



Exchanging large files with eDelivery

Webinar

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

7 March 2023

The background features a dark blue field with several glowing, multi-pointed starburst light effects. Overlaid on this are several translucent, wireframe-like geometric shapes, including cubes and hexagons, which appear to be floating or vibrating. The overall aesthetic is futuristic and digital.

**Welcome to
the webinar on exchanging large
files with eDelivery**

**Monika Kokstaite
Stakeholder Management and
Communication Office**



**eDelivery
ICN 11
May**

[Register](#)



eDelivery

2023 Calendar



**Webinar
eDelivery
value
proposition
28 September**

[Register](#)



**eDelivery
Interoperability
Forum
3 May**

[Register](#)

**Webinar
Dynamic
Discovery
8 June**

[Register](#)



eDelivery webinars <https://tube.network.europa.eu/w/p/sEJXBWDLipQYZfBGDKhvme>

Would like to learn more about eDelivery?

Stay in touch!

eDelivery User Community

The eDelivery User Community space enables stakeholders to share experiences and best practices on the exchange of electronic data and documents between public administrations, businesses and citizens. Pan-European projects (re)using CEF eDelivery have their own sub-communities within this space.

Register for personalised eDelivery news here

 <https://europa.eu/!8rtpfj>

 @edeliveryBB

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

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



Newsletter



Events



Interoperability Forum



Webinars

Introduction and house rules



Please note that this event is recorded.



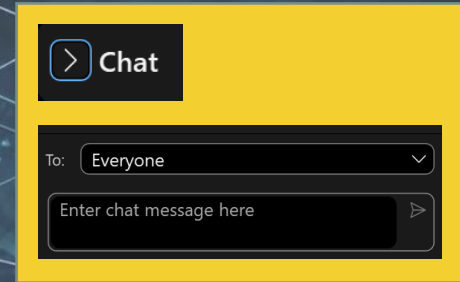
Please ask your questions in the chat section on the right side of the screen.



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



Mute your microphone.



Speakers of the webinar on Exchanging large files



Bogdan Dumitriu

Project Officer

Project officer in charge of
eDelivery implementation in
DIGIT



Cosmin Baciu

External Consultant

Lead Architect of the
eDelivery project



Monika Kokstaite

External Consultant

Stakeholder management
and communication for
eDelivery

Agenda



14h30 – 14h35 | Welcome



14h35 – 14h45 | Introduction to eDelivery



14h45 – 15h35 | Exchanging large files with eDelivery



15h35 – 16h00 | 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.



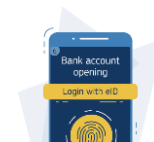
eInvoicing

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.



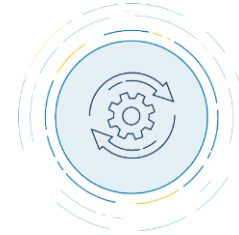
Buy

Buy a compliant, interoperable solution from the market.



Reuse

Reuse sample software available through Digital Europe.

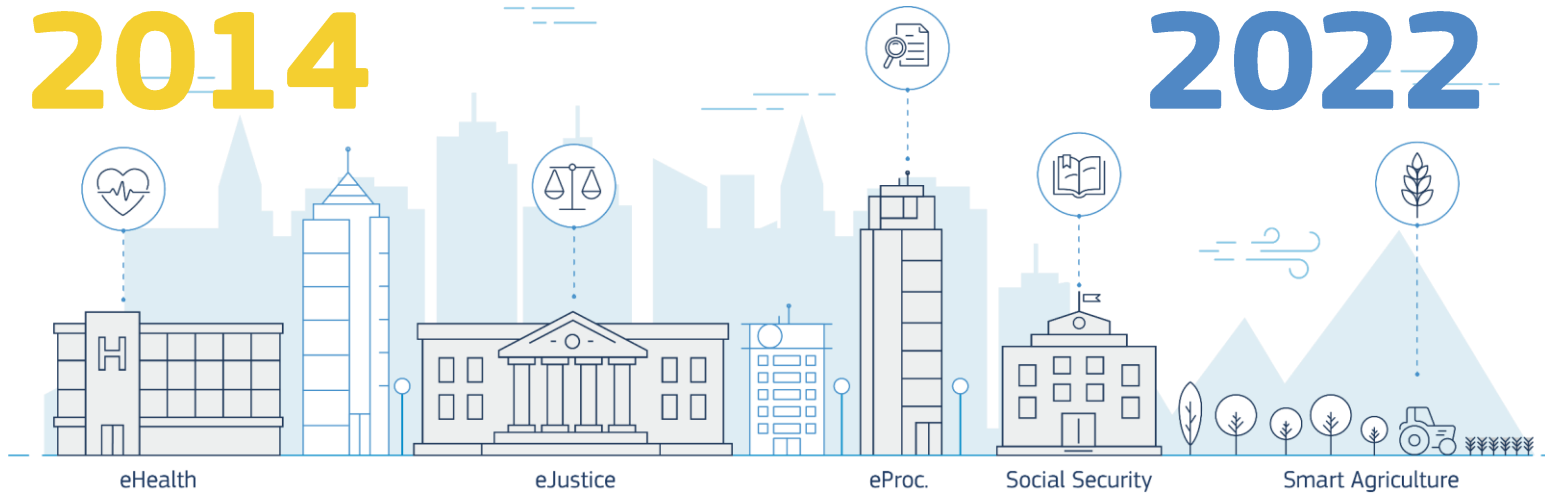


Build

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



- **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**

Receipt of acknowledgment

eDelivery

1

Submit

Sender sends message to sending AP

2

Send

Sending AP processes message

- Validation and compression of the user message;
- Signing of the compressed message;
- Encryption of the signed compressed message.

3

Receive

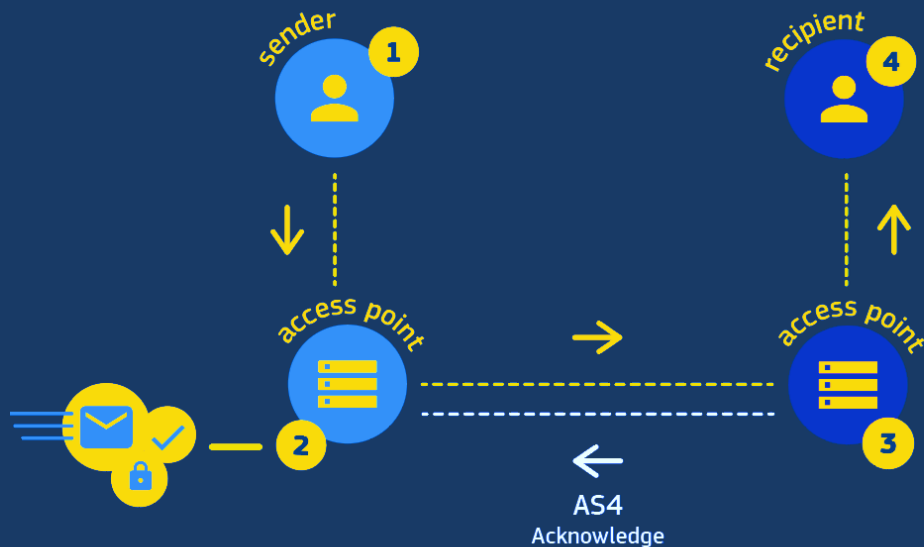
Receiving AP processes message

- Receives and decrypts the encrypted message;
- Verifies the sender's signature;
- Decompresses the decrypted message;
- Validates the original user message;
- Sends the acknowledgement to the sending AP;
- Stores the user message for download.

4

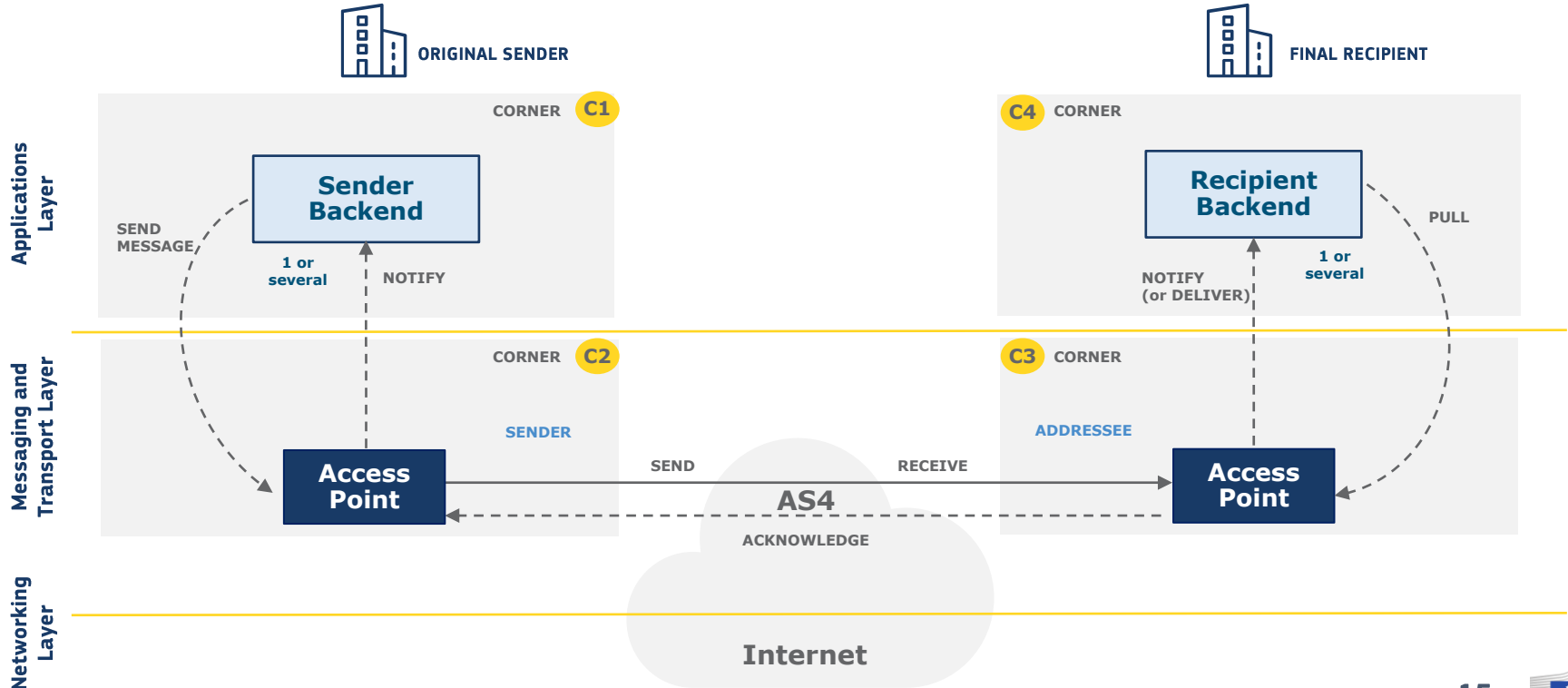
Deliver

Recipient receives message from receiving AP



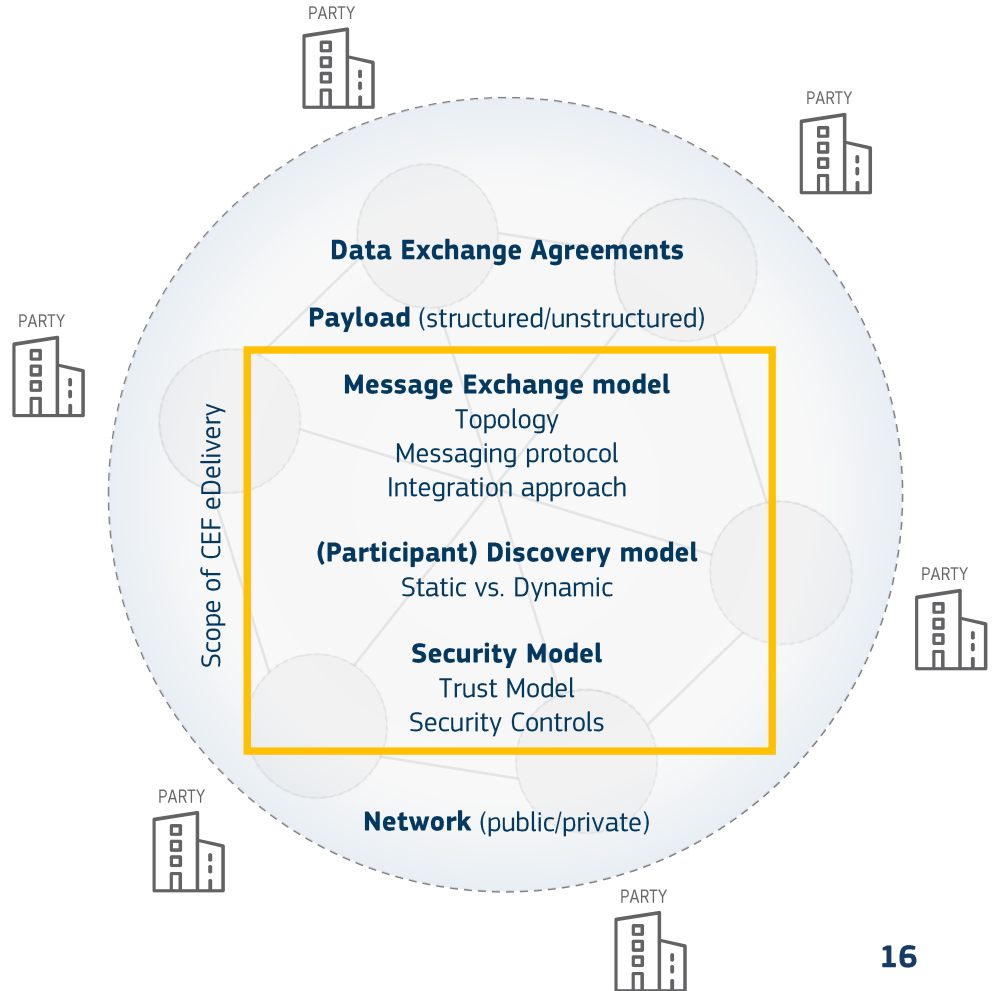
eDelivery Four-Corner Model

Static discovery



A message exchange infrastructure is

A combination of a message exchange model, discovery model and security model on top of the internet, or of a private network, to exchange structured or unstructured information encapsulated in messages.



Trans-European projects based on eDelivery



Once-Only Technical System (OOTS)



European Health Data Space (HealthData@EU)



Interconnection of Insolvency Registers (IRI)



Business Registers Interconnection System (BRIS)



European Product Registry for Energy Labelling (EPREL)



DECIDE Decision



European Maritime Single Window environment (EMSWe)



European Database on Medical Devices (EUDAMED)



Beneficial Ownership Registers Interconnection System (BORIS)



Council Information Exchange Platform (CIxP)



EU Common Entry Gate for Tobacco Reporting (EU-CEG)



Poison Centres Notification System (PCN)



European Hull Database (EHD)






European Crew Database (ECDB)


eDelivery Service offering

SOFTWARE












Sample software maintained by the EC (with documentation)

Access Point (AP)  


Service Metadata Publisher (SMP) 

Service Metadata Locator (SML) 

OPERATIONS SERVICES

Managed services	Testing services	Supporting services
Public Key Infrastructure (PKI)  	Connectivity testing 	Training & Deployment  
Service Metadata Locator (SML)  	Conformance testing  	Service Desk  

STAKEHOLDER MANAGEMENT SERVICES

Onboarding services (for stakeholders)	Community management services
Self-assessment tool	Developers Community
Onboarding of new stakeholders 	Market guide
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

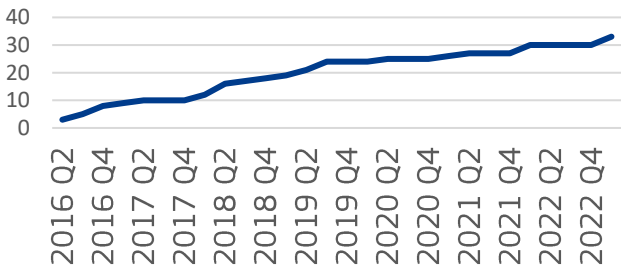
eDelivery AS4 conformant solutions



More information on Digital Europe

[Conformant Solutions >](#)

Evolution of eDelivery AS4 conformant solutions (2016-2023)



Exchanging large files with eDelivery

Cosmin BACIU

Exchanging large files with eDelivery

- ...  **Transfer of large files: overview** ...
- ...  **Transfer of large files using Split & Join** ...
- ...  **Demo** ...
- ...  **Split & Join feature limitations in Domibus** ...

Transfer of large files: overview (1)

- The eDelivery AS4 specifications for Access Points (APs) define no limitation regarding the size of messages.
- In practice, concrete AP implementations each have their own limitations:
 - E.g., Domibus has a limitation of 2GB, coming from how the libraries it uses sign/encrypt messages.
 - To overcome the limitation, use of streaming was attempted when signing/encrypting, but so far unsuccessfully.
- When exchanging large files, it is important to not only ensure that the APs themselves can send and receive them, but also that the various intervening infrastructure components support the transfer of large files:
 - Sender side: forward proxy, etc.
 - Receiver side: reverse proxy, load balancer, WAF, etc.
 - Depending on the speed of the intervening network(s), the timeouts of all components must be adequately configured to avoid cutting the connection during the (long) transfer

Transfer of large files: overview (2)

- Even in the absence of a limitation in software products, it is still preferable to split the transfer into smaller fragments. This limits:
 - Data loss in case of errors while transferring the message.
 - The time it takes to transfer the message.
 - Memory consumption/requirements on both the sending and receiving side.
- If the transmission of a message fails, there is no possibility to resume the transfer, and the transmission will be retried from zero.
- Usually this means the message must also be re-signed and re-encrypted before it can be re-sent over the network to C3.
- When testing Domibus, we observed that it uses 3-4 times more memory than the exchanged message size due to the inner working of the libraries used for signature/encryption (Apache CXF/WSS4J/Santuario). CPU consumption is the same as in the case of sending normal UserMessages, but the faster the CPU the faster the signing and encryption are performed

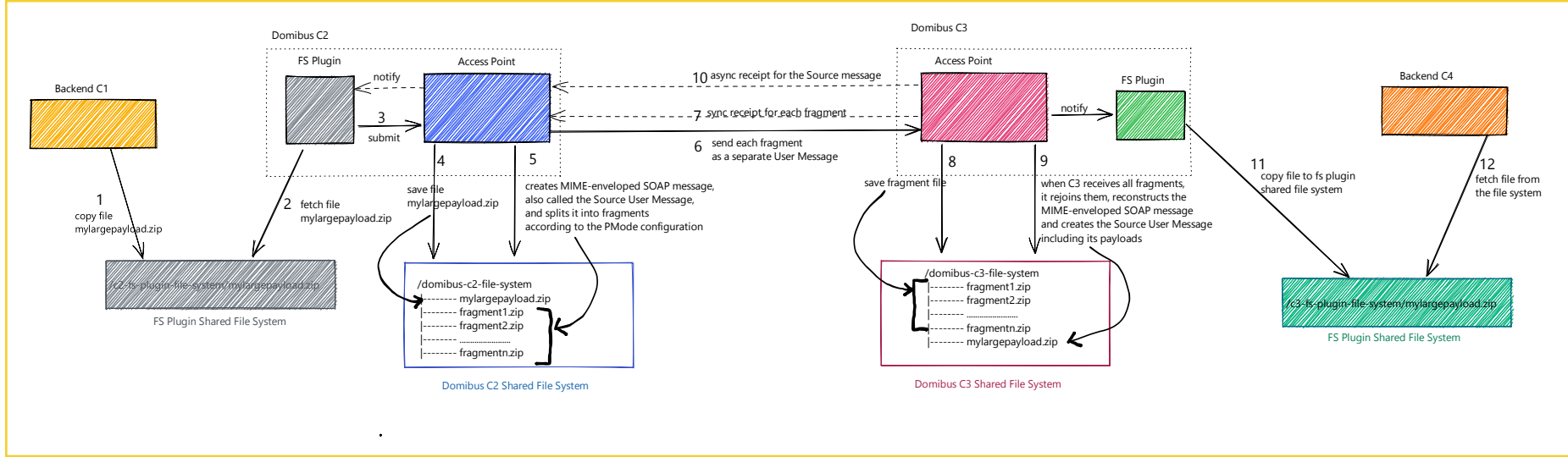
Transfer of large files using Split & Join (1)

- **Large Message Split & Join** is an optional enhancement of the eDelivery AS4 profile.
- **Large Message Split & Join** is based on OASIS ebXML Messaging Services Version 3.0: Part 2, Advanced Features.
- In the specification there are no limitations regarding file types or file formats.
- The **Large Message Split & Join** profile enhancement is:
 - Supported in the Domibus Access Point.
 - To our knowledge, not yet supported in any other AS4 conformant solution.

Transfer of large files using Split & Join (2)

- **Large Message Split & Join** provides a mechanism for a C2 AP to send very large files, even hundreds of GB, to an C3 AP.
- C2 splits the message that must be sent, called source message, into smaller message fragments. C2 sends each message fragment independently to C3.
- Once C3 receives all message fragments it joins them and reconstructs the original source message sent by C2.
- If one message fragment is lost or received incorrectly, only that fragment is resent.
- The use of **Large Message Split & Join** is transparent from C1 and C4 point of view.

Split & Join flow



Split & Join: Security

- Each message fragment is signed and encrypted.
- Source message is neither signed nor encrypted.
- Non-Repudiation of Origin and Receipt is established as follows:
 - Fragments are signed individually. The signature commits the Sender to the content of the fragment message.
 - Collectively, the fragment messages sign the entire source message content.
 - Fragments are individually numbered and the number of fragments a source message is split into is communicated to the Receiver.
 - It is not possible to alter the source message content by re-ordering, adding or omitting fragments without detection.
 - Receipts MUST be generated for message fragments and returned synchronously to prevent unnecessary Message Retry.
 - A Receipt MUST also be generated and returned for the source message. A Receipt for a source message indicates that all related fragment messages have been successfully processed and joined to form the source message has been successfully.
- Reviewed by a security consultant:
 - A third-party security consultant was involved to review the non-repudiation. In their opinion, non-repudiation of signing is obtained by transitivity (or by "assembling"), by considering the sum of the non-repudiation of signing on each piece (fragment). More details can be found in the [document](#) created following the public consultation.

Split & Join: Reliability

- AS4 Message Retry is applied to message fragments. It is not applied to the source message.
- AS4 Duplicate Elimination is applied to both message fragments and to the source message.

Split & Join: Error handling

- For fragment messages, error reporting is done synchronously, as in the Common Profile.
- For the source message, error reporting is done asynchronously.
- Full list of errors can be found in section 4.6 in [OASIS ebXML Messaging Services Version 3.0 Committee Specification](#).
- Most important error codes:

Error code	Short Description	Description
EBMS:0040	BadFragmentGroup	A fragment is received that relates to a group that was previously rejected.
EBMS:0048	BadFragmentNum	An incoming message fragment has a value greater than the known FragmentCount.
EBMS:0051	ReceiveIntervalExceeded	More time than Pmode[].Splitting.JoinInterval has passed since the first fragment was received but not all other fragments are received.

Split & Join: Compression

- Compression of a source message is done by the Sending MSH after submission, before Splitting. This generally results in a better compression ratio than if compression were to be applied separately to fragments after Splitting.
- Decompression is done after all the fragments have been joined, but before message delivery.

Split & Join: Configuration in Domibus

- Domibus must be configured to save the message payloads on the file system:
 - domibus.attachment.storage.location=your_file_system_location
 - domibus.attachment.temp.storage.location=your_temp_file_system_location
- Split & Join must be activated in the Domibus PMode file on a specific leg configuration. Once activated, Split & Join will be used for all messages matching the specific leg configuration.

```
<splittingConfigurations>  
<splitting name="default"  
  fragmentSize="500"  
  compression="true"  
  joinInterval="1440"/>  
</splittingConfigurations>
```

Fragment size in MB

Activate compression for the source message before creating fragments

```
<legConfigurations>
```

```
<!--  
Please add the attribute "splitting"(pointing to a splitting configuration)  
to a specific leg in case you want to activate splitAndJoin feature  
-->
```

How much time is allocated for all files to be sent or received. If the files are not sent within the allocated time, the whole transmission fails

```
<legConfiguration name="pushTestcase1tc1Action"  
  splitting="default"  
  service="testService1"  
  action="tc1Action"  
  defaultMpc="defaultMpc"  
  reliability="AS4Reliability"  
  security="eDeliveryAS4Policy"  
  receptionAwareness="receptionAwareness"  
  propertySet="eDeliveryPropertySet"  
  payloadProfile="MessageProfile"  
  errorHandling="demoErrorHandling"  
  compressPayloads="true"/>
```

Split & Join: Message fragment example

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <mf:MessageFragment xmlns:mf="http://docs.oasis-open.org/ebxml-msg/ns/v3.0/mf/2010/04/"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      env:mustUnderstand="true" href="cid:fragment9"
      wsu:id="_366829fa3df340e81b7cba121d26d280330a40bd05f481383eb7ff778b05baeef">
      <mf:GroupId>dab1c73b-b28d-11ed-a28e-0242ac140003@domibus.eu</mf:GroupId>
      <mf:MessageSize>607160216</mf:MessageSize>
      <mf:FragmentCount>12</mf:FragmentCount>
      <mf:FragmentNum>9</mf:FragmentNum>
      <mf:MessageHeader>
        <mf:Content-Type>Multipart/Related</mf:Content-Type>
        <mf:Boundary>uuid:7040de39-bc91-4562-87c6-c0b2097f0605</mf:Boundary>
        <mf:Type>text/xml</mf:Type>
        <mf:Start>split.root.message@cxf.apache.org</mf:Start>
      </mf:MessageHeader>
      <mf:Action></mf:Action>
      <mf:CompressionAlgorithm>application/gzip</mf:CompressionAlgorithm>
      <mf:CompressedMessageSize>607343991</mf:CompressedMessageSize>
    </mf:MessageFragment>
    <eb:Messaging xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      env:mustUnderstand="true"
      wsu:id="_166829fa3df340e81b7cba121d26d280330a40bd05f481383eb7ff778b05baeef">
      <eb:UserMessage>
```


Split & Join: Reception awareness receipt for the Source message

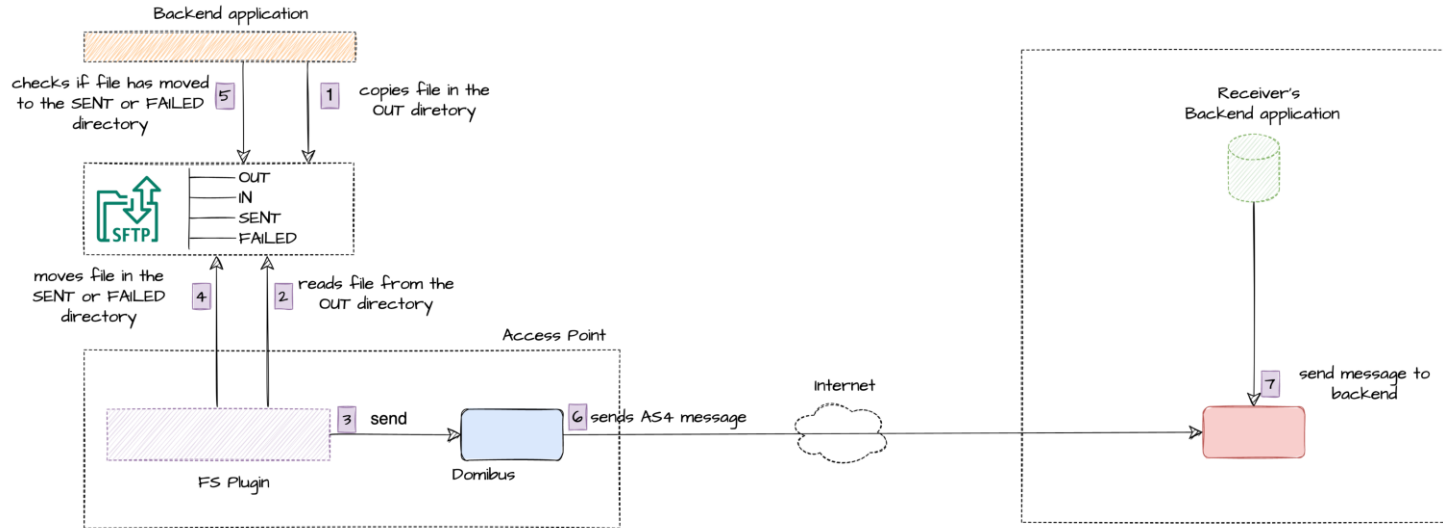
```
<S12:Envelope xmlns:S12="http://www.w3.org/2003/05/soap-envelope" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:eb3="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
  xmlns:ebbp="http://docs.oasis-open.org/ebxml-bp/ebbp-signals-2.0"
  xmlns:ebint="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/multihop/200902/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <S12:Header>
    <eb3:Messaging S12:mustUnderstand="true" id="_ebmessaging_N65541"
      wsu:id="_1c7fd79470ca567dc9d6b9420fdc4dc9885530402243911c53d52a7999cfb0df1">
      <eb3:SignalMessage>
        <eb3:MessageInfo>
          <eb3:Timestamp>2023-02-22T08:51:47.000Z</eb3:Timestamp>
          <eb3:MessageId>23ca1624-b28e-11ed-894f-0242ac150003@domibus.eu</eb3:MessageId>
          <eb3:RefToMessageId>dab1c73b-b28d-11ed-a28e-0242ac140003@domibus.eu</eb3:RefToMessageId>
        </eb3:MessageInfo>
        <eb3:Receipt>
          <eb:UserMessage xmlns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
            xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
            xmlns:env="http://www.w3.org/2003/05/soap-envelope"
            mpc="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC">
```

.....

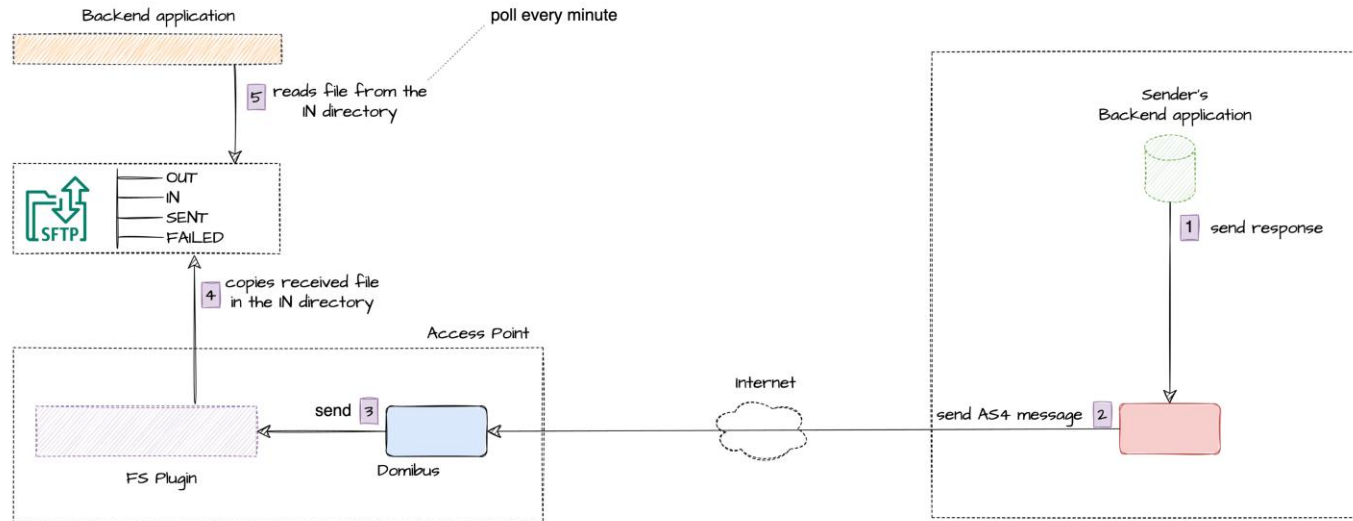
Demo

Send a message using Split & Join and File System Plugin on Tomcat.

Demo – Sending a file with the File system plugin overview



Demo – Receiving a file with the File system plugin overview



Split & Join clustering

In case Domibus is running in a cluster setup:

- On the sending side
 - one C2 node is splitting the source message into fragments
 - all C2 nodes are sending message fragments
- On the receiving side
 - all C3 nodes are receiving message fragments
 - one C3 node is rejoining the fragments into the source message

Split & Join feature limitations in Domibus

Officially,

- Domibus supports Split & Join only in single node setups, when using the File System Plugin on Tomcat.

Unofficially,

- WildFly and WebLogic are also supported
- The WS and JMS plugins are also supported
- Multi-node setups (clustering) is also supported
- Use of these configurations is not recommended in production until made official



Questions?

Received Questions #1

Will you support Docker images for ARM64?

ARM64 images are not planned for the moment.

When will you release production-ready Docker images?

We will release production-ready Docker images based on Tomcat and MySQL at the latest by the end of the year.

Can you please share the documentation for your Docker images (e.g., for domibus-mysql8:5.0.2)?

The documentation for our Docker images is here: <https://code.europa.eu/edelivery/docker/-/blob/22ba82e211ee99f80da2cef9bf8682f06788afb4/README.md>

Received Questions #2

In addition to using docker pull to download images from <https://code.europa.eu:4567>, can we browse Docker image tags and their versions?

You can see the available images at https://code.europa.eu/edelivery/docker/container_registry.

How should we configure a Domibus cluster so that I don't get logged out from the Domibus Admin Console session when the node I'm logged in dies?

Session replication in a cluster is not currently supported. You can introduce a change request to ask for this feature through the [eGovernance space page](#).

Does Domibus support metrics in Prometheus format for external monitoring using Prometheus and Grafana? If not, how can we monitor instances?

Domibus does not support metrics in Prometheus format. Domibus can export metrics in a log file, via a dedicated HTTP endpoint or via JMS. You can find more information in the Domibus Admin Guide section 24. Domibus statistics.

Received Questions #3

Have you done performance tests in Domibus 5.x? Can you please share the results?

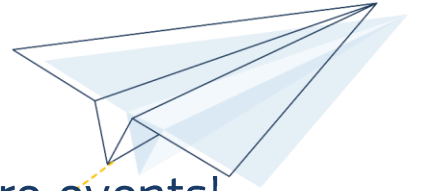
Yes. Starting with 5.0, Domibus can reliably handle a throughput of **more than 1,000 messages/s** and, with added support for table partitioning, ensures this high level of performance even as the size of the database increases. These results were measured during a 2-hour period with Domibus:

- working in single-tenancy mode
- deployed in a 4-node cluster
- using Oracle WebLogic Server
- using Oracle Database configured with partitioning
- configured to receive 500 messages/s of 5 kB and send 1,000 messages/s of 500B

The performance results are dependent on several variables, such as message size, data base disk speed, RAM, etc.

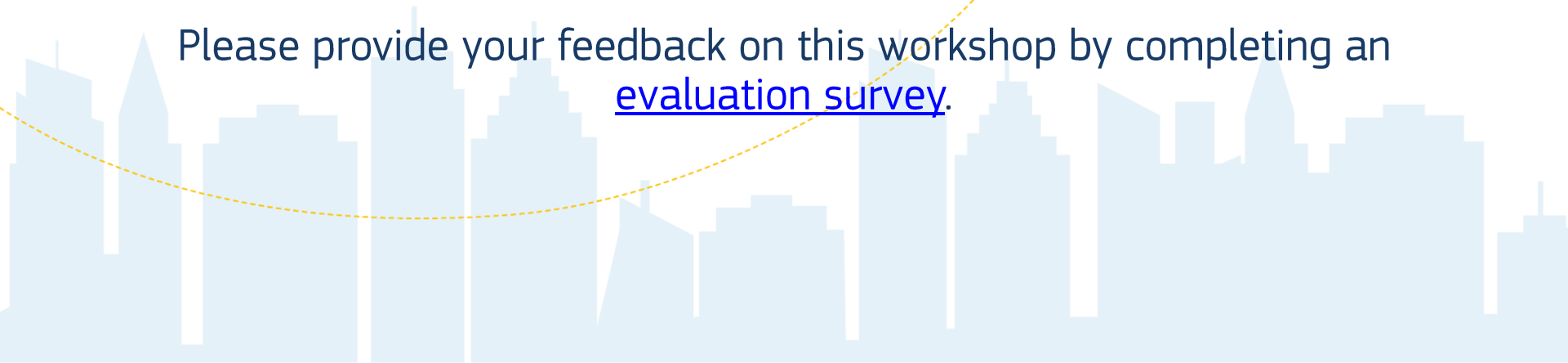


THANK YOU!



We look forward to welcoming you in our future events!

Please provide your feedback on this workshop by completing an [evaluation survey](#).



A dark, atmospheric scene featuring a large, glowing blue cube structure composed of many smaller cubes. The structure is illuminated from within, creating a bright blue glow. The text "The end" is overlaid in a bold, yellow font. The background is dark with some faint light sources and a silhouette of a person's head in the bottom right corner.

The end