iADAATPA Domibus connector. Technical Document



Description of the configuration and setup to use Domibus as AS4 enabler for iADAATPA platform

## Table of Contents

[Table of Contents 1](#_Toc533407048)

[Introduction 2](#_Toc533407049)

[Contributors 2](#_Toc533407050)

[Revisions 2](#_Toc533407051)

[Overall Setup 3](#_Toc533407052)

[Domibus Installation & Setup 4](#_Toc533407053)

[Client and Server Bridges 7](#_Toc533407054)

[Testing and verifying the setup 9](#_Toc533407055)

# Introduction

This document contains a description of the setup and configuration to use Domibus to enable AS4 communications between an iADAATPA client and server. The testing process and the validation was made by Everis using the client and server bridges developed by Pangeanic.

## Contributors

The following consortium members were involved in the development of this document.

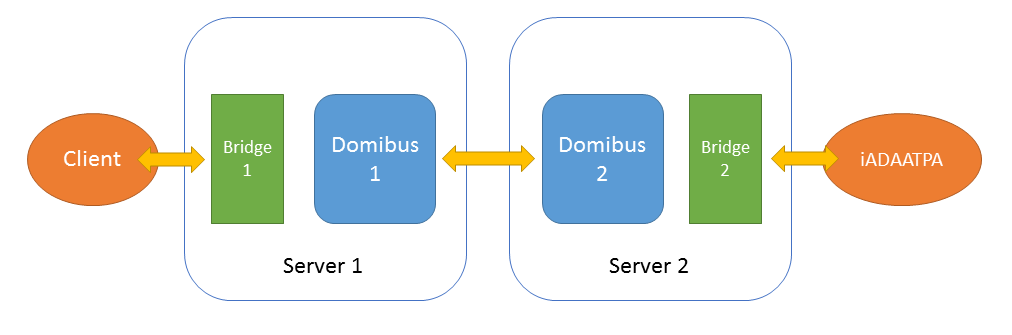
|  |  |
| --- | --- |
| **Member** | **Organisation** |
| Amando Estela | Pangeanic |
|  |  |
|  |  |
|  |  |

## Revisions

|  |  |  |
| --- | --- | --- |
| **Who** | **What** | **When** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Overall Setup

First of all, the architecture of the environment will be described using the following picture:



It is divided in different parts:

* **Client**: It is the iADAATPA client that sends the petition for the translation.
* **Bridge 1**: It is the endpoint for the client side, a Python program.
* **Domibus 1**: It is the first point of the Domibus connector, installed in the Server 1 and where the **File System Plugin** is enabled.
* **Domibus 2**: It is the second point of the Domibus connector, installed in the Server 2 and where the **File System Plugin** is enabled.
* **Bridge 2**: It is the enpoint for the server side (in this case, it is iADAATPA Server).
* **iADAATPA**: It is the iADAATPA platform, with its original endpoint.

# Domibus Installation & Setup

We will be using an unmodified distribution of Domibus and the File Plugin for version V3.3.3. There’s no reason at all the next version (V4 for instance) are not used but at the time of tests V3.3.3 was the most stable distribution.

The distributions and the installation and setup documentation that will be used can be found at https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus+-+v3.3.3

The required documents are:

Quick Start Guide: https://ec.europa.eu/cefdigital/wiki/download/attachments/59196398/%28eDelivery%29%28AP%29%28QSG%29%28Domibus%203.3.3%29%282.2.1%29.pdf?version=1&modificationDate=1529586185594&api=v2

Testing Guide:

<https://ec.europa.eu/cefdigital/wiki/download/attachments/59196398/%28CEFeDelivery%29.%28AccessPoint%29.%28Test%20Guide%29.%28v1.04%29.pdf?version=1&modificationDate=1523001873258&api=v2>

Administration Guide:

<https://ec.europa.eu/cefdigital/wiki/download/attachments/59196398/%28eDelivery%29%28AP%29%28AG%29%283.3.3%29%281.10%29.pdf?version=1&modificationDate=1529585768087&api=v2>

File System Plugin:

<https://ec.europa.eu/cefdigital/wiki/download/attachments/59196398/%28eDelivery%29%28FS%29%28AG%29%281.4%29.pdf?version=1&modificationDate=1529585909742&api=v2>

The test setup was installed in 2 Amazon AWS machines where Ubuntu 16.04 was running. An initial check is required to verify the basic requirements are fulfilled regarding Java, Apache Tomcat and MySQL.

The two machines are identified as BLUE and RED nodes and will be the internal corners of the AS4 Architecture.

The installation checklist in both machines was as follows:

Installing the distribution

mkdir cef

cp domibus-distribution-3.3.3-tomcat-full.zip cef

cd cef

unzip domibus-distribution-3.3.3-tomcat-full.zip

cd ..

mkdir tmp

cp domibus-distribution-3.3.3-sample-configuration-and-testing.zip tmp

cd tmp

unzip domibus-distribution-3.3.3-sample-configuration-and-testing.zip

cp -R conf/\* ../cef/domibus/conf

Setting up the Database

cd ../cef/sql-scripts/

ubuntu@PGMT-Dev:~/cef/sql-scripts$ mysql -uroot -pmypassword -e "create schema domibus"

ubuntu@PGMT-Dev:~/cef/sql-scripts$ mysql -uroot -pmypassword -e "alter database domibus charset=utf8 collate=utf8\_bin"

ubuntu@PGMT-Dev:~/cef/sql-scripts$ mysql -uroot -pmypassword -e "create user edelivery@localhost identified by 'edelivery'"

ubuntu@PGMT-Dev:~/cef/sql-scripts$ mysql -uroot -pmypassword -e "grant all on domibus.\* to edelivery@localhost"

ubuntu@PGMT-Dev:~/cef/sql-scripts$ mysql -uroot -pmypassword domibus < mysql5innoDb-3.3.3.ddl

cp mysql-connector-java-5.1.45-bin.jar cef/domibus/lib

ubuntu@PGMT-Dev:/etc/mysql$ sudo vi mysql.conf.d/mysqld.cnf

Configuring Environment

bin/setenv.sh

export CATALINA\_HOME=/home/ubuntu/cef/domibus

export JAVA\_OPTS="$JAVA\_OPTS -Dfile.encoding=UTF-8 -Xms128m -Xmx1024m"

export JAVA\_OPTS="$JAVA\_OPTS -Ddomibus.config.location=$CATALINA\_HOME/conf/domibus"

chmod +x setenv.sh

chmod +x catalina.sh

chmod +x startup.sh

chmod +x shutdown.sh

Installing and configuring File System plugin

mkdir fileplugin

cp domibus-distribution-3.3.3-default-fs-plugin.zip fileplugin/

cd fileplugin/

unzip domibus-distribution-3.3.3-default-fs-plugin.zip

cp conf/domibus/plugins/lib/domibus-default-fs-plugin-3.3.3.jar ../cef/domibus/conf/domibus/plugins/lib

cd conf/domibus/plugins/config/tomcat/

cp fs-plugin.\* /home/ubuntu/cef/domibus/conf/domibus/plugins/config

cd /home/ubuntu/

mkdir filestore

cd cef/domibus/conf/domibus/plugins/config/

vi fs-plugin.properties

fsplugin.messages.location=/home/ubuntu/filestore

Final Tasks

(start domibus)

(reorder message filters)

cd /home/ubuntu/filestore/OUT

vi metadata.xml

(upload pmode-file.xml)

# Client and Server Bridges

Both bridges are Python3.5 scripts that can be installed anywhere provided:

each one access the file repository used by Domibus

the client bridge can be reached by the iADAATPA client

the server bridge can access iADAATPA server

To start the CLIENT bridge

cd {path where the python bridge is installed}

sudo python3.5 sClient.py

That will start the bridge listening for requests from the client and mocking up iADAATPA endpoint.

It will also start monitoring the filesystem for incoming (translated) files, the directory where files will appear is filestore/IN under the Domibus installation directory.

Remember to start the bridge as a service if you plan to keep the program running unattended.

To start the SERVER bridge

cd {path where the python bridge is installed}

sudo python3.5 sServer.py

that Will start the bridge and will start monitoring the filesystem for incoming files, the directory where files will appear is filestore/IN under the Domibus installation directory.

When a file appears the file is read and sent to iADAATPA server waiting for the translation. After the file is translated it is copied into filestore/OUT

# Testing and verifying the setup

A simple test consists in sending a HTTP POST to the client bridge and send afterwards another POST to retrieve the translated file.

1. Translation request:

POST /api/atranslatefile HTTP/1.1

Host: 54.171.7.119:8090

Content-Type: application/json

{

"token":"123456",

"source":"es",

"target":"en",

"fileType":"txt",

"file":"SG9sYSEKRXN0YSBlcyBvdHJhIGzDrW5lYS4="

}

The response:

{"success": "true", "data": {"guid": 1}, "error": "null"}

2. Retrieval request:

POST /api/aretrievefiletranslation HTTP/1.1

Host: 54.171.7.119:8090

Content-Type: application/json

{

"token":"123456",

"guid":"**1**"

}

The response while the file is not translated:

{"error": {"code": 16, "statusCode": 400, "message": "Missing <guid>"}, "success": "false", "data": "null"}

The response when the file is translated:

{"error": "null", "success": "true", "data": {"guid": "1", "file": "SGVsbG8hIAoKVGhpcyBpcyBhbm90aGVyIGxpbmUuCg==", "fileType": "txt"}}