contains configuration (which can be adjusted). The file usef-environment.yaml is used to define the configuration of the environment. The environment contains configuration of all partipants.
- template: This folder contains template files which can be changed by advanced users. The templates are used when the USEF environment is generated.

2.2.1 Environment variables

The following environment variables need to be defined correctly to be able to run the scripts which are covered in this document:

- JAVA_HOME – needs to be set to the location of the folder in which the Java JDK 8 will be installed.
- PATH – must contain a reference to the location of the bin folder where Apache Maven has been installed, and it must contain the directory \$JAVA_HOME/bin (or %JAVA_HOME%\bin on Windows). If BIND is used on the same Windows machine, the path “C:\Program Files\ISC BIND 9\bin” must be added too.
- LD_LIBRARY_PATH – (UNIX-only) must contain the path “/usr/local/lib”, as that’s the recommended place where libsodium will be installed.
- JBOSS_HOME – needs to be set to the location of the folder in which JBoss Wildfly will be installed.

2.1 Installation prerequisites

For building and using USEF, the software specified in the following sections must be installed.

2.1.1 Oracle Java SE Development Kit 8 or OpenJDK JDK 8:

To build and run source code Java 8 JDK is required. This can be downloaded from

<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html#javasejdk>

Or follow the instructions from

<http://openjdk.java.net/install/>

Note 1: The JDK must be explicitly downloaded and installed. Installing just the JRE (Java Runtime Environment) is not sufficient for building a USEF environment.

Note 2: 64 bit version of the JDK must be used, because the 64-bit encryption libraries are needed.

2.1.2 Apache Maven

To build and run source code, at least Apache Maven version 3 is required. The latest stable version can be downloaded from:

<http://maven.apache.org/download.cgi>

Note that the environment variables (M2_HOME, M2) mentioned in the Maven installation instructions are not necessary for the USEF environment.

2.1.3 JBoss Wildfly

To run USEF on an application server, JBoss Wildfly version 9.0.2 Final is required, which can be downloaded from:

<http://wildfly.org/downloads>

Please note that USEF has only tested with version 9.0.2.Final. Only this version of JBoss Wildfly is strongly recommended to be used, and not later versions.

2.1.4 H2

The USEF Reference Implementation is shipped with the H2 database for convenience purposes only. This database should not be used in production environments or any other USEF application where high performance and data integrity are deemed critical.

JBoss Wildfly 9.0.2 Final is shipped with the H2 Database Engine version 1.3.173. This version contains bugs that prevents the USEF Reference Implementation from working correctly. Beta version 1.4.190 does not contain these bugs.

The following instruction should be executed to upgrade:

- Download the Platform Independent Zip for H2 1.4.190 from

<http://www.h2database.com/html/download.html>

- Unzip the downloaded file
- Copy the file h2/bin/h2-1.4.190.jar into
\${JBOSS_HOME}/modules/system/layers/base/com/h2database/h2/main
- Remove h2-1.3.173.jar from
\${JBOSS_HOME}/modules/system/layers/base/com/h2database/h2/main
- Modify the resource root source path in
\${JBOSS_HOME}/modules/system/layers/base/com/h2database/h2/main/module.xml into h2-1.4.190.jar.

2.1.5 Libsodium

To be able to securely transmit and authenticate messages, Libsodium 1.0.8 must be built and installed.

On Windows:

- Download <https://download.libsodium.org/libsodium/releases/libsodium-1.0.8-mingw.tar.gz> to a new directory and unpack it with your favorite tool (e.g. 7-zip).
- Copy the file “libsodium-win64/bin/libsodium-18.dll” to some directory in your PATH, e.g. %JAVA_HOME%/bin, and rename the file to “sodium.dll”.

On other platforms:

- Make sure gcc and make are installed in the environment where to install USEF:

```
sudo apt-get install gcc make
```

- Download <https://download.libsodium.org/libsodium/releases/libsodium-1.0.8.tar.gz> to a new empty directory and start in this directory:

```
cd <build_directory>
```

- Unpack the archive, build it and install it:

```
tar xzf libsodium-1.0.8.tar.gz
cd libsodium-1.0.8
./configure && make
sudo make install
```

- Add /var/local/lib to the LD_LIBRARY_PATH by adding the following to :

```
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

2.1.6 Root certificate (optional, for demo purposes).

The reference implementation contains a Java key store with 3 entries, for demo purposes: usef_ri/usef-environment/template/usef.jks.

The root CA certificate can be extracted from this file as follows:

```
cd <path_to>/usef_ri/usef-environment/template
keytool -exportcert -keystore usef.jks -alias usef_ri -
file rootCA.der
(password: "usef1234")
```

This certificate needs to be installed in the Oracle JVM key store.

To do so, the following instructions should be executed with administrator rights:

```
cd $JAVA_HOME/jre/lib/security
keytool -importcert -keystore cacerts -file \
<path_to>/usef_ri/usef-environment/template/rootCA.der
(password: "changeit")
```

If another tool is used to access USEF using https, and this tool uses its own key store (e.g. postman), this root CA certificate file will also be required to be installed there.

2.1.7 ISC BIND

Installing ISC BIND 9.10.x is optional in demo environment, mandatory in production.

The procedure as described here is intended as a simple example and works for the default USEF configuration. For more information, please consult the Bind 9 Administrator Reference Manual .

On windows:

- Download <ftp://ftp.isc.org/isc/bind9/9.10.2/BIND9.10.2.x64.zip> (or later) to a new directory and unpack it.
- Locate the file “BINDInstall.exe”, right-click on it, select “Run as administrator”. The following dialog will be shown:



Figure 1: BIND installation window

- Enter the information as shown, the two password fields can be left empty. Then press “Install”.
- Copy the files “named.conf” and “usef_bind.zone” to your “C:\Program Files\ISC BIND 9\etc” directory.
Those files can be found in “usef_ri/usef-environment/config” folder.
- Manually change the following line in named.conf (as administrator):
 directory “/var/cache/bind”;

to:

directory "C:\Program Files\ISC BIND 9\etc";

- Be sure that the "C:\Program Files\ISC BIND 9\bin" folder has been added to the PATH as described in Environment variables.
- Add Full Control Rights for user "LOCAL SERVICE" to folder "C:\Program Files\ISC BIND 9\etc".
- Start the "ISC BIND" Service.

On other platforms (e.g. Ubuntu 14.04):

- Execute:

```
sudo apt-get install bind9
```

- Copy the file usef_ri/usef-environment/config/named.conf to your /etc/bind directory.
- Copy the file usef_ri/usef-environment/config/usef_bind.zone to your /var/cache/bind directory.
- Restart bind by executing:

```
sudo service bind9 restart
```

2.1.8 Hat-open-dist stack and JavaPythonBridge

Hat-open-dist stack is a XMPP-based IEC 61850 communication API developed in WP3 of OS4ES project using Python language.

JavaPythonBridge has been developed in WP6 of OS4ES project, and it provides a means of using the hat-open-dist API from within Java code.

These two software applications are required for useOS4ES because the OS4ES role of useOS4ES communicates with the DER-s via the IEC 61850 XMPP based communication standard. Hat-open-dist provides the functionality for this type of communication. As this software is developed in Python, the JavaPythonBridge which translates the function calls from Java into Python is also required.

Once the code of both applications is downloaded, the instructions detailed in the following subsections must be followed.

2.1.8.1 Dependencies

- Python >= 3.5
- Java version >= 1.8

2.1.8.2 Windows development machine setup

The "JavaPythonBridge/hat" folder along with its files should be copied to "installation_python_path\Lib/site-packages" folder.

Copy "JavaPythonBridge/client/os4es_iec61850_client.py" file to the "installation_python_path\Lib" folder.

2.1.8.3 Linux/Ubuntu operating systems

Copy the “JavaPythonBridge/hat” folder along with its files to python3.5 site-packages folder.

```
$ cd @JavaPythonBridge_path
$ sudo cp -avr hat/ /usr/local/lib/python3.5/site-
packages/
```

Copy “JavaPythonBridge/os4es_iec61850_client.py” file to the installation_python_path folder folder:

```
$ cd @JavaPythonBridge_path
$ sudo cp client/os4es_iec61850_client.py
/usr/local/lib/python3.5/
```

2.2 Environment utilization

The USEF environment can be started on Windows or Linux. The usef-environment/bin folder contains Windows batch scripts and Linux scripts:

- cleanup.cmd and cleanup.sh: Clean the USEF environment and stops the H2 database.
- prepare.cmd and prepare.sh: Build USEF and generates the USEF environment before it can be started.
- start-h2-database.cmd and start-h2-database.sh: Start the H2 database as a separate process.
- Log in the USEF database, as described in section 2.2.3, and execute the following SQL scripts to populate the AGR1 and OSES1 databases with initial data:
 - RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_zones_load_script.sql';
 - RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_contract_templates_load_script.sql';
 - RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_OSES_addressing_load_script.sql';
- start-usef-environment.cmd and start-usef-environment.sh: Start the USEF environment with JBoss Wildfly.
- stop-h2-database.cmd and stop-h2-database.sh: Stops the H2 database.
- stop-usef-environment.cmd and stop-usef-environment.sh: Stop the USEF environment.

When starting using USEF, the scripts must be run in the order mentioned to get the USEF environment up and running. Once this is done for a first time, the cleanup script has not to be executed again.

2.2.1 Starting the USEF environment

To start the USEF environment:

- Run the prepare script (The parameter `--skipBuild` skips building USEF, when USEF has already been built)
- Run the start-h2-database script.
- Log in the USEF database, as described in section 2.2.3, and execute the following SQL scripts to populate the AGR1 and OSES1 databases with initial data:
 - `RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_zones_load_script.sql';`
 - `RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_contract_templates_load_script.sql';`
 - `RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_OSES_addressing_load_script.sql';`
- Run the start-usef-environment script
- After the start-usef-environment script is executed, you can start sending messages to the participants.

2.2.2 Stopping the USEF environment

To stop the USEF environment:

- Stop the USEF Environment manually by running the stop-usef-environment script
- Stop the USEF database by running the stop-h2-database script
- Run the cleanup script

2.2.3 Accessing the USEF database

All participants, which are configured in the `usef-environment.yaml` file, are located in one H2 database file which is encrypted by AES.

To access the USEF database, connect to the following location using a web-browser:

<http://localhost:8082/>

At connection, the following screen appears:

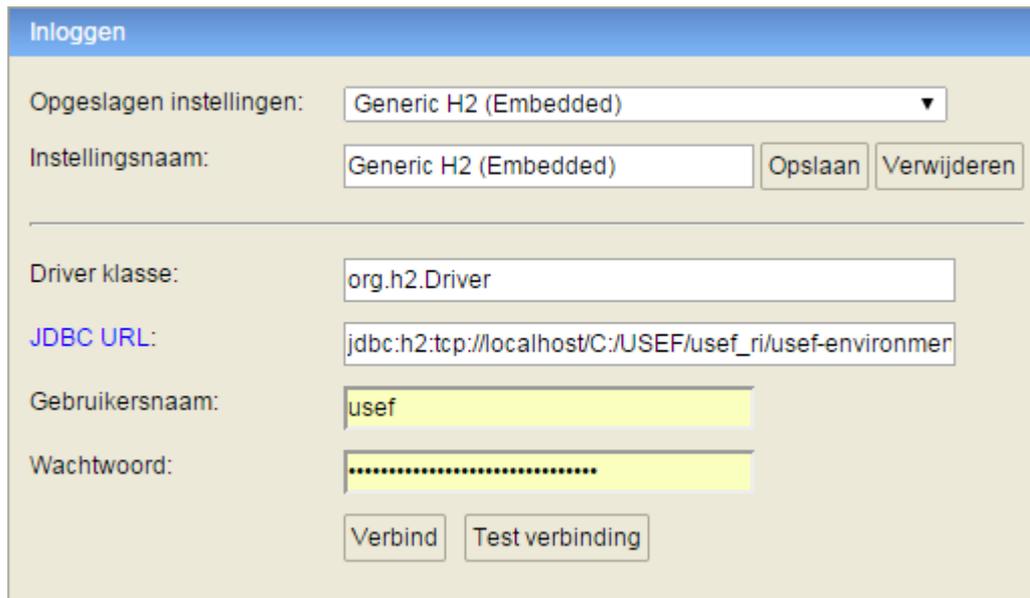


Figure 2: H2 Database connection dialog

In this dialog, the Driver, JDBC URL, username and password have to be filled according to the contents of the following file:

```
usef-environment/nodes/localhost/configuration/credentials.properties
```

The username is 'usef' without quotes and the Driver must be org.h2.Driver. The password and JDBC URL can be extracted from the DB_PASSWORD and DB_URL fields from the credentials.properties (excluding the backslashes).

Note: this file must be removed from the file system to be USEF compliant.

2.2.4 Sending messages

By default, 7 different participants are deployed which can be accessed through their own URL:

AGR1: https://agr1.usef-example.com:8443/agr1.usef-example.com_AGR/rest/MessageService/sendMessage
BRP1: https://brp1.usef-example.com:8443/brp1.usef-example.com_BRP/rest/MessageService/sendMessage
CRO1: https://cro1.usef-example.com:8443/cro1.usef-example.com_CRO/rest/MessageService/sendMessage
DSO1: https://dso1.usef-example.com:8443/dso1.usef-example.com_DSO/rest/MessageService/sendMessage
MDC1: https://mdc1.usef-example.com:8443/mdc.usef-example.com_MDC/rest/MessageService/sendMessage

TSO1: https://tso1.usef-example.com:8443/tso1.usef-example.com_TSO/rest/MessageService/sendMessage

OSES1: https://oses1.usef-example.com:8443/oses1.usef-example.com_OSES/rest/MessageService/sendMessage

The name of the domain (like agr1.usef-example.com) is configured in the usef-environment.yaml file.

With HTTP Post (header should be Content-Type = text/xml), an XML message can be send to one of the mentioned URLs:

```
<?xml version="1.0" encoding="UTF-8"?>
<TestMessage><MessageMetadata
  SenderDomain="agr1.usef-example.com"
  SenderRole="AGR"
  RecipientDomain="oses1.usef-example.com"
  RecipientRole="OSES"
  TimeStamp="2014-10-14T09:30:47-05:00"
  MessageID="00000000-0010-0000-0000-000000000001"
  ConversationID="00000000-0000-0000-0000-000000000000"
  Precedence="Routine"
  ValidUntil="2014-11-14T09:30:47-05:00" />
</TestMessage>
```

For example, Postman can be used to send this message to OSES1 from AGR1, using the following URL:

https://agr1.usef-example.com:8443/agr1.usef-example.com_AGR/rest/MessageService/sendMessage

The console or logging can be checked to see if the message is sent.

2.2.5 Aggregator GUI

This GUI is used by the aggregator to search and contract DER system services, sending SearchRequest and ContractRequest from the AGR to the OS4ES Registry. The AGR role has a Web page at the following URL:

https://agr1.usef-example.com:8443/agr1.usef-example.com_AGR/marketWise.jsp

Open System for Energy Services

OS4ES

About Contract Service

Fill in the following fields to search for services:

Initial date: 2016-07-01

End date: 2016-07-31

Service Type: Active power - generator

Zone: DE

Contract template type: Contract for APG 01

PCCId: 1234567890123456

Maximal rated active power supply:

Minimum: 2 W

Maximum: 5 W

Minimal rated active power supply:

Minimum: 7 W

Maximum: 3 W

Min activation speed MaxStartTime:

4 s

Optional parameters:

Figure 3: Web page to contract new DER system services (Market-wise contract type) –part 1

Optional parameters:

Nominal voltage:

Minimum: V

Maximum: V

Reliability:

Minimum: %

Maximum: %

Dispatchability of the DER system service

Forecastability of the DER system service

Acceptable technology:

- CHP
- PV
- Battery
- Genset
- Wind
- Load
- Washing machine
- Dishwasher
- Heat pump
- Undefined

Figure 4: Web page to contract new DER system services (Market-wise contract type) –part 2

The objective of this Web page is to allow the OAA user to contract new DER system services to update the portfolio of the flexibility aggregator. The OAA user introduces the information needed to make a search within the OS4ES registry.

In this first version of usefOS4ES, only “Active Power Generator” service types can be reserved, and the contract type is Market Wise.

When the OAA user selects the “Search services” button, a SearchRequest message is sent to the OS4ES Registry and its corresponding SearchResponse results are shown in the Web page:

Dispatchability of the DER system service
 Forecastability of the DER system service
 Acceptable technology:

CHP
 PV
 Battery
 Genset
 Wind
 Load
 Washing machine
 Dishwasher
 Heat pump
 Undefined

Search Result

Selected	Servid	PCC Identifier	PCC DERSystem identifier	PCC DERSystem TechnicalCapabilities techType	PCC DERSystem TechnicalCapabilities detType	PCC DERSystem TechnicalCapabilities sysType	PCC DERSystem DERService reliability	Show message
<input type="checkbox"/>	A5167455053069.AP1	DE00713749419DV100000000000412539	A5167455053069	PV	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A6389986046511.AP1	UW Marne West	A6389986046511	WIND	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A5183329048900.AP1	110 kV Freileitung Breklum-Haurup Mast 19 UW Vollstedt	A5183329048900	WIND	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A1742349050702.AP1	DE00053525560SZP00000000000580765	A1742349050702	CHP	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A7415653061712.AP1	Husum 61020	A7415653061712	PV	NONDETERMINISTIC	GEN	1.0	[+]

Figure 5: Web page showing the SearchResponse results

The OAA user can select the “+” link to see a more detailed information of the information provided by the OS4ES Registry about a specific service:

Service ID: A5167455053069.AP1

PCC.DERSystem.TechnicalCapabilities.techType - -	PV
PCC.DERSystem.TechnicalCapabilities.detType - -	NONDETERMINISTIC
PCC.DERSystem.TechnicalCapabilities.sysType - -	GEN
PCC.DERSystem.DERService.reliability - -	1.0
PCC.DERSystem.identifier - -	A5167455053069
PCC.identifier - -	DE00713749419DV100000000000412539
PCC.DERSystem.DERService.DERServiceAPG.Rated.APG.maxP - -	500000.0
PCC.DERSystem.DERService.DERServiceAPG.Rated.APG.minP - -	0.0
PCC.DERSystem.DERService.ContractTemplate.contractType - -	MARKET_WISE
PCC.DERSystem.DERService.ContractTemplate.identifier - -	ConAPG01
PCC.DERSystem.DERService.ContractTemplate.acceptableNumDenialOfDelivery - -	0.0
PCC.DERSystem.DERService.ContractTemplate.enableFutures - -	false
PCC.DERSystem.DERService.ContractTemplate.exclusive - -	true
PCC.DERSystem.DERService.ContractTemplate.currency - -	EUROCENT

PriceReference: ActiveEnergyDeliveredFixedPrice

Valid from:
Valid until
Units: currency/kWh
BasePrice: 555.0
MultPrice: 1.0
AddPrice: 0.0

PriceReference: ActiveEnergyReservedAsOptions

Figure 6: Web page showing detailed info of a specific service – Part1

PriceReference: ActiveEnergyDeliveredFixedPrice
 Valid from:
 Valid until
 Units: currency/kWh
 BasePrice: 555.0
 MultPrice: 1.0
 AddPrice: 0.0

PriceReference: ActiveEnergyReservedAsOptions
 Valid from:
 Valid until
 Units: currency/kWh
 BasePrice: 322.0
 MultPrice: 1.0
 AddPrice: 0.0

PriceReference: ActiveEnergyDeviation
 Valid from:
 Valid until
 Units: currency/kWh
 BasePrice: 2.0
 MultPrice: 1.0
 AddPrice: 0.0

Figure 7: Web page showing detailed info of a specific service – Part2

Next, when the OAA user selects a service to contract, a ContractRequest message is sent to the OS4ES Registry and its corresponding ContractResponse results are shown in the Web page:

Search Result

Selected	ServId	PCC Identifier	PCC DERSystem Identifier	PCC DERSystem TechnicalCapabilities techType	PCC DERSystem TechnicalCapabilities detType	PCC DERSystem TechnicalCapabilities sysType	PCC DERSystem DERService reliability	Show message
<input checked="" type="checkbox"/>	A5167455053069.AP1	DE00713749419DV10000000000412539	A5167455053069	PV	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A6389986046511.AP1	UW Marne West	A6389986046511	WIND	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A5183329048900.AP1	110 kV Freileitung Breklum-Haurup Mast 19 UW Vollstedt	A5183329048900	WIND	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A1742349050702.AP1	DE00053525560SZP0000000000580765	A1742349050702	CHP	NONDETERMINISTIC	GEN	1.0	[+]
<input type="checkbox"/>	A7415653061712.AP1	Husum 61020	A7415653061712	PV	NONDETERMINISTIC	GEN	1.0	[+]

Contract Result

ServID	ContractID	Result	Message
A5167455053069.AP1	1467367427912	Accepted	Accepted

Figure 8: Web page showing the ContractResponse results

In this initial version of useOS4ES, when the ContractResponse is received, the AGR role will send a ReserveRequest to the OS4ES Registry, and when the corresponding Reserveresponse

is received, a ControlCommand message is sent to the DER system. This exchange of messages can be seen in the log files of the AGR and OSES roles (see section 2.3.4 for USEF logging information).

2.2.6 JBoss Management Console

The JBoss Management Console can be found on URL:

<http://localhost:9990>

with user “usef” and password “usef”

Click on the tab “Runtime” and then “Manage Deployments” to check if the war files are deployed:

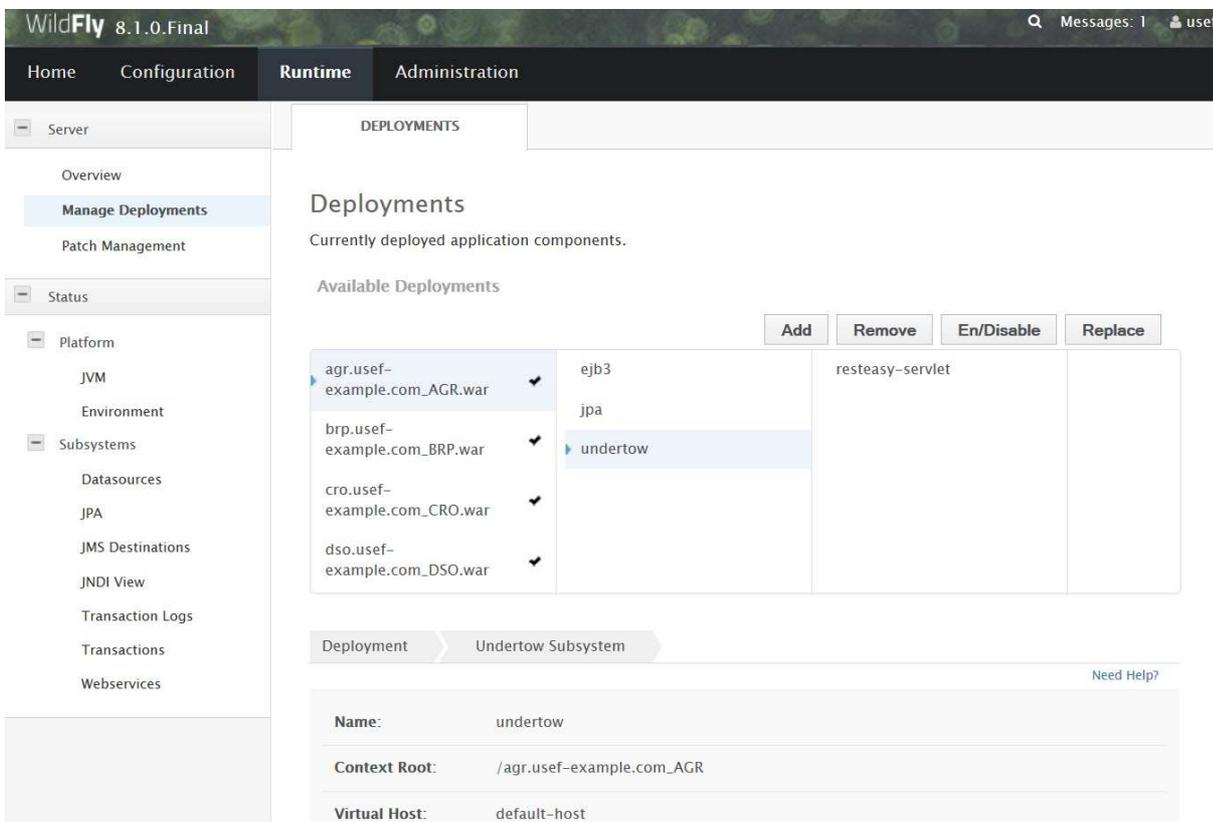


Figure 9: WildFly mangement console

At the bottom the value of “Context Root” is displayed, which is in this case “/agr1.usef-example.com_AGR”. That is, messages can be sent to the URL:

https://agr1.usef-example.com:8443/agr1.usef-example.com_AGR/rest/MessageService/sendMessage

2.2.7 Nodes folder

For every node which has been configured in the usef-environment.yaml file, the usef-environment/nodes folder contains a folder with the running configuration and files which

are used by USEF and JBoss Wildfly. By default, the usef-environment/nodes folder contains the following folders:

- configuration: The configuration folder with various configuration files.
- content: Folder which contains static files.
- data: The H2 database file is located in this folder.
- deployments: The WAR file per participant is deployed via this folder.
- lib: Library folder.
- log: Folder which contains per participant the log-files.
- temp: Temporary folder.

The content of these folders should not be changed.

2.3 Configuration

2.3.1 Adjusting the usef-environment.yaml configuration

By default, the usef-environment.yaml file (located in the usef-environment/config folder) contains the configuration for 7 USEF participants (1 AGR, 1 BRP, 1 CRO, 1 DSO, 1 MDC, 1 TSO and 1 OSES) for node 'localhost'. The usef-environment.yaml file can be opened with a text editor, like Notepad++. At the bottom of the file, the domains and its participants are defined.

The usef-environment.yaml file describes all configuration items, which are explained in [Annex A](#) and in the file itself.

The standard value of 'recipient_endpoint' assumes no proxy is used. If you are configuring a proxy, this entry should be removed.

If the usef-environment.yaml or the standalone-usef.xml file are changed, then:

- Stop the USEF Environment manually by running the stop-usef-environment script or by pressing Ctrl+C in the console.
- Stop the USEF database by running the stop-h2-database script
- Run the cleanup script.
- Run the prepare script
- Run the start-h2-database script
- Log in the USEF database, as described in section 2.2.3, and execute the following SQL scripts to populate the AGR1 and OSES1 databases with initial data:
 - RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_zones_load_script.sql';
 - RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_contract_templates_load_script.sql';
 - RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_OSES_addressing_load_script.sql';
- Run the start-usef-environment script

Check if the configuration affects the behavior of USEF (Notice that the log files and folder structure can be checked to see if the changed parameters had effect, e.g. are all roles deployed).

2.3.2 Adjusting `iec61850_client.yml` configuration

The `iec61850_client.yml` file (located in the `usef-environment/template` folder) is required by the OS4ES participant only. When the USEF environment is started, it will be automatically copied to the OS4ES role configuration folder. The `iec61850_client.yml` file can be opened with a text editor, like Notepad++.

The `iec61850_client.yml` file contains the required parameters regarding the DER addressing for XMPP protocol, and that are used by the `hat-open-dist` stack specified in section 2.1.8.

If the `iec61850_client.yml` is changed, then:

- Stop the USEF Environment manually by running the `stop-usef-environment` script or by pressing `Ctrl+C` in the console.
- Stop the USEF database by running the `stop-h2-database` script
- Run the cleanup script.
- Run the prepare script
- Run the `start-h2-database` script
- Log in the USEF database, as described in section 2.2.3, and execute the following SQL scripts to populate the AGR1 and OS4ES1 databases with initial data:
 - `RUNSCRIPT FROM 'usef_ri /usef-environment/config/usef_AGR_zones_load_script.sql';`
 - `RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_AGR_contract_templates_load_script.sql';`
 - `RUNSCRIPT FROM 'usef_ri/usef-environment/config/usef_OS4ES_addressing_load_script.sql';`
- Run the `start-usef-environment` script

2.3.3 Zones, contract templates and DER addressing related data in USEF database

Before executing the “`start-usef-environment`” script, the following SQL scripts must be loaded in USEF database:

- `'usef_ri /usef-environment/config/usef_AGR_zones_load_script.sql'`: It initializes TSO, DSO and ZONE tables of the AGR1 database, with the zones related information. The default values are:

```
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.TSO (ID, DOMAIN) VALUES (1,
'tso1.usef-example.com');
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.TSO (ID, DOMAIN) VALUES (2,
'tso2.usef-example.com');

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.DSO (ID, DOMAIN, TSO_ID)
VALUES (1, 'dso1.usef-example.com', 1);
```

```

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.DSO (ID, DOMAIN, TSO_ID)
VALUES (2, 'dso2.usef-example.com', 1);
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.DSO (ID, DOMAIN, TSO_ID)
VALUES (3, 'dso3.usef-example.com', 2);

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.ZONE (ID, DSO_ID,
ZONE_OS4ES_ID) VALUES (1, 1, 'DE.TENNET.AVACON');
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.ZONE (ID, DSO_ID,
ZONE_OS4ES_ID) VALUES (2, 2, 'DE.TENNET.SCHLESWIG-HOLSTEIN-NETZ');
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.ZONE (ID, DSO_ID,
ZONE_OS4ES_ID) VALUES (3, 1, 'DE.TSO1.DSO1');
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.ZONE (ID, DSO_ID,
ZONE_OS4ES_ID) VALUES (4, 2, 'DE.TSO1.DSO2');
INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.ZONE (ID, DSO_ID,
ZONE_OS4ES_ID) VALUES (5, 3, 'DE.TSO2.DSO3');

```

- 'usef_ri/usef-environment/config/usef_AGR_contract_templates_load_script.sql': It initializes CONTRACT_TEMPLATE_TYPE and CONTRACT_PRICE_REF_TYPE tables of the AGR1 database. The default values are:

```

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.CONTRACT_TEMPLATE_TYPE (ID,
SERVICE_TYPE, CONTRACT_TEMPLATE_OS4ES_ID, CONTRACT_TEMPLATE_NAME,
CONTRACT_TYPE, ACCEPTABLE_NUM_DENIAL_DELIVERY, ENABLE_FUTURES,
EXCLUSIVE, CURRENCY) VALUES (1, 'ActivePowerGen', 'ConAPG01',
'Contract for APG 01', 'MARKET_WISE', 0, false, true, 'EUROCENT');

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.CONTRACT_PRICE_REF_TYPE (ID,
CONTRACT_TEMPLATE_TYPE_ID, REF_PRICE, MULT_PRICE, ADD_PRICE, UNITS)
VALUES (1, 1, 'ActiveEnergyDeliveredFixedPrice', 1, 0,
'currency/kWh');

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.CONTRACT_PRICE_REF_TYPE (ID,
CONTRACT_TEMPLATE_TYPE_ID, REF_PRICE, MULT_PRICE, ADD_PRICE, UNITS)
VALUES (2, 1, 'ActiveEnergyReservedAsOptions', 1, 0, 'currency/kWh');

INSERT INTO AGR1_USEF_EXAMPLE_COM_AGR.CONTRACT_PRICE_REF_TYPE (ID,
CONTRACT_TEMPLATE_TYPE_ID, REF_PRICE, MULT_PRICE, ADD_PRICE, UNITS)
VALUES (3, 1, 'ActiveEnergyDeviation', 1, 0, 'currency/kWh');

```

- 'usef_ri/usef-environment/config/usef_OSES_addressing_load_script.sql': It initializes OAA_GATEWAY_ADDRESSING table of the OSES1 database. The default values are:

```

INSERT INTO OSES1_USEF_EXAMPLE_COM_OSES.OAA_GATEWAY_ADDRESSING (ID,
DER_JABBER_ID, REG_JABBER_ID, IP_ADDRESS, XMPP_SERVER, XMPP_PORT,
XMPP_DOMAIN, XMPP_USER, XMPP_PASSWORD, XMPP_RESOURCE) VALUES (1,
'der1@os4eshypertech.gr/SERV1', 'ap_user@os4eshypertech.gr/TNO',
'oses1.usef-example.com', 'adsl.hypertech.gr', '52',
'os4eshypertech.gr', 'os4es_test', 'test', 'AGR1');

```

This table contains the following fields required for the XMPP communication from the XMPP client of the OSES role to the XMPP clients of the OS4ES Registry and DER:

- DER_JABBER_ID: the jabberId of the DER XMPP client.
- REG_JABBER_ID: the jabberId of the Registry XMPP client.
- IP_ADDRESS: IP address of the OSES role.
- XMPP_SERVER: XMPP server IP address.
- XMPP_PORT: XMPP server port number.
- XMPP_DOMAIN: XMPP domain in the server.
- XMPP_USER: XMPP user name for the OSES role.
- XMPP_PASSWORD: XMPP user password for the OSES role.
- XMPP_RESOURCE: XMPP resource name for the OSES role.

These default values should be changed according to the actual zones, contract templates and DER addressing values that are required. A detailed description of these tables can be found in [2].

2.3.4 Logging

The USEF Reference Implementation uses the LogBack framework for application logging. Information exchanged may contain confidential information that must not be logged to the default log files. The LogBack configuration facilitates this.

For more information on this framework, please visit (<http://logback.qos.ch/>).

The logging configuration of USEF Reference Implementation contains information on e.g. log file locations and log levels. Each deployment type has its own logging configuration files, located in the deployment folder.

Below is an example of where these files are located:

- usef-deployments/usef-deployment-agr/src/main/resources/logback.xml and
- usef-deployments/usef-deployment-agr/src/main/resources/LogBackAgr.xml

Note that the filters specified should not be modified, because that could result in a non-compliant USEF implementation.

After installation of the USEF Reference Implementation, each deployment has its own logging configuration XML file that enables you to change the log levels for this specific deployment.

Below is an example of where this file is located:

- usef-environment/nodes/localhost/configuration/agr1.usef-example.com_AGR/LogBackWithAdditionalLoggers.xml

Application Server logging for generic non-application logging is configured in

- usef-environment/template/standalone-usef.xml

2.3.5 Reserved TCP ports

JBoss Wildfly and the H2 database are using TCP ports which are fixed. So, if any application or process is running on these TCP ports, USEF will fail to start. When the USEF environment

is started, the TCP ports will be checked and report the PID of the process which is using the TCP port, if so.

Please, make sure the following TCP ports are reserved for USEF:

- 443: the HTTPS port, used by the Apache proxy (optional)
- 8082: the HTTP port, used by the H2 database console
- 8443: the HTTPS port, used by JBoss Wildfly
- 9092: the H2 database
- 9990: the JBoss management console HTTP port
- 9993: the JBoss management console HTTPS port

2.3.6 Configure resolver entries

For participants to be able to reach each other, their domain names must resolve to a valid network address. For example, if the domains, “agr1.usef-example.com” and “dso1.usef-example.com” are hosted by the local machine, then those two domains must resolve to the IP address of the local machine. A separate domain could be specified where Wildfly will be running.

If BIND is installed, the IP addresses will be resolved automatically, otherwise – for demo purposes – this can be worked around by adding the following lines to “/etc/hosts” (or “C:\Windows\System32\drivers\etc\hosts” on Windows):

```
127.0.0.1 agr1.usef-example.com
127.0.0.1 brp1.usef-example.com
127.0.0.1 cro1.usef-example.com
127.0.0.1 dso1.usef-example.com
127.0.0.1 mdc1.usef-example.com
127.0.0.1 tso1.usef-example.com
127.0.0.1 oses1.usef-example.com
127.0.0.1 jboss.usef-example.com
```

2.3.7 Configuring a proxy server (optional)

The Apache configuration must be modified in the following ways. Instead of “usef-example.com” use your own registered domain. As an example, the following configuration steps can be followed:

- Create the file /etc/apache2/conf-available/usef-proxy.conf with the following contents:

```
<VirtualHost _default_:443>
# This VirtualHost's identity
ServerAdmin webmaster@usef-example.com
ServerName www.usef-example.com
ServerAlias *.usef-example.com
```

```
# Enable SSL proxy
SSLEngine on
SSLProxyEngine on

# SSLProxyCheckPeerName can be set to off for test
purposes
SSLProxyCheckPeerName on

# SSL certificate files
SSLCertificateKeyFile "/etc/apache2/usef/usef.key"
SSLCertificateFile "/etc/apache2/usef/usef.crt"

# Proxy matching rules
# matching URLs for testing purposes
ProxyPassMatch ^/(.*\.usef-example\.com_.*)$
https://jboss.usef-example.com:8443/$1
# matching USEF compliant URLs
ProxyPassMatch ^/USEF/2014/I/(.*)/(.*)/SignedMessage$ \
  https://jboss.usef-
example.com:8443/$1_$2/rest/ReceiverService/receiveMessage
</VirtualHost>
```

- Create the folder `/etc/apache2/usef` as follows:

```
sudo mkdir /etc/apache2/usef
```

- Generate or obtain a private key and certificate and place them as `usef.key` and `usef.crt` in `/etc/apache2/usef`.
- Enable all configuration items and restart apache:

```
sudo a2enmod ssl proxy proxy_http
sudo a2ensite usef-proxy
sudo service apache2 restart
```

Please note that in the `environment.yaml` file, the `recipient_endpoint` parameter must be removed to enable use of the proxy. This is documented in section 2.3.1. This will also enable full USEF compliant URLs.

2.3.8 Resolving participant information

Every participant participating in USEF must be able to resolve information of every other participant it communicates with. This information consists of the IP address and a public key for sealing and unsealing messages.

2.3.8.1 Secure information provision using DNSSEC

To guarantee using DNSSEC for secure communication between participants, a recursive DNS server must be configured which is dedicated to USEF DNS queries. To be USEF compliant, the path to this server must be 100% trusted and the server must only allow secure DNSSEC responses from a participant's DNS server.

With Bind version 9.9.5, this can be configured by defining option 'dnssec-must-be-secure <domain> yes;' in the options section which will only accept secure DNSSEC responses from the hierarchy starting with <domain>.

Installing ISC BIND 9.10.x is optional in demo environment, mandatory in production. It will have the secure information provision option 'dnssec-must-be-secure <domain> yes;' enabled already.

2.3.8.2 Configuring DNS in wildfly

By default, Wildfly is configured to use a name server which is located on localhost (127.0.0.1). When the file `participants_dns_info.yaml` is used to resolve participant keys and IP addresses of participants are registered in the hosts file, this will still work without local DNS server.

If a DNS server is to be used, the `willdfly.properties` file has to be changed. This file can be found in folder `usef-environment/template`. The property `dns.server=127.0.0.1` can then be changed to refer to the IP address of the actual trusted DNS server. Changes to this file will be taken into account when the USEF environment is rebuilt by using the prepare script.

2.3.8.3 Resolving without DNS server

For demonstration and testing purposes, the file `participants_dns_info.yaml` is automatically configured for the participants which are deployed by default. This file contains the public key for sealing and unsealing and can be found in the generated folder `usef-environment/nodes/localhost/configuration` after the prepare script is executed.

In addition to the public key for sealing and unsealing and in absence of a DNS server, the local resolver library must be able to resolve the names of the participants to valid IP addresses. An example of how to do this by defining entries in the system's hosts file is described in section 2.3.6.

Annex

Annex A. Configuration of scheduled processes

The usef-environment.yaml file (located in the usef-environment/config folder) contains the configuration of all USEF roles and processes. This includes scheduled times for processes that are time triggered. This appendix explains the default configuration for these scheduled times and how they are related to each other.

For ease of understanding, these scheduled times are explained in two parts:

- **Extraday triggers:** triggers that are fired once a day or even less frequently. These triggers are typical for the processes in plan, validate (excluding flex trading) and settlement phases.
- **Intraday triggers:** triggers that are fired multiple times per day. These triggers are typical for the flex trading processes and processes in the operate phase.

Extraday triggers

The table below describes all processes that are triggered once a day or less frequently, to which roles they apply, which configuration parameters are involved and the default values. Also a remark is included to indicate if there is a relation to other processes and/or configuration parameters.

Each value is defined as the time on which the trigger is fired. If a configuration parameter is defined for multiple roles, the “{role}” is replaced by the applicable role in the configuration file.

Process	AGR	BRP	DSO	MDC	yaml configuration parameters	remark
Initialize Non-UDI Clusters	00:01				agr_initialize_non_udi_time_of_day	
Common Reference Query				05:00	mdc_common_reference_query_time	
Common Reference Update	11:00	11:00	11:00		{role}_common_reference_update_time	
Initialize Planboard	12:00	12:00	12:00		{role}_initialize_planboard_time {role}_initialize_planboard_days_interval brp_initialize_planboard_days_ahead	The common reference update of all AGRs, BRPs and DSOs must be finished before the planboards are initialized to get consistent results.
Collect Forecast (DSO)			13:00		dso_connection_forecast_time dso_connection_forecast_days_interval	The DSO needs to generate its non-aggregator forecast before it is able to process D-prognoses. Therefore the DSO Collect Forecast process must be scheduled earlier than the AGR equivalent.
Collect Forecast (AGR)	13:30				agr_connection_forecast_time agr_connection_forecast_days_interval	
Finalize A-Plan (AGR)	19:30				day_ahead_gate_closure_time - agr_finalize_aplans_ptus_before_gate_closure * ptu_duration	The AGR must finalize A-plans and/or start the validate phase before the DSO can create missing D-prognoses.
Create Missing Prognoses		20:00	20:00		day_ahead_gate_closure_time - day_ahead_gate_closure_ptus * ptu_duration	

Finalize A-Plan (BRP)		21:00			day_ahead_gate_closure_time - brp_finalize_aplans_ptus_before_gate_closure * ptu_duration	The BRP can only finalize A-plans after it has created any missing A-plans.
Day Ahead Gate Closure Time	22:00	22:00	22:00		day_ahead_gate_closure_time	All participants must agree on this time.
Initiate Settlement Time	01:00	10:00	11:00		{role}_initiate_settlement_time {role}_initiate_settlement_day_of_month	
Dispute Settlement Without Response		22:00	22:00		{role}_settlement_message_disposal_time	
Collect Orange Regime Data			12:00		dso_initiate_collect_orange_regime_data_time dso_initiate_collect_orange_regime_data_day_of_month	

Table 1: Extraday triggers

Intraday triggers

The table below describes all processes that are triggered multiple times a day, to which roles they apply, which configuration parameters are involved and the default values. Each value is defined as an initial delay, followed by the time interval between subsequent triggers (initial delay -> interval).

If a configuration parameter is defined for multiple roles, the “{role}” is replaced by the applicable role in the configuration file.

Process	AGR	BRP	DSO	unit	yaml configuration parameters
Flex Offer Schedule	5 -> 900			seconds	agr_flexoffer_initial_delay_in_seconds -> agr_flexoffer_interval_in_seconds
Determine Net Demands	5 -> 300			seconds	agr_determine_netdemands_initial_delay_in_seconds -> agr_determine_netdemands_interval_in_seconds
Control ADS	5 -> 900			seconds	agr_control_ads_initial_delay_in_seconds -> agr_control_ads_interval_in_seconds
Flex Order		5 -> 30	5 -> 900	seconds	{role}_flexorder_initial_delay_in_seconds -> {role}_flexorder_interval_in_seconds
Meter Data Query Expiration			0 -> 60	minutes	0 -> dso_meter_data_query_expiration_check_interval_in_minutes
Operate			1 -> 60	seconds	dso_operate_initial_delay_in_seconds -> dso_operate_interval_in_seconds
Identify Changes in Forecast	5 -> 900			seconds	agr_identify_change_in_forecast_initial_delay_in_seconds -> agr_identify_change_in_forecast_interval_in_seconds=900
Non-UDI Retrieve ADS Goal Realization	15 -> 15			minutes	agr_non_udi_retrieve_ads_goal_realization_interval_in_minutes -> agr_non_udi_retrieve_ads_goal_realization_interval_in_minutes

Table 2: Intraday triggers

References

- [1] OS4ES, «D1.1 OS4ES Requirement specification for an OS4ES».
- [2] OS4ES, «D5.3 Design of the DER management applications».
- [3] OS4ES, «D7.1 Validation results».
- [4] USEF Foundation, «USEF The Framework Implemented - Installation Manual,» 2016.
- [5] OS4ES, «D6.4 OS4ES Integrated system».