EUROPEAN COMMISSION

DIRECTORATE-GENERAL FOR INFORMATICS

Piloting Domibus integration with CEF EBSI (blockchain): Building and Configuring

Published on 8 September 2022

Document Author(s)

Author Name	Role
Joze Rihtarsic	Software Developer

Document Reviewer(s)

Reviewer Name	Role
Bogdan Dumitriu	eDelivery Project Officer

Document Approver(s)

Reviewer Name	Role
Bogdan Dumitriu	eDelivery Project Officer

This document was created as a deliverable in the 2020 ISA^2 Innovative Public Services (IPS) action.

Table of Contents

1 Introduction

The purpose of piloting Domibus integration with CEF EBSI (blockchain) was to explore several areas where the CEF eDelivery message exchange could be enriched with functionality relying on the blockchain technology. In a practical sense, the exploration consisted of modifying the **Domibus** sample software to use the **CEF EBSI infrastructure** to pilot two different scenarios that with the potential to provide significant added value to the users of CEF eDelivery.

Domibus is an Access Point implementation that is conformant to the CEF eDelivery AS4 profile, maintained by the European Commission, with the purpose to ensure the safe, non-repudiable and reliable message exchanges.

The European Blockchain Services Infrastructure (EBSI) is a joint initiative from the European Commission and the European Blockchain Partnership (EBP) established to deliver cross-border public services using blockchain technology in the EU.

1.1 Purpose of this document

This document is intended for developers, architects and analysts who would like to study, build and/or reuse the deliverables created during the pilot.

The document provides instructions for installing and testing the pilot artefacts that can be used as a basis for further prototyping and development of enriched message exchange using blockchain technology.

1.2 Limitations

Before the results can be used for production purposes, additional quality and reliability assurance work must be carried out and, at the minimum, a service level agreement would need to be established between the concerned eDelivery network and EBSI.

2 Build artefacts

In order to build and run the pilot components, the following tools/applications must be installed:

- **Java JDK 1.8** (tested with Oracle JDK 1.8 and Oracle JDK 11, not tested with OpenJDK versions)
- **Maven 3.6+**
- **docker 19.03+** (needed to build and run demo environment)
- **docker-compose 1.24+** (needed to build and run demo environment)
- **linux bash** (to build demo docker image using example build.sh script, a standard Linux shell/a command interpreter is needed)

2.1 EBSI Java client library

The EBSI Java client library for eDelivery is a Java client for calling EBSI services. It is based on:

- auth0.com: for JWT session token generation and
- web3j: which is a lightweight, reactive, type safe library for Java for connecting to Ethereum blockchains.

NOTE: The libraries are not Domibus-dependent and thus can be equally used also for other Access Point implementations based on Java.

(The library is already built and accessible from the address: https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-snapshots:eu%2Feuropa%2Fec%2Fedelivery%2Febsi%2Febsi-core%2F1.0-SNAPSHOT)

```
# checkout pilot code
git clone https://github.com/isa2-api4ips/ebsi-pilot.git

# go to ebsi-client project
cd ebsi-pilot/ebsi-client

# execute maven command to build library
mvn clean install
```

Code Block 1 Build domibus

2.2 "EBSI enhanced" Domibus

The "EBSI enhanced" Domibus is a customised version of Domibus created during the pilot project to provide UI features for a Domibus administrator to be able to validate the EBSI timestamp. It must be built from the GitHub repository.

NOTE: The "EBSI enhanced" Domibus was developed and tested with Tomcat 9 and WildFly 20 application servers only.

2.2.1 Build Domibus

To build the "EBSI enhanced" Domibus, first checkout the code and execute the maven command to build all Domibus artefacts.

(The custom version of Domibus is already built and accessible from the address: https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-

snapshots:eu%2Fdomibus%2Fdomibus-distribution%2F4.2.1-EBSI-SNAPSHOThttps://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus)

```
# checkout pilot code (skip this step if the code was already cloned when
building the EBSI Java client library)
git clone https://github.com/isa2-api4ips/ebsi-pilot.git

# go to domibus project
cd ebsi-pilot/domibus

# execute maven command to build domibus
mvn clean install -Ptomcat -Pwildfly -Pdefault-plugins -Pdatabase -
Psample-configuration -Pdistribution -PUI
```

Code Block 2 Build domibus

2.2.2 Deploy Domibus

After the artefacts are built, please use the Quick Start Guide (pdf) or the Domibus Administration Guide from the Domibus web page: Domibus - v4.2.1.

Once Domibus is successfully deployed and running, deploy the EBSI extension covered in the next section.

2.3 Domibus EBSI extension

The Domibus EBSI extension implements both the **Domibus interceptor SPI** and **Domibus IAM SPI** to provide pilot-specific functionality (please refer to the document "Piloting Domibus integration with CEF EBSI (blockchain): Implementation Overview" for details).

2.3.1 Build extension

The extension is built at the same time as Domibus (see above) and is included in the Domibus maven folder structure.

2.3.2 Deploy extension

In order to install the EBSI extension for Domibus, please follow the steps below:

1. Stop the application server.
2. Deploy the EBSI extension using Domibus:

```
# copy extension to the Domibus configuration extension folder
cp ${domibus.code.location}/Domibus-EBSI-extension/target/Domibus-
   EBSI-extension-4.2.1-EBSI-SNAPSHOT.jar
   ${domibus.config.location}/extensions/lib

# copy sample keystores and EBSI configuration properties to extension
   configuration folder
cp ${domibus.code.location}/Domibus-EBSI-extension/src/main/conf/*.*
   ${domibus.config.location}/extensions/conf/
```

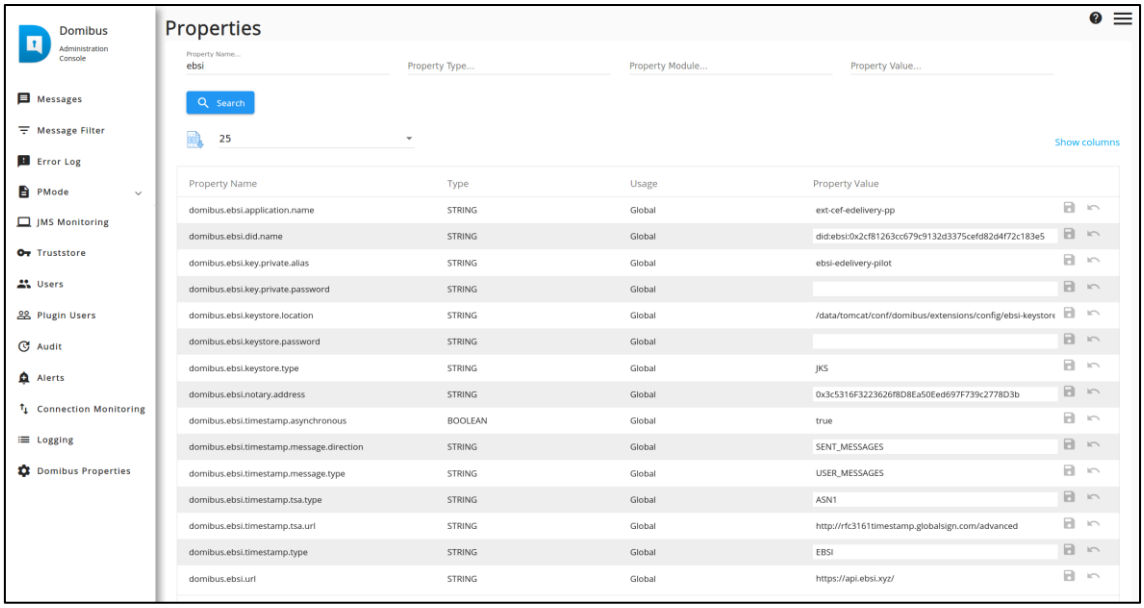
Code Block 3 Deploy extension

3. Start the application server.

For more details, please refer to the Extension cookbook (pdf).

2.3.3 Domibus EBSI extension properties

The Domibus EBSI extension properties can either be set in the Domibus EBSI extension property file or dynamically updated via the Domibus property editor:



Domibus EBSI extension properties:

Property	Description	Values
General properties		
domibus.ebsi.url	URL for accessing the EBSI REST API services.	**Allowed values:** any URL **Default value:** https://api.ebsi.xyz/
domibus.ebsi.application.name	Application name registered in EBSI system for the corresponding private key used for authentication and authorization.	**Allowed values:** Any string **Default value:** ext-cef-edelivery-pp
domibus.ebsi.keystore.location	Keystore with the keys for accessing the EBSI services.	**Allowed values:** Any path **Default value:** ${domibus.config.location}/extensions/config/ebsi-keystore.jks
domibus.ebsi.keystore.type	The type of the used Keystore.	**Allowed values:** JKS, PCKS12 **Default value:** JKS
domibus.ebsi.keystore.password	Keystore password.	**Allowed values:** Any string **Default value:**
domibus.ebsi.key.private.alias	The alias of the private key in the Keystore.	**Allowed values:** Any string **Default value:** ebsi-edelivery-pilot
domibus.ebsi.key.private.password	The password for accessing the private key.	**Allowed values:** Any string **Default value:**
Parameters for "Timestamping of eDelivery AS4 messages using the EBSI notarisation"		

Property	Description	Values
domibus.ebsi.timestamp.type	Extension support EBSI and TSA (rfc3161) timestamp. TSA timestamp was implemented as example of the exiting XMLDSig timestamp. In case of TSA value also the attribute ***domibus.ebsi.timestamp.tsa.url*** must be set.	**Allowed values:** EBSI, TSA **Default value:** EBSI
domibus.ebsi.timestamp.tsa.url	The RFC 3161 timestamp service provider.	**Allowed values:** any URL of the RFC 3161 TSA service provider. **Default value:**
domibus.ebsi.timestamp.message.type	Timestamp ebMS user messages, signal messages or all messages.	**Allowed values:** ALL_MESSAGES, SIGNAL_MESSAGES or USER_MESSAGES **Default value:** ALL_MESSAGES
domibus.ebsi.timestamp.message.direction	Timestamp ebMS sent messages, received messages, all mesages or disable timestamp.	**Allowed values:** DISABLED, SENT_MESSAGES, RECEIVED_MESSAGES OR ALL_MESSAGES **Default value:** ALL_MESSAGES
domibus.ebsi.timestamp.asynchronous	Because the Blockchain timestamp has an impact on the message exchange performance, the timestamp can be performed asynchronously. If the value is set to true, the signature value is extracted, and the timestamp request is submitted asynchronously using the JMS queue.	**Allowed values:** true, false **Default value:** true
domibus.ebsi.notary.address	Blockchain Notary smart contract address.	**Allowed values:** Any string **Default value:** 0x3c5316F3223626f8D8Ea50Eed697F739c2778D3b
Parameters for "Using the EBSI DID Registry for authentication/authorisation in an eDelivery network"		
domibus.ebsi.did.name	DID document name.	**Allowed values:** Any string **Default value:** did:ebsi:0x2cf81263cc679c9132d3375cefd82d4f72c183e5
Domibus extension regististration paramters		
domibus.extension.interceptor.in.classes	Comma-separated list of CXF-interceptor classes for incoming SOAP messages. eDelivery EBSI extension implemented incomming	**Allowed values:** Comma-separated class name list of Interceptors

Property	Description	Values
	message signature timestamp in class: eu.domibus.ebsi.ebms3.interceptor.EBSISignatureTimestampInIntercep tor	**Default value:** eu.domibus.ebsi.ebms3.interceptor.EBSISignatureTimestampInIntercep tor
domibus.extension.interceptor.out.classes	Comma-separated list of CXF-interceptor classes for outgoing SOAP messages. eDelivery EBSI extension implemented outgoing message signature timestamp in class: eu.domibus.ebsi.ebms3.interceptor.EBSISignatureTimestampOutInterceptor	**Allowed values:** Comma-separated class name list of Interceptors **Default value:** eu.domibus.ebsi.ebms3.interceptor.EBSISignatureTimestampOutInterceptor
domibus.extension.iam.authorization.identifier	Identifier of the IAM extension used to verify the authorization of the message's signing certificate. Default Domibus provided authorization implementation value is DEFAULT_AUTHORIZATION_SPI. EBSI_AUTHORIZATION_SPI is authorization implementation provided by the Domibus EBSI extension.	**Allowed values:** Any string **Default value:** EBSI_AUTHORIZATION_SPI

2.4 eDelivery EBSI DEMO application

The eDelivery EBSI Demo application is a simple web application for administering the DID document and for submitting messages to two Domibus Access Points (C2 and C3). The application was built for demonstration purposes.

2.4.1 Build the Demo application

To build the Demo application, first checkout the code and execute the maven command to build the application.

```
# checkout pilot code (skip this step if the code was already cloned when
building Domibus or the EBSI Java client library)
git clone https://github.com/isa2-api4ips/ebsi-pilot.git

# go to eDelivery EBSI DEMO project
cd isa2-api4ips/ebsi-pilot/ebsi-client/example/ebsi-did-webadmin

# execute maven command to build the application
mvn clean install
```

Code Block 4 Build domibus

2.4.2 Deploy the Demo application

In order to deploy the Demo application copy the "war" artefact from the target folder to the deploy folder of the existing tomcat or WildFly server:

```
# example of deployment in tomcat application (set the TOMCAT_HOME
environment variable first to target the tomcat home folder!)
cp ebsi-did-webadmin/target/ebsi-did-webadmin-1.0-SNAPSHOT.war
${TOMCAT_HOME}/webapp/ebsi-webadmin.war

# example of deployment in WildFly application server
cp ebsi-did-webadmin/target/ebsi-did-webadmin-1.0-SNAPSHOT.war
${WILDFLY_HOME}/standalone/deployment/ebsi-webadmin.war
```

Code Block 5 Build domibus

3 Deliverables

3.1 "EBSI enhanced" Domibus

Domibus stores UserMessages and AS4Receipt signal SOAP envelope messages.

Option to download AS4Receipt (Used for integration testing)

Domibus message table has EBSI Timestamp action buttons:

- Validate Timestamp
- Download SOAP Envelope
- Download action message button contains full SOAP Envelope and payloads + Timestamp metadata

GIT URL: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus

Artefact repository: https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-snapshots:eu%2Fdomibus%2Fdomibus-distribution%2F4.2.1-EBSI-SNAPSHOT

3.2 Domibus interceptor SPI

Domibus SPI for adding custom apache-cxf interceptors.

Documentation: Domibus Interceptor SPI

GIT URL: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus/Domibus-interceptor-spi

Module: Domibus-interceptor-spi

Artefact repository: https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-snapshots:eu%2Fdomibus%2Fdomibus-interceptor-spi%2F4.2.1-EBSI-SNAPSHOT

3.3 Domibus EBSI interceptor SPI implementation

Domibus interceptor SPI implementation for enhancing ebMS 3.0 signatures with EBSI (and TSA) timestamps

GIT URL: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus/Domibus-EBSI-extension
Package: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus/Domibus-EBSI-extension/src/main/java/eu/domibus/ebsi/ebms3/interceptor

Module: Domibus-EBSI-extension

Artefact repository: https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-snapshots:eu%2Fdomibus%2FDomibus-EBSI-extension%2F4.2.1-EBSI-SNAPSHOT

3.4 eDelivery EBSI IAM SPI implementation

Domibus IAM SPI implementation for authorization with DID document

GIT URL: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus/Domibus-EBSI-extension
Package: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/domibus/Domibus-EBSI-extension/src/main/java/eu/domibus/ebsi/authentication

Module: Domibus-EBSI-extension

Artefact repository: https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-snapshots:eu%2Fdomibus%2FDomibus-EBSI-extension%2F4.2.1-EBSI-SNAPSHOT

3.5 eDelivery EBSI library

EBSI library for invoking EBSI smart contact and REST services for supporting the implementation of the use cases

GIT URL: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/ebsi-client

Artefact repository: : https://ec.europa.eu/digital-building-blocks/artifact/#browse/browse:eDelivery-snapshots:eu%2Feuropa%2Fec%2Fedelivery%2Febsi%2Febsi-core%2F1.0-SNAPSHOT

3.6 eDelivery EBSI DEMO application

Web application for managing DID document and submitting messages to Domibus C2 and C3

GIT URL: https://github.com/isa2-api4ips/ebsi-pilot/tree/main/ebsi-client/example/ebsi-did-webadmin

Module: examples/ebsi-did-webadmin

Artefact repository: https://ec.europa.eu/digital-building-blocks/artifact/service/rest/v1/search/assets/download?repository=eDelivery-snapshots&group=eu.europa.ec.edelivery.ebsi&name=ebsi-did-webadmin&maven.baseVersion=1.0-SNAPSHOT&maven.extension=war

4 Contact information

For any questions regarding this document please contact EC-EDELIVERY-SUPPORT@ec.europa.eu.