eDelivery

New features in Domibus 5.0

Webinar

Contact: EC-DIGITAL-BUILDING-BLOCKS@ec.europa.eu

18 October 2022

Welcome to “New features in Domibus 5.0” webinar

Monika KOKSTAITE

Stakeholder Management and Communication Office

Speakers of the “New features in Domibus 5.0” webinar

Bogdan Dumitriu

Project Officer

Project officer in charge of eDelivery implementation in DIGIT

Cosmin Baciu

External Consultant

Chief Architect of the eDelivery Building Block

Monika Kokstaite

External Consultant

Stakeholder management and communication for eDelivery

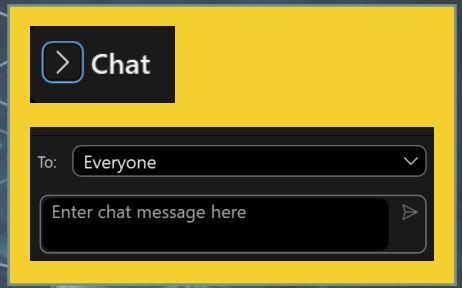
Introduction and house rules

Please note that this event is recorded.

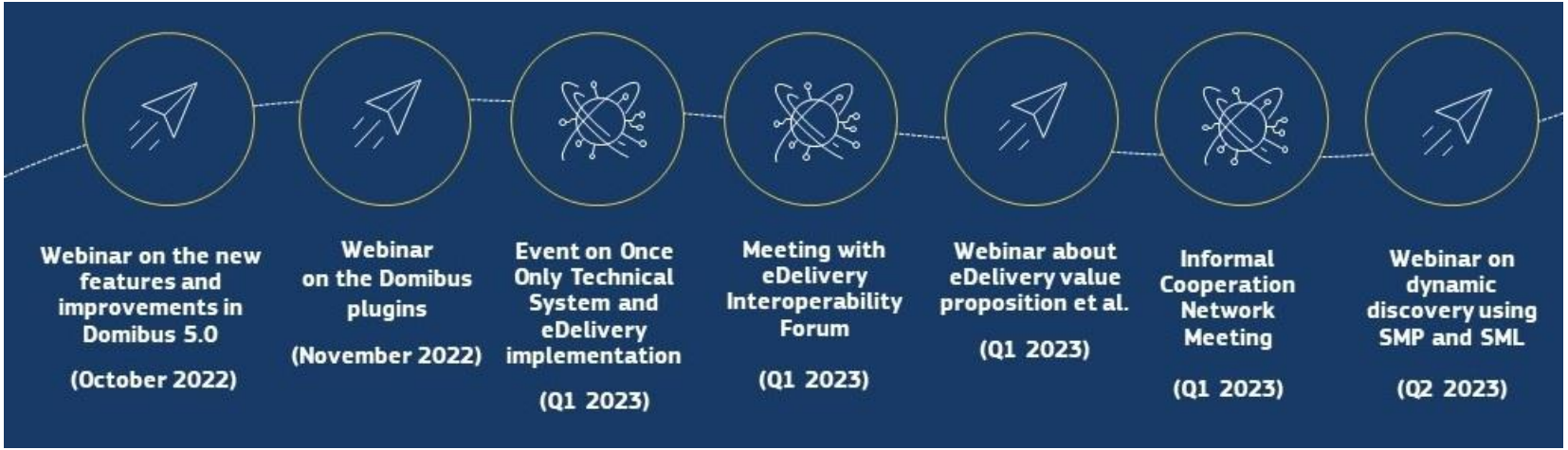
Please ask your questions in the chat section on the right side of the screen.

Raise your hand if you want to speak or ask a question.

Mute your microphone.



eDelivery updates



DIGIT
@EU_DIGIT

EU Digital Identity
@EUdigitalID

Agenda

The "New features in Domibus 5.0" webinar will guide the users through the new features of Domibus.

- **14h00 – 14h05 | Welcome**
- **14h05 – 14h15 | Introduction to eDelivery**
- **14h15 – 15h05 | New features in Domibus 5.0**
- **15h05 – 15h30 | Q&A and sharing of experiences**

Introduction to eDelivery

Bogdan DUMITRIU

Digital Europe Building Blocks

A **Building Block** is an open and reusable digital solution.

What?

It can take the shape of **frameworks, standards, software products or software as a service (SaaS)**, or any combination thereof.

How?

It promotes the adoption of the same **open standards and technical specifications** by the **different sectors** of the Union for the most basic & common functionalities of their projects or platforms.

Why?

Building Blocks enable interoperability across borders and sectors.

eDelivery

Exchange online data and documents reliably and securely.

eInvoicing

Promote the implementation of the European standard for electronic invoicing across borders.

eSignature

Create and verify electronic signatures between businesses and EU citizens.

eID

Allow citizens to prove who they are across borders, making it easier to access online services in another EU Member State.

and more...

How to use a Building Block?

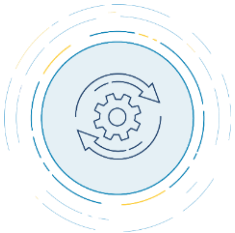
There are 3 options: **buy**, **build** or **reuse** and you can always **co-develop** your solution with other parties.

Buy

Buy a compliant, interoperable solution from the market.

Reuse

Reuse sample software available through Digital Europe.



Build

Build an EU-compliant solution from scratch based on Building Block standards.

Whatever you choose, **the relevant Digital Europe team will support you** in implementing the Building Block into your project.

Building Block adoption throughout the years & sectors

- **338** projects reusing a Building Block.
- **53** projects committed to reuse.
- **88** projects committed to analyse.
- **150%** increase in reuse since 2018.

What is eDelivery?

eDelivery

Exchange data and documents online reliably and securely.

Interoperability

eDelivery enables the exchange of documents and data among heterogeneous information systems using a standardized protocol, thereby laying the foundation for cross-domain and cross-project interoperability.

Scalability and performance

eDelivery solutions ensure sustainable levels of performance and maintainability even as the number of participants and/or messages in a network grows.

Security and accountability

eDelivery ensures data integrity and confidentiality in every transmission through the use of digital signatures and encryption. eDelivery also guarantees legal assurance and accountability by mandating that the recipient of a message must send a digitally signed acknowledgement of receipt for every message received.

Vendor and platform agnostic

Because this is a vendor and platform neutral solution, its specifications are not proprietary or controlled by one vendor alone. Also, eDelivery is available in multiple products and solutions from different vendors you can choose from.

Paper delivery

Write the message

Look for a destination address

Put the message in an envelope

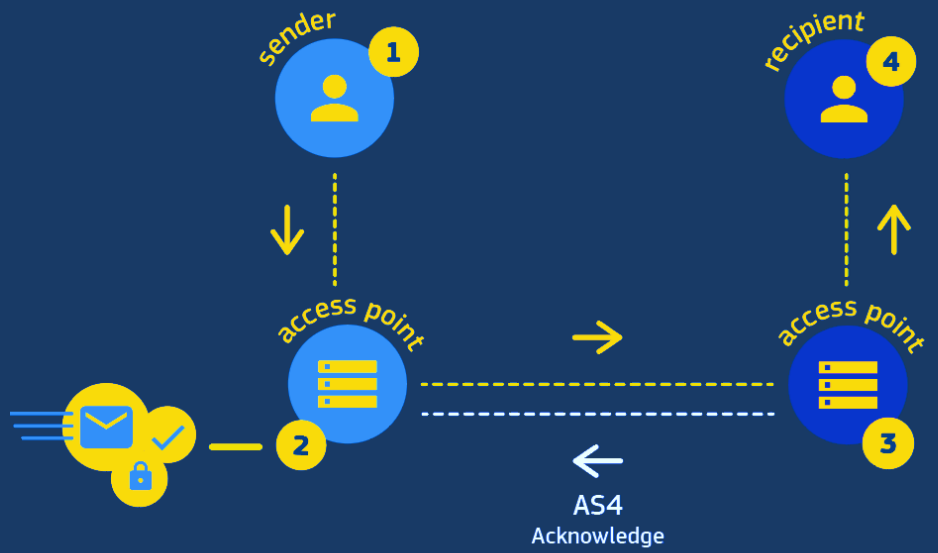
Post the letter

Transport the message to its destination

Deliver the message

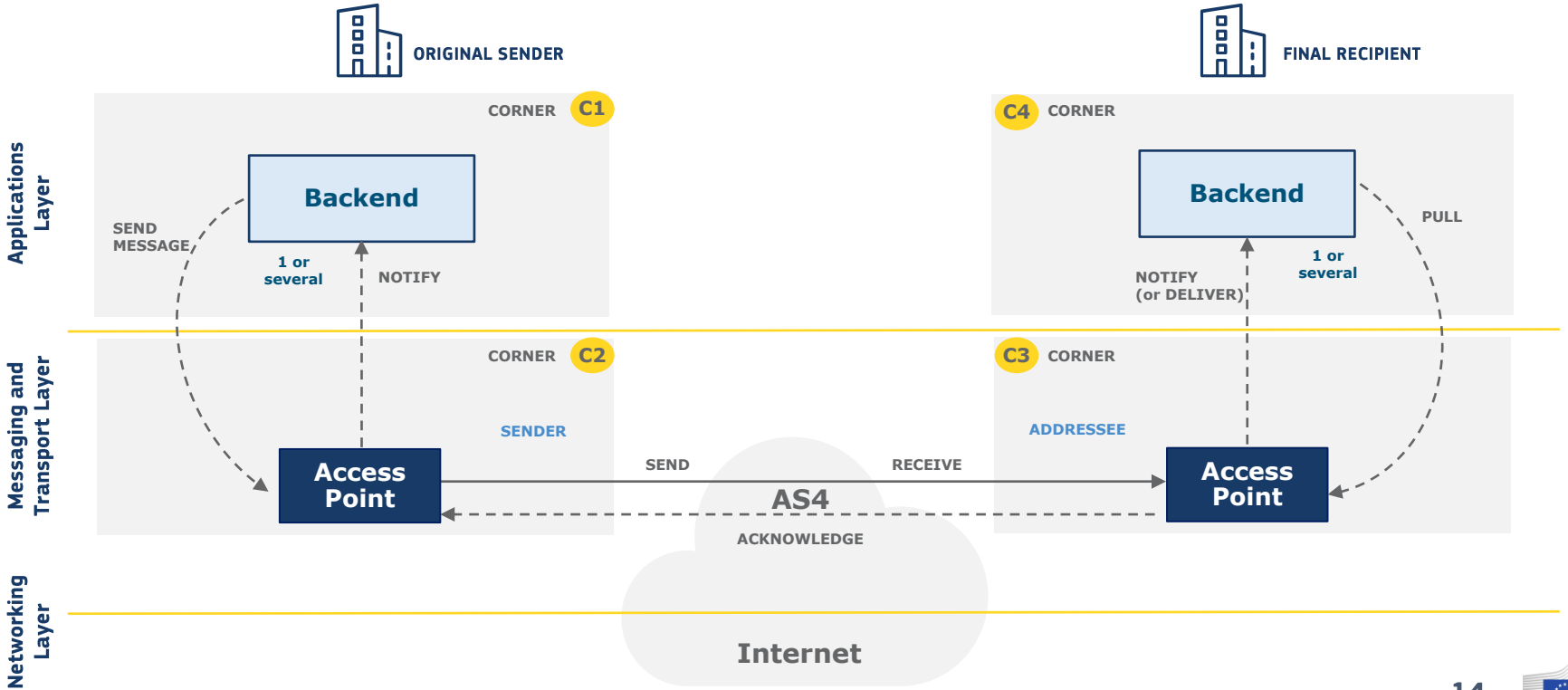
eDelivery

1. **Submit**
   Sender sends message to sending AP
2. **Send**
   Sending AP processes message
   a) Validation and compression of the user message;
   b) Signing of the compressed message;
   c) Encryption of the signed compressed message.
3. **Receive**
   Receiving AP processes message
   a) Receives and decrypts the encrypted message;
   b) Verifies the sender's signature;
   c) Decompresses the decrypted message;
   d) Validates the original user message;
   e) Sends the acknowledgement to the sending AP;
   f) Stores the user message for download.
4. **Deliver**
   Recipient receives message from receiving AP



eDelivery Four-Corner Model

Static discovery



eDelivery Service offering

SOFTWARE	OPERATIONS SERVICES			STAKEHOLDER MANAGEMENT SERVICES	
Sample software maintained by the EC (with documentation)	**Managed services**	**Testing services**	**Supporting services**	**Onboarding services (for stakeholders)**	**Community management services**
Access Point (AP)	Public Key Infrastructure (PKI)	Connectivity testing	Training & Deployment	Self-assessment tool	Developers Community
Service Metadata Publisher (SMP)	Service Metadata Locator (SML)	Conformance testing	Service Desk	Onboarding of new stakeholders	Market guide
Service Metadata Locator (SML)				Cost estimation tool	

TECHNICAL SPECIFICATIONS					
Access point specifications	SMP specifications	SML specifications	Security control guidance	Trust models guidance	Guidance on digital certificates

STANDARDS OF SOs
Standards monitoring

Service offering Description (SoD)

All services are described in an SoD describing its purpose, the users for which it is for, its benefits and the process to obtain it

Service Level Arrangements (SLA)

Documents that describe Service Level Targets to be reached when delivering Building Block Services.

eLearning, videos, success stories

Some services feature multimedia such as eLearnings, instructional videos or success stories to help grasp what the service is about

Digital Europe platform

eDelivery service offering, and more about the building block, can be found online

Digital Europe

eDelivery AS4 conformant solutions

eDelivery AS4 conformant solutions

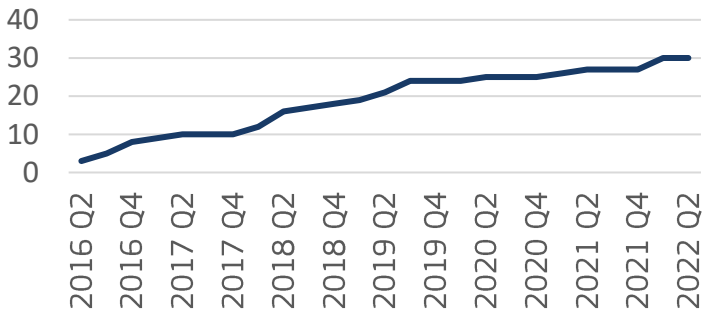
This page lists the solutions that have passed the conformance testing according to the eDelivery AS4 profile:

- Axway
- B2BRouter
- Babelway
- Bizbrains
- CData Arc
- Cleo Integration Cloud
- Data Interchange
- DCS EIP
- Domibus (EC sample implementation)
- Edicom ASx Server
- eefacta Server
- EESSI AS4.NET
- Flame
- Galaxy Gateway
- Harmony eDelivery Access
- Holodeck B2B
- IBM
- Ida Infront iipax com
- iFenix
- ion-AP
- Laurentius
- Mendelson
- Navitasoft - IP Systems AS4-IP
- Nota
- OXALIS
- Pagero
- phase4
- Qvalia
- SEEBURGER
- Software AG
- ViaAdValvas Gateway

More information on Digital Europe

Conformant Solutions >

Evolution of AS4 conformant solutions (2016-2022)



Product updates: Domibus 5.0

- Full database redesign
- Major performance improvements:
  - From 300 to >1,000 messages / second
  - Avoid performance degradation as millions of messages accumulate in the database
- Support for using an external agent to archive Domibus messages (eArchiving BB)
- Web service plugin: support for “push” mode in addition to “pull” mode
- Further technical improvements
- Maintenance of supported releases

Domibus 5.0: supported platforms

- Application servers:
  - WildFly 26.1.x
  - WebLogic 12.2.1.4 (tested version, future versions might work)
  - Apache Tomcat 9.0.x
- Database:
  - MySQL 8 (future versions might work)
  - Oracle 12c R2 and Oracle 19c
- Java 8 features / compile with Oracle JDK 8: tested to run correctly with:
  - Oracle JDK 8/ WebLogic
  - OpenJDK 11/ WildFly + Tomcat (tested with AdoptOpenJDK 11 version 11.0.9.1+1)


New features in Domibus 5.0
Cosmin BACIU

5.0 New Features and Improvements

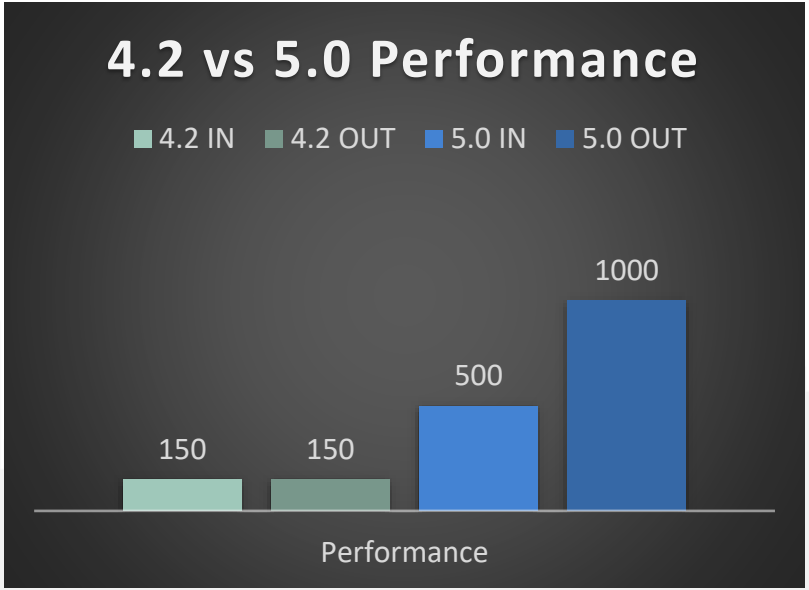
- Improved overall performance
  - Refactoring of the database to allow partitioning and improve performance
  - Optimized pull locking for Oracle
  - UI Replication has been removed
- UI Domibus Admin Console improvements
  - New page in Admin Console to manage the TLS truststore
  - Possibility to visualize the non-repudiation receipts in the Admin console
  - Possibility to add and remove a domain at runtime when using multitenancy

5.0 New Features and Improvements

- New WS Plugin
  - The WSDL/XSD schema is using now the ‘edelivery’ namespace
  - Possibility to operate the plugin in PUSH mode
  - Possibility to filter messages retrieved by the ‘listPendingMessages’ method
- Possibility to use an external agent to archive Domibus messages
- Possibility to validate incoming user messages using a Validation Extension
- Possibility to reference JMS Plugin payloads via HTTP endpoints
- Possibility to offload the TLS traffic to an external component

Improved overall performance

- 5x improvement in Domibus 5.0 performance compared to Domibus 4.2
- Performance tested using a cluster with 4 Domibus instances:
  - Domibus 4.2: 300 messages/s in parallel
    - 150 messages/s received
    - 150 messages/s sent
  - Domibus 5.0: 1500 messages/s in parallel
    - 500 messages/s received
    - 1000 messages/s sent



Improved overall performance

Refactoring of the database to allow partitioning and improve performance

- What is database partitioning?

Partitioning allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these database objects to be managed and accessed at a finer level of granularity.

- Domibus has to be configured to use partitions on an Oracle database (but not on a MySQL one)
- Domibus partitions are created based on the format of the primary key with a granularity of

Domibus 4.2.x	Domibus 5.0
• The primary key is a simple number and it doesn't contain any embeded metadata • Example of a primary key: 1	• The primary has a specific format yyMMddHHDDDDDDDDDD • Example of a primary key: 2210050700000001000 where • 221005: date • 07: hour • 0000001000: number of 10 digits

Improved overall performance

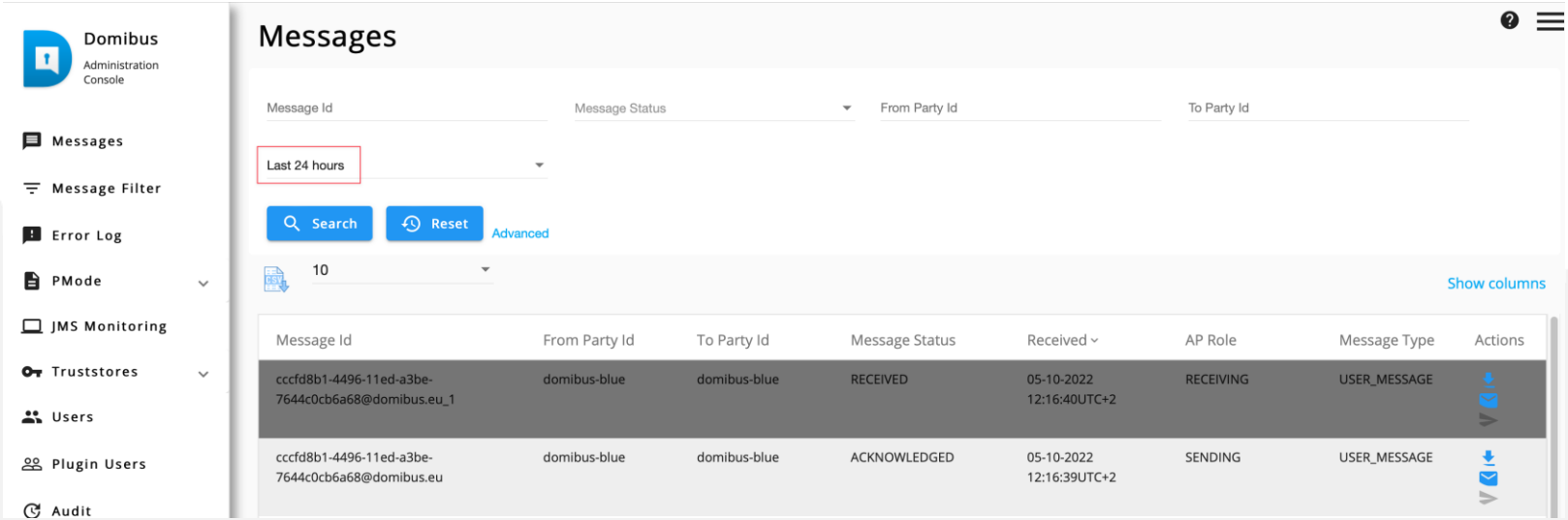
Refactoring of the database to allow partitioning and improve performance

- To allow database partitioning by primary key we had to link all database tables in which the AS4 messages are stored using foreign keys
- Benefits of database partitioning:
  - Faster search in the application and in the Admin Console by taking advantage of the primary key format and partitions. The old UI replication mechanism is not needed anymore and it was removed
  - Faster deleting of messages by the retention mechanism: partitions are dropped almost instantaneously
  - No degradation of performance over time, even with millions of messages in the database

Improved overall performance

Faster response times in the Domibus Admin Console

- Domibus displays by default only messages sent and received in the last 24 hours to speed up the loading time of the messages page
- Can be adapted in domibus.properties



Improved overall performance

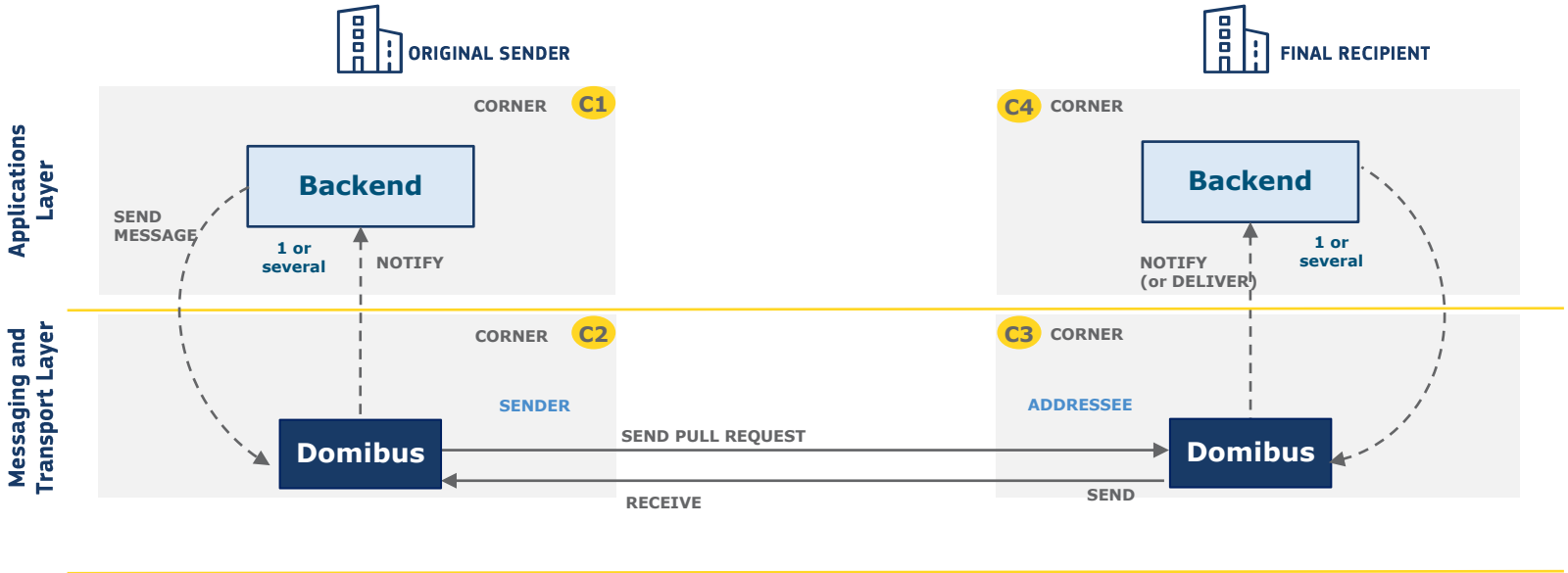
Other technical improvements

- Removed XA transactions
- Code changes to minimize the time spent in transactions
- Use second level cache for Hibernate
- Use caching for database sequence
- Possibility to use asynchronous logging

Improved overall performance

Optimized pull performance for Oracle

- In a classical scenario, C2 sends (pushes) the messages to C3
- When using the pull mechanism, C2 gets (pulls) the messages from C3



Improved overall performance

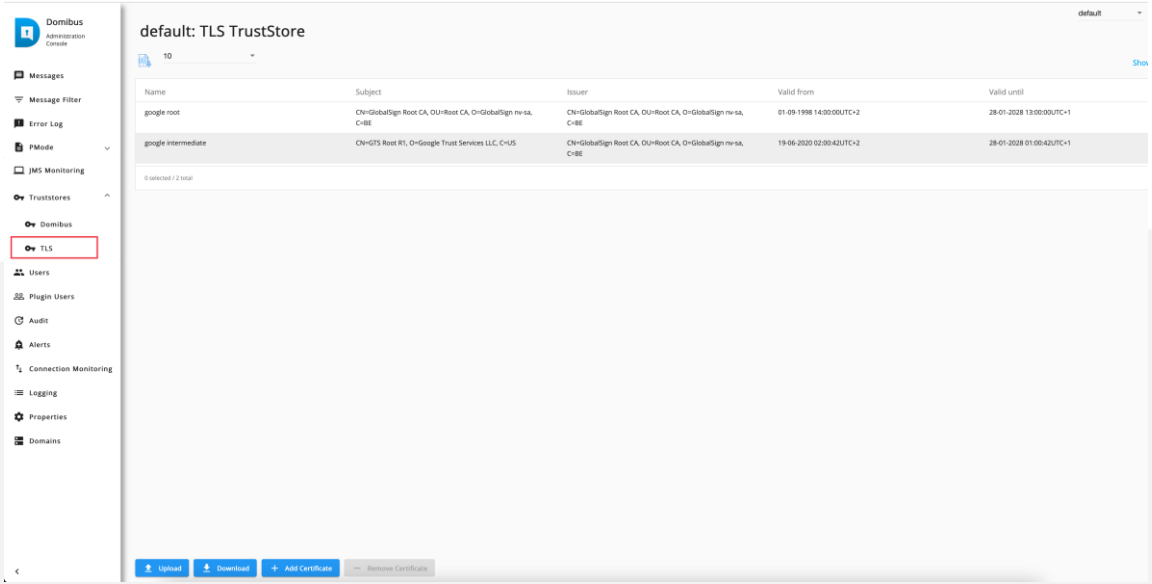
Optimized pull performance for Oracle

- In Domibus 4.2.x, the locking mechanism used at C3 side for pull mechanism had poor performance due to unoptimized database locking
- Why is locking need?
  - Multiple pull requests are received by C3 in parallel and C3 needs to avoid sending the same message twice to C2
- Domibus 5.0 improves the locking mechanism giving a better performance for the pulling mechanism

UI Domibus Admin Console improvements

New page in Admin Console to manage the TLS truststore

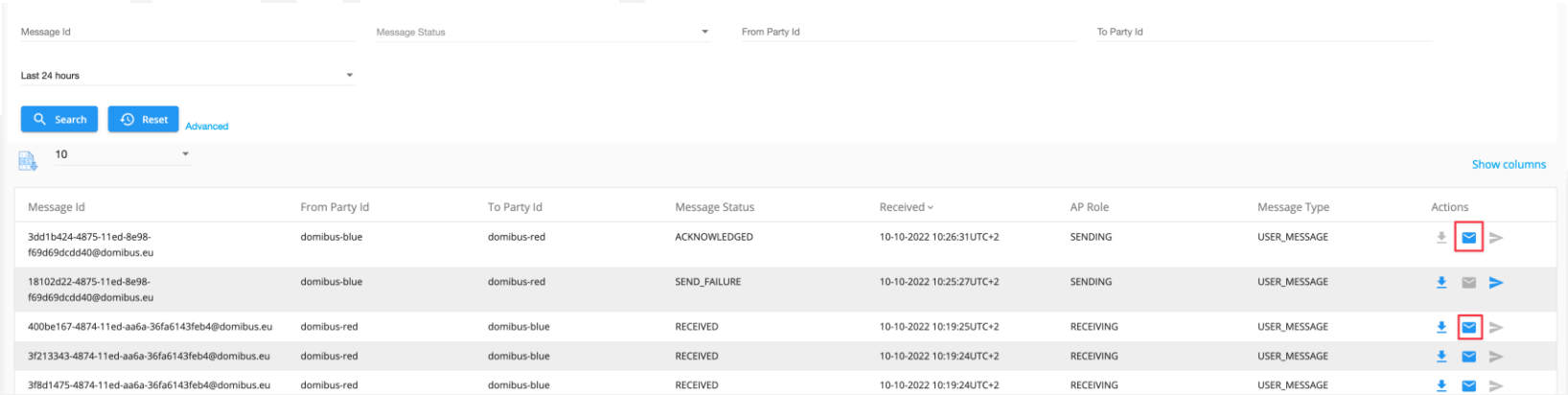
- Live demo



UI Domibus Admin Console improvements

Possibility to visualize the non-repudiation of origin and receiptin the Admin console

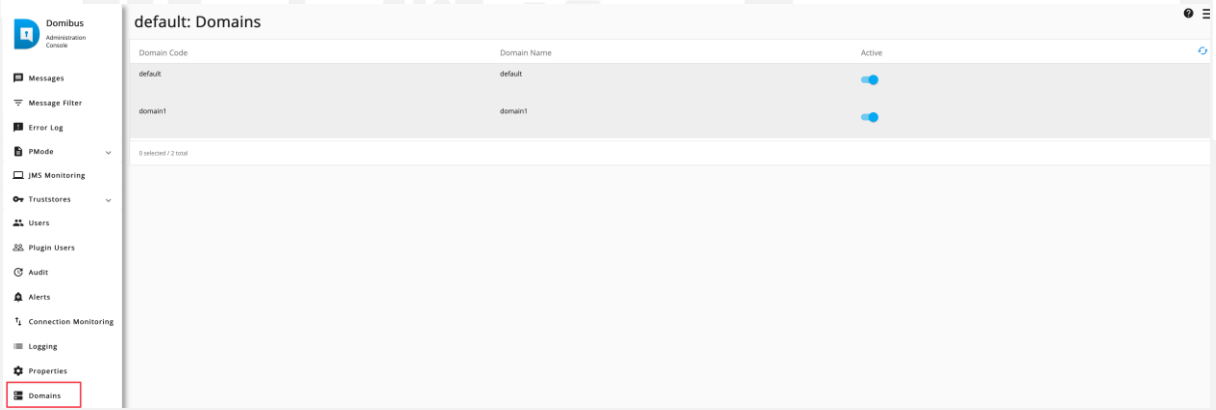
- Sender and receiver cannot deny that they have sent or received a message
- Domibus stores the non-repudiation of origin and receipt
- Can be used to resolve any dispute between parties
- Live demo



UI Domibus Admin Console improvements

Possibility to add and remove a domain at runtime when using multitenancy

- Domibus supports multitenancy
  - No need to deploy multiple instances of Domibus
  - Each tenant has its own set of configuration: database schema, keystore, truststore, Pmode, properties.
- Not needed anymore to stop the server when adding/removing a new domain
- Can be done live while the application is running, increasing the availability of Domibus
- Live demo



WS Plugin

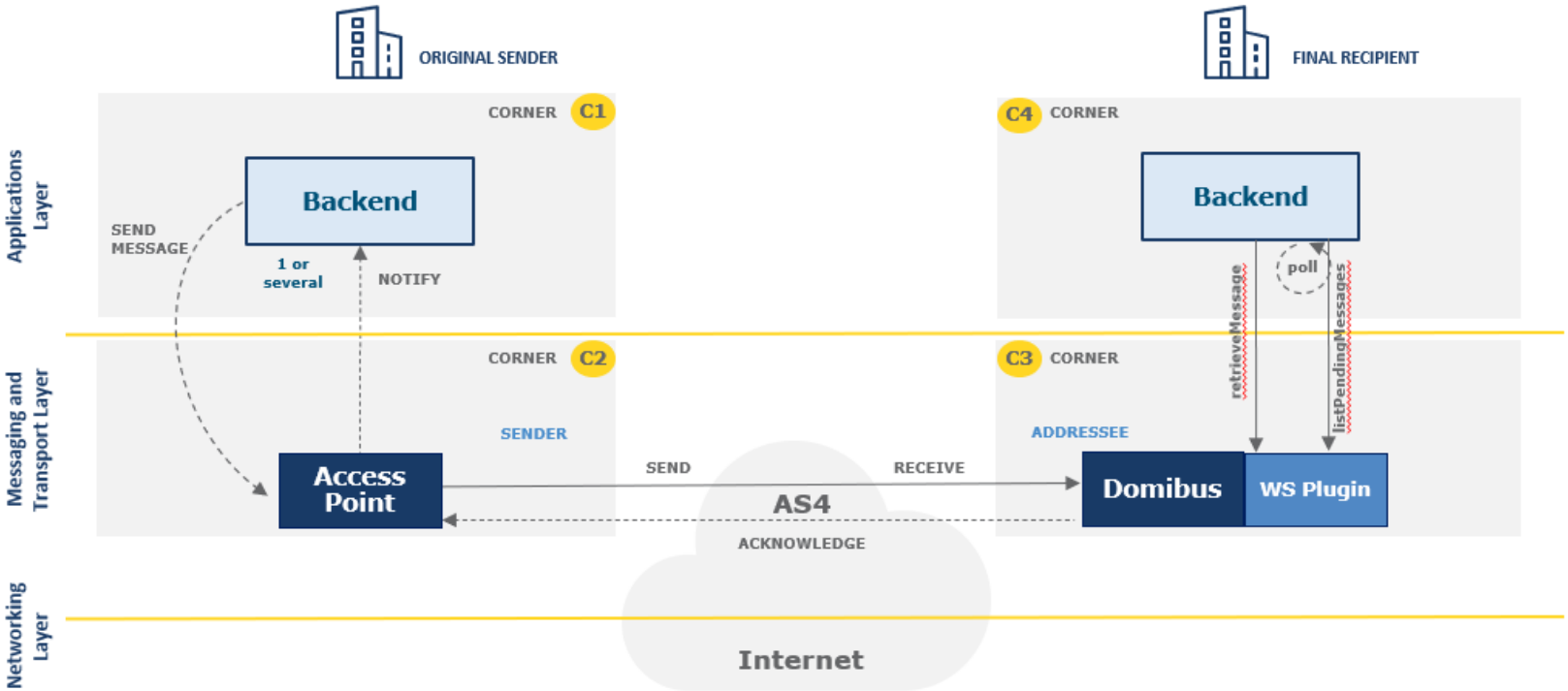
New WS Plugin

- New WS Plugin added in Domibus 5.0
- Old WS Plugin was retained to avoid breaking backward compatibility
- Old WS Plugin was deprecated and it will be removed in one of the next Domibus major versions
- We recommend that you plan the migration from the old plugin to the new plugin

	Old WS Plugin	New WS Plugin
Endpoint	/domibus/services/backend	/domibus/services/wsplugin
WSDL namespace	http://org.ecodex.backend/1_1	http://eu.domibus.wsplugin
Integration	Pull	Pull and push

WS Plugin

PULL mode



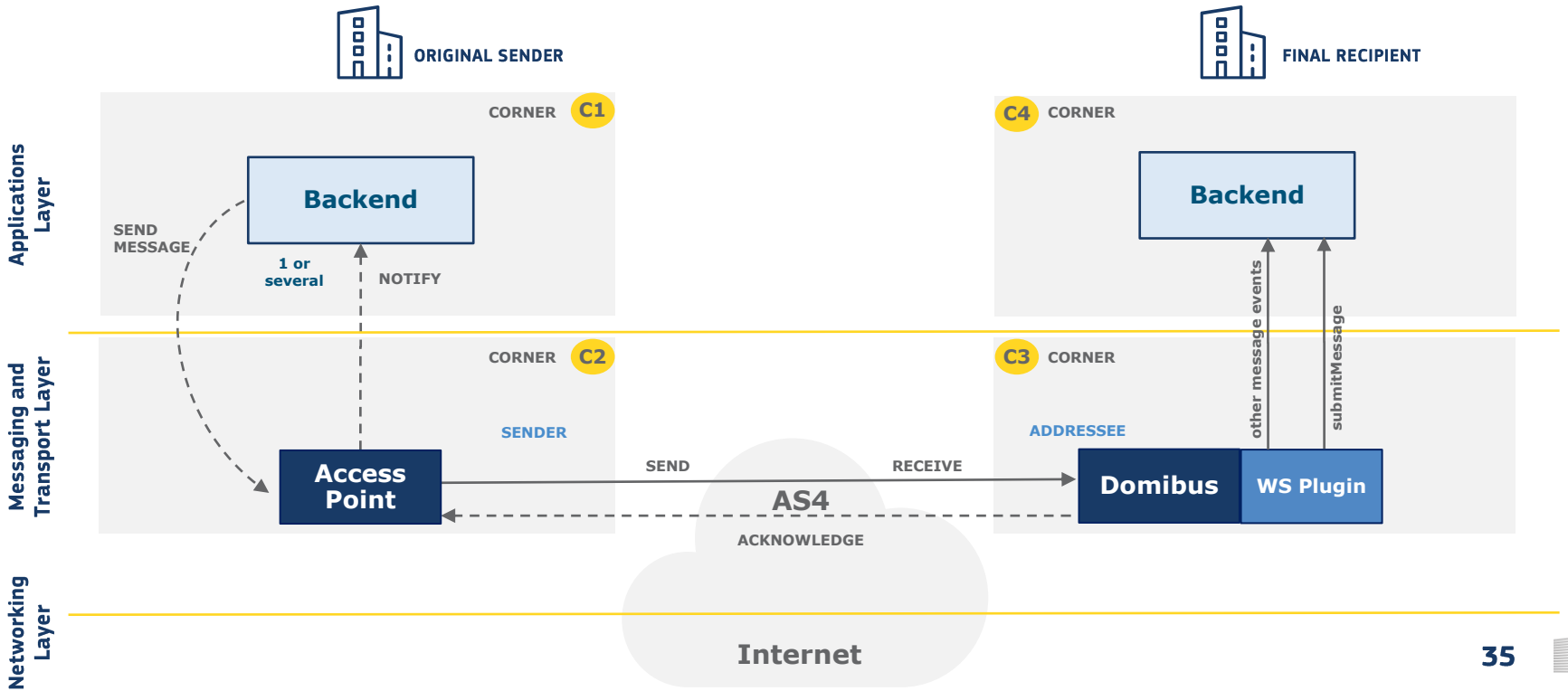
WS Plugin

PULL mode

- Pull mechanism can be useful when the backend cannot implement a SOAP Web Service or when it cannot allow incoming HTTP traffic
- Backend user message retrieval flow:
  - Needs to call/poll *listPendingMessages* operation regularly to check for incoming messages
  - Obtains the IDs of the incoming messages
  - Calls *retrieveMessage* for each message ID
- Backend can check on demand the status of a specific message by calling the *getStatus* operation

WS Plugin

PUSH mode



WS Plugin

PUSH mode

- Backend is not required to poll regularly for incoming messages
- Backend must implement a SOAP Web Service WSDL for push mechanism and must allow incoming HTTP traffic
- Domibus pushes the message to the backend once it receives a message from C2
- Same push mechanism is used for other message events, Domibus informs the backend using the following operations:
  - *sendSuccess: when it sends successfully a UserMessage to C3*
  - *sendFailure: when it fails to send a UserMessage to C3*
  - *submitMessage: used to send the message to the backend*
  - *receiveSuccess: when it receives successfully a UserMessage from C2*
  - *receiveFailure: it fails to receive a UserMessage from C2*
  - *messageStatusChange: when the status of a UserMessage changes*
  - *delete: when a UserMessage is deleted*
  - *deleteBatch: when a batch of UserMessages is deleted*

WS Plugin

List pending messages

- The old WS Plugin *listPendingMessages* does not support filtering the messages, the list of all received messages is returned
- The new WS Plugin *listPendingMessages* operation supports message filtering. Available filters:
  - *messageId: the ID of the received UserMessage*
  - *conversationId: the conversation ID of the received UserMessage*
  - *refToMessageId: the refToMessageId ID of the received UserMessage*
  - *fromPartyId: the party ID of the C2 Access Point*
  - *originalSender: the originalSender value from the received UserMessage*
  - *finalRecipient: the finalRecipient value from the received UserMessage*
  - *receivedFrom: a date representing the start of the interval from which we search for received UserMessages*
  - *receivedTo: a date representing the end of the interval from which we search for received UserMessages*

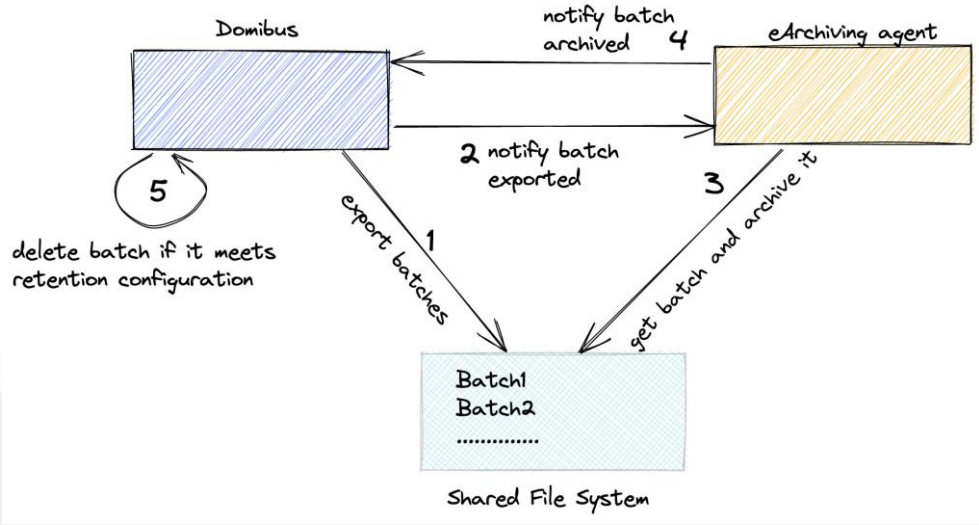
WS Plugin

Live Demo

eArchiving

Overview

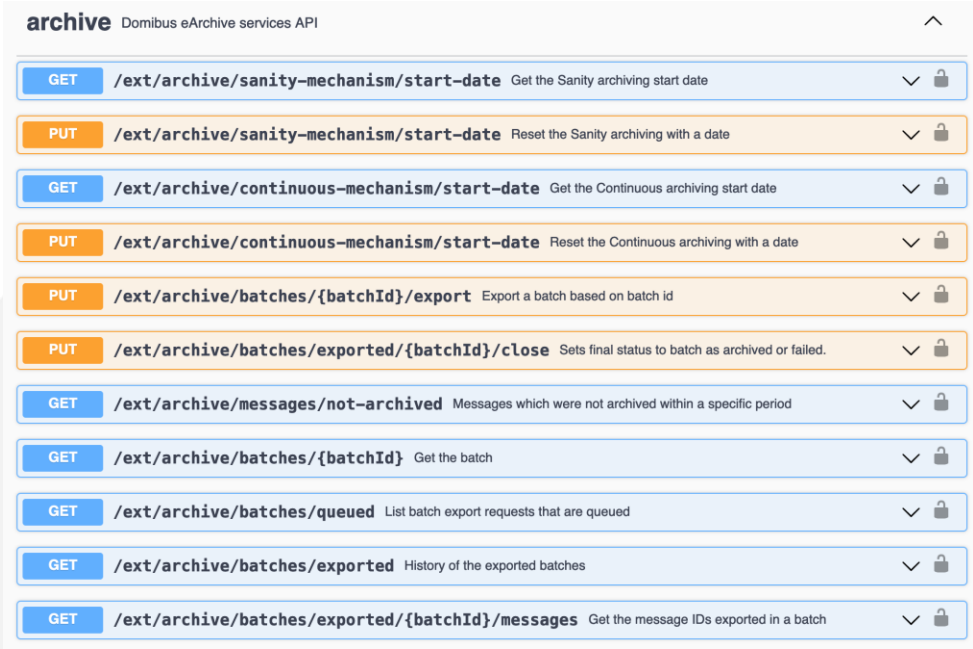
- eArchiving gives the possibility to export the exchanged user messages from Domibus and integrate with an eArchiving client using REST API
- Domibus continuously exports the data to be archived in batches in a shared file system
- eArchiving agent reads each batch, archives it and notifies Domibus



eArchiving

Domibus eArchiving REST API

- Can be found here: https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Domibus+v5.0.1+REST+services+documentation



eArchiving

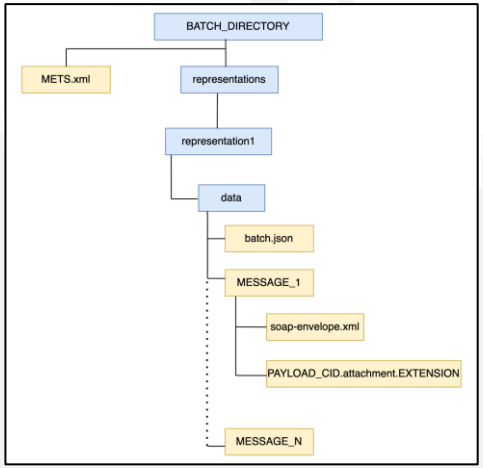
eArchiving agent REST API

- An eArchiving agent must implement the following REST API to integrate with the eArchiving mechanism of Domibus:
- https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Domibus+v5.0.1+eArchiving+REST+services+documentation

eArchiving

eArchiving exported data format

- Domibus exports only messages that are not yet archived and that are in a final state: RECEIVED, DELETED, DOWNLOADED or ACKNOWLEDGED. Payloads are decompressed before being exported
- Messages are exported in batches
- The structure of the batch is using EARK SIP format as illustrated below



BATCH_DIRECTORY

- The directory in which Domibus exports messages contained in the batch.
- This directory is named based on a UUID.

eArchiving

eArchiving exported data format – METS.xml

Contains:

- A list of all the exported files and their checksum
- For all the exported messages
  - The batch id

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<mets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.loc.gov/METS/"
      xmlns:xlink="http://www.w3.org/1999/xlink" OBJID="a46ab3d0-c710-4d73-b58d-e93e30b53a80" TYPE="ERMS"
      xlink:CONTENTTYPESPECIFICATION="SMURFERMS" PROFILE="http://www.dasboard.eu/specifications/sip/v03/METS.xml"
      xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd" LABEL="root level METS file for an IP">
   <metsHdr CREATEDATE="2017-01-31T13:07:22.6970809+02:00" RECORDSTATUS="NEW"
            LASTMODDATE="2017-01-31T13:07:22.6970809+02:00">
      <agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
         <name>Domibus</name>
      </agent>
      <!-- Batch Id -->
      <metsDocumentID>a46ab3d0-c710-4d73-b58d-e93e30b53a80</metsDocumentID>
   </metsHdr>
   <fileSec ID="e2a80c69-1eb4-4c2a-a029-5d77bf53d325">
      <fileGrp ID="e4d9422e-4c26-493b-929b-580198c34055" USE="Files root">
         <fileGrp ID="b6f4b954-9bd4-4179-93d3-37732b90923d" USE="data">
            <file ID="fec0430c-9152-4662-86fe-f9e78dad9baf" MIMETYPE="application/ octet-stream"
                  CREATED="2017-01-31T13:07:22.7470810+02:00"

>
               <FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="file:data/batch.json"/>
            </file>
         </fileGrp>
         <!-- message 1 files -->
         <!-- The message 1 files are grouped in a fileGrp with an ID that uses the AS4 UserMessage ID -->
            <file ID="bec0430c-9152-4662-86fe-f9e78dad9baf" MIMETYPE="text/xml" SIZE="11717"
                  CREATED="2017-01-31T13:07:22.7470810+02:00"
                  CHECKSUM="0d71382407d6a13af515761a6e1abd0e8b0784dab73e1a52427aaa9dbc4f73a9"
                  CHECKSUMTYPE="SHA-256">
               <FLocat LOCTYPE="URL" xlink:type="simple"
                       xlink:href="file:data/9a0c6088-70ac-43b1-ab57-2f9d1f0204b7/soap-envelope.xml"/>
            </file>
            <file ID="bec0430c-9152-4662-86fe-f9e78dad9baf" MIMETYPE="text/xml" SIZE="12717"
                  CREATED="2017-01-31T13:07:22.7470810+02:00"
                  CHECKSUM="1d71382407d6a13af515761a6e1abd0e8b0784dab73e1a52427aaa9dbc4f73a9"
                  CHECKSUMTYPE="SHA-256">
               <FLocat LOCTYPE="URL" xlink:type="simple"
                       xlink:href="file:data/9a0c6088-70ac-43b1-ab57-2f9d1f0204b7/message.attachment.xml"/>
            </file>
            <file ID="cec0430c-9152-4662-86fe-f9e78dad9baf" MIMETYPE="application/pdf" SIZE="42717"
                  CREATED="2017-01-31T13:07:22.7470810+02:00"
                  CHECKSUM="2d71382407d6a13af515761a6e1abd0e8b0784dab73e1a52427aaa9dbc4f73a9"
                  CHECKSUMTYPE="SHA-256">
               <FLocat LOCTYPE="URL" xlink:type="simple"
                       xlink:href="file:data/9a0c6088-70ac-43b1-ab57-2f9d1f0204b7/invoice.attachment.pdf"/>
            </file>
         </fileGrp>
         <!-- .........-->

         <!-- message n files -->
         <fileGrp ID="0a0c6088-70ac-43b1-ab57-2f9d1f0204b7" USE="data">
            <!-- message n files -->
         </fileGrp>
      </fileGrp>
   </fileSec>
</mets>
```

METS.xml example

eArchiving

eArchiving exported data format – batch.json

A JSON file containing metadata related to the batch such as:

- The version of data format exported
- Batch id
- Request type: continuous or manual
- The status of the batch export: success
- Error code of the error in case of failure.
- Error description of the error in case of failure
- A timestamp of the batch export
- The time period of the messages included in the batch: message start date and message end date.
- A checksum of the batch manifest METS.xml file
- The list of exported message ids


```
{
  "version" : "1",  ````
  "batchIid": "9a0c6088-70ac-43b1-ab57-2f9d1f0204b7",
  "requestTtype": "continuous",
  "status": "failed",
  "errorCode": "DOM_001",
  "errorDescription": "Failed to export batch",
  "timestamp": "2021-06-25T12:00:00Z",
  "messageStartId": "2021-01-25T12:00:00Z",
  "messageEndId": "2021-01-26T12:00:00Z",
  "manifestChecksum":
"sha256:01ba4719c80b6fe911b091a7c05124b64eeece964e09c058ef8f9805daca546b",
  "messages": [
    "123c6088-70ac-43b1-ab57-2f9d1f0204b7",
    "567c6088-70ac-43b1-ab57-2f9d1f0204b7"
  ]
}
```


Batch.json example

eArchiving

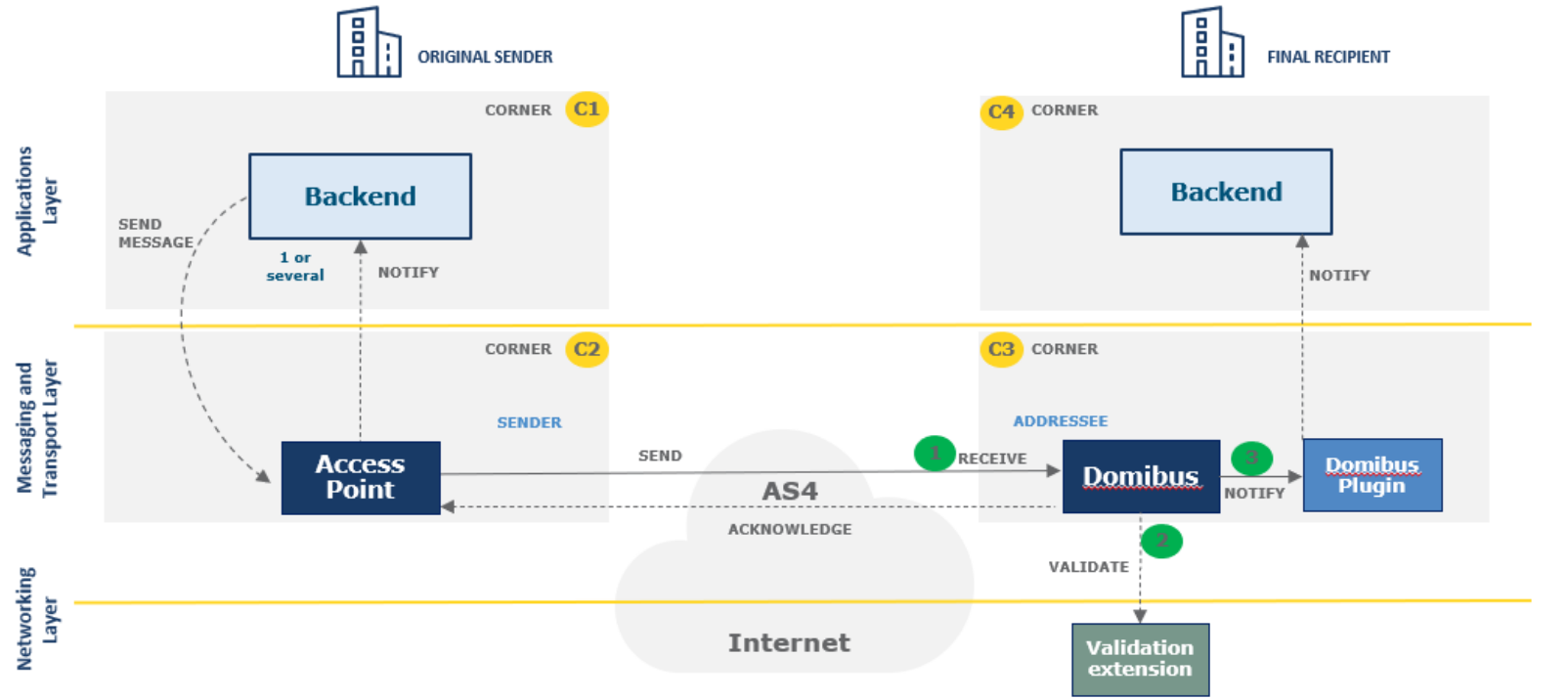
eArchiving exported data format – batch.json

- Live demo

Validation Extension

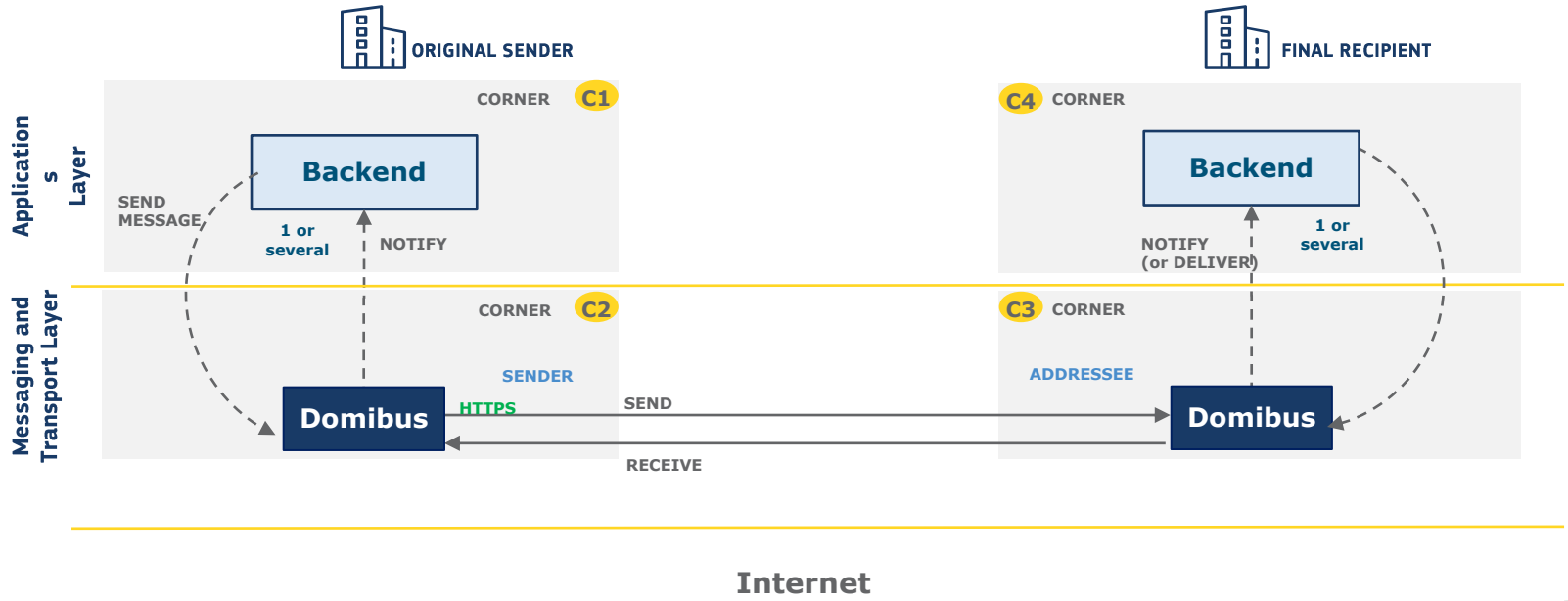
- The Validation Extension gives the possibility to centrally validate the incoming user message before it is delivered to the plugins
- A validation extension implementation must implement the SPI defined in the domibus-MSH-spi module
- Deploy the extension by copying the validation extension a jar into ${domibus.config.location}/extensions/lib
- Scenarios in which the validation extension can be used:
  - Antivirus scan can be plugged into the validation extension to validate the incoming user message payloads
  - Audit
  - Centrally enforce user message payload validation; it executes before the user message reaches a Domibus plugin

Validation Extension



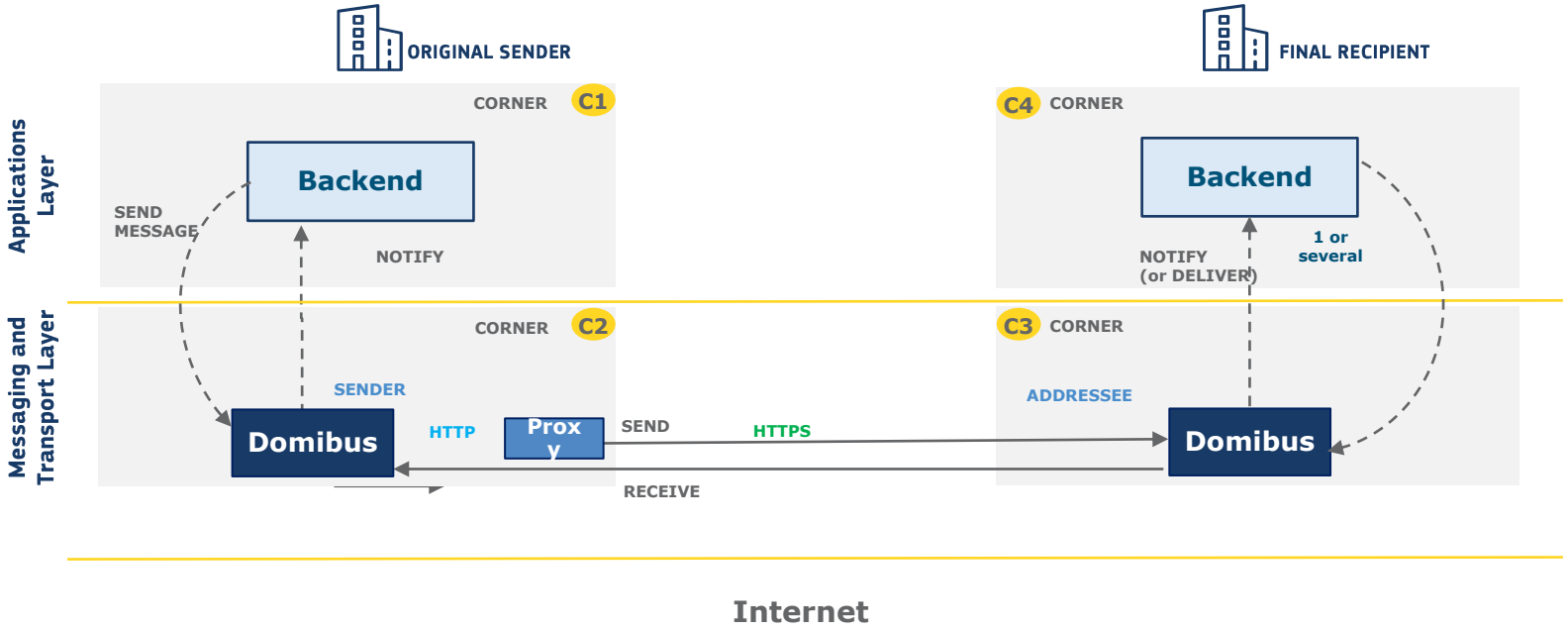
Offload TLS traffic

- In a default approach, Domibus initiates the TLS connection to C3



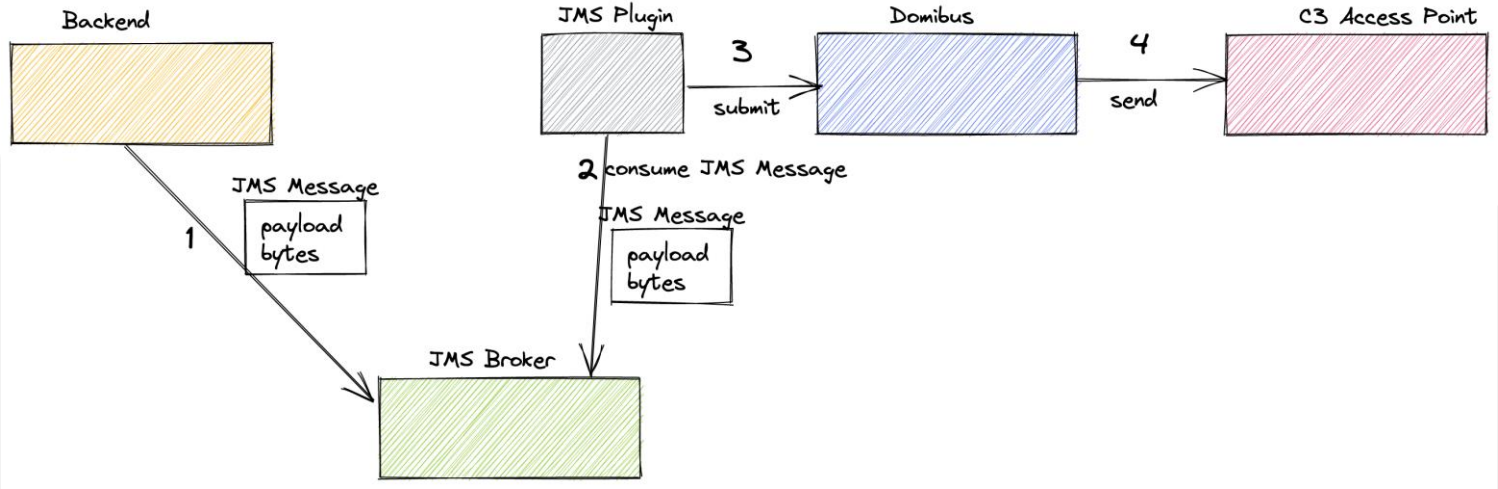
Offload TLS traffic

- When traffic is offloaded, Domibus sends the request to the forward proxy through a HTTP connection
- The forward proxy initiates the HTTPS connection to C3
- Can be used in an organisation which centrally manages the HTTPS traffic



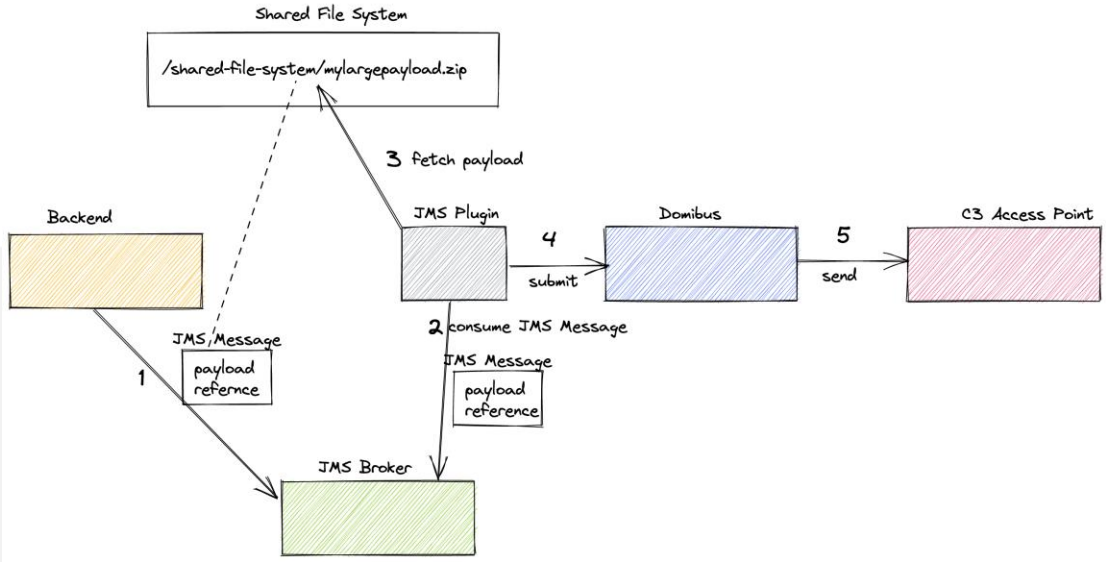
JMS Payload by reference

- In a default approach, the payload bytes are included in the JMS message
- Sending the payload bytes in the JMS message is not suited for large files



JMS Payload by reference

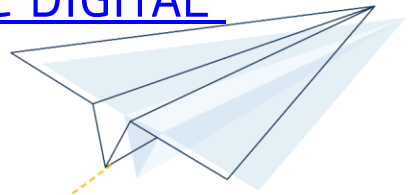
- When using the payload by reference, the JMS message only carries the payload reference
- The JMS Plugin fetches the payload using the reference from a shared file system or from a file server



Questions?

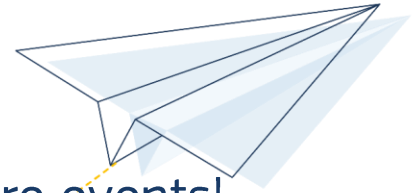
Any further questions?

For policy-related questions, send us an email at at EC-DIGITAL-BUILDING-BLOCKS@ec.europa.eu.



For technical support-related questions, write us at EC-EDELIVERY-SUPPORT@ec.europa.eu.

THANK YOU!



We look forward to welcoming you in our future events!

Please provide your feedback on this workshop by completing an evaluation survey.