NAVELINK

Developer forum

22-02-2024

Navelink.org

Agenda

1) Navelink Platform status & update

2) Navelink Roadmap (Head of concept Navelink)

3) Service development discussions & information
   a) Forum service developers (Each developer)
   b) Forum security and interoperability (Each developer)

4) Overview of Navelink usage

5) Q&A
   a) New questions (All)

6) Presentation - SECOM Service and Lessons Learned by Mikael Olofsson (Navelink)

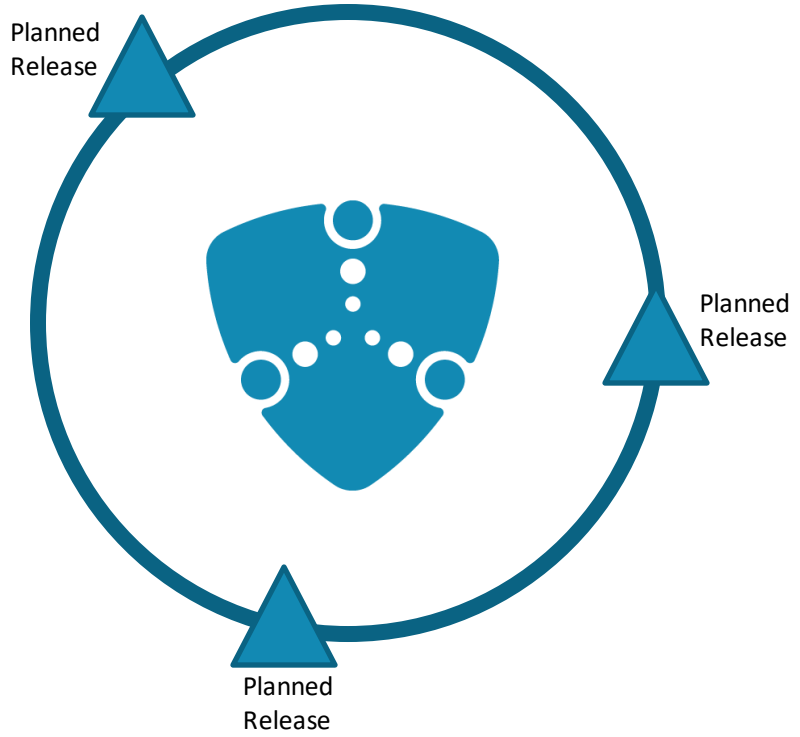
7) Closing remarks

1) Navelink Platform status & update

- Since the last meeting:
  - Work in progress with creation of SECOM Hotel
  - Implementation of MIR v.1.2.2 to PROD
- Future
  - Working on finishing touches on the SECOM Hotel

Received questions

-



2) Navelink Roadmap

0-2 years

2-5 years

Add SECOM Hotel

Increase SECOM Compliance

Add MMS support

MRR

Future MSR

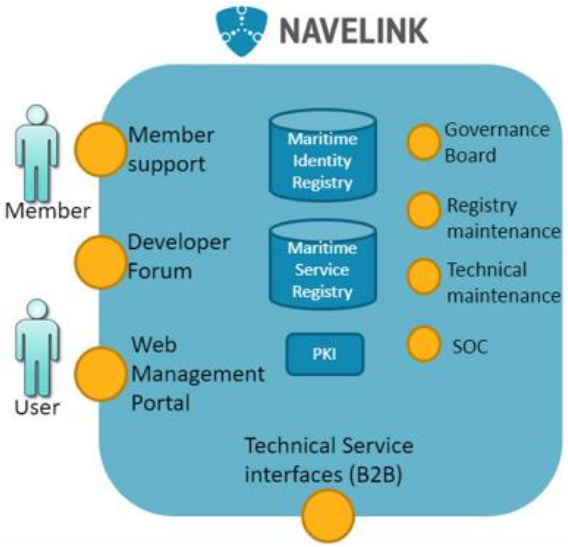
New Management Portal

Support new Service Specifications and Designs

Increase VDES support

Enable subscription on Navelink technical notes

Add support for Service Payment

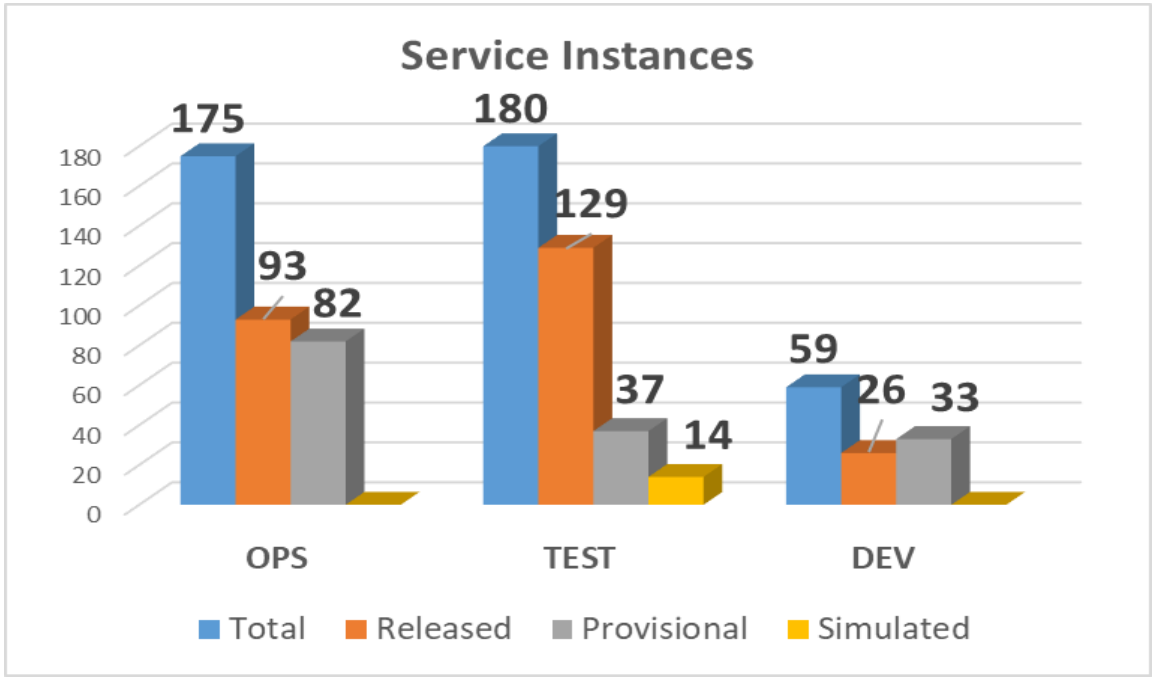


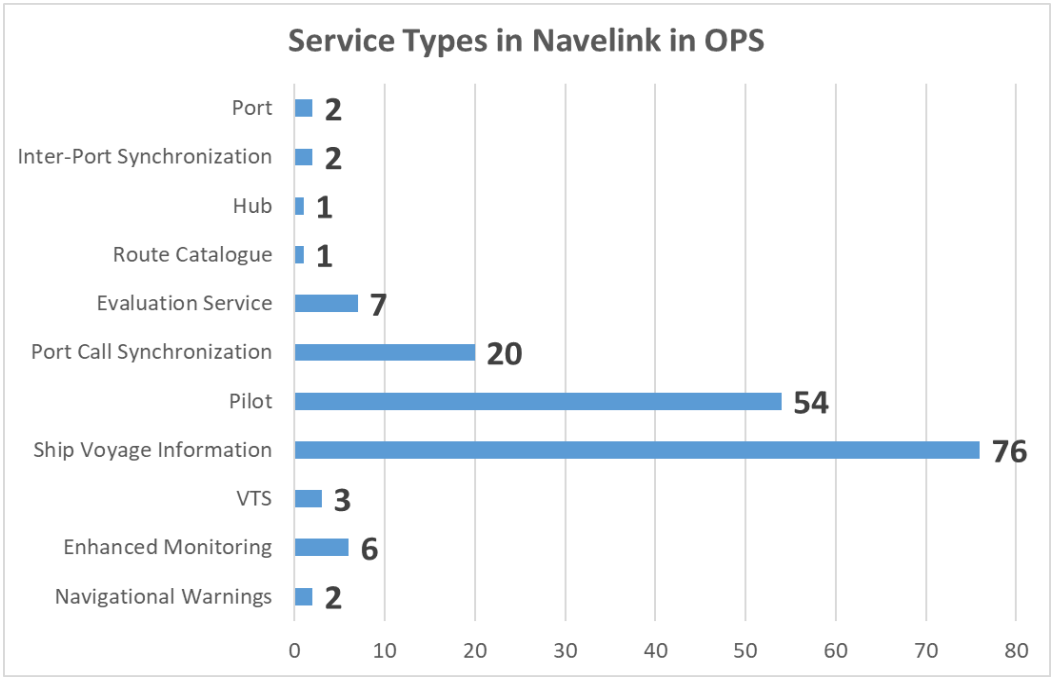
3) Service development discussions & information

- Forum service developers
  - Common discussions
- Forum Security and interoperability
  - Common discussions

4) Overview on Navelink usage

2024-02-13





Navelink Operational environment Service Registrations

Service Specifications: 2 (Voyage Information Service v2.2) + **SECOM Generic Service Specification v1**
Service Technical Design: 2 (Voyage Information Service Design v2.2) + **SECOM Service Design Template v1**
Service Instances: 175

NAVELINK

Services in Operational Navelink

NAVELINK

5) Q&A

- Any Questions? The floor is open.

6) Presentation

- SECOM Service and Lessons Learned by Mikael Olofsson (Navelink)

SECOM Service and Lessons Learned

The following presentation is based on lessons learned from creating a SECOM Service hotel.

Short recap of SECOM Services

In IEC 63173-2 SECOM there is a REST service interface defined, hereafter called SECOM Service.

The architectural design is service oriented where the same REST service interface includes both provided and consumed interfaces, hence in several cases it is expected that there is a SECOM Service in both ends of the communication.

The SECOM Service can be used for both pulling data and pushing data. The pushing of data can be either one-way, or part of a publish-subscribe pattern. When pushing data you can also request acknowledgement to close the loop.

There is also a set of supporting operations for requesting access, pinging the service and reading capability of the service.

SECOM Service interface

Reference: IEC 63173-2 Clause 5

Interface	Comment
Upload	This interface is called when the client uploads (pushes) data to the service. The sender (client) decides the format and protection of the data.
Upload Link	This interface is called when the client uploads (pushes) a reference pointer to large data. The data is downloaded using interface Get By Link.
Acknowledgement	This interface is called as response to Acknowledgement request in Upload.
Get Summary	This interface is called when the client gets a summary of available data from the service. The data is retrieved (pulled) using the interface Get.
Get	This interface is called when the client gets (pulls) data from the service.
Get By Link	This interface is called when the client downloads (pulls) large data by reference given from interface Upload Link.
Access	This interface is called when the client asks for access to data from the service. Response is given by callback to Access Notification.
Access Notification	This interface is called as response to interface Access.
Subscription	This interface is called when the client or server initiates subscription on data from the service. Response is given with interface Upload and Subscription Notification.
Remove Subscription	This interface is called when the client or server removes subscription. Response is given with interface Subscription Notification.
Subscription Notification	This interface is called as response from Subscription or Remove Subscription.
Capability	This interface is called when the client asks for the service capabilities.
Ping	This interface is called when the client checks the availability of the service.
EncryptionKey	This interface is called when sending (pushing) encryption key to a receiver.
PublicKey	This interface is called when the client gets (pulls) the public certificate(s) from the service.

NAVELINK

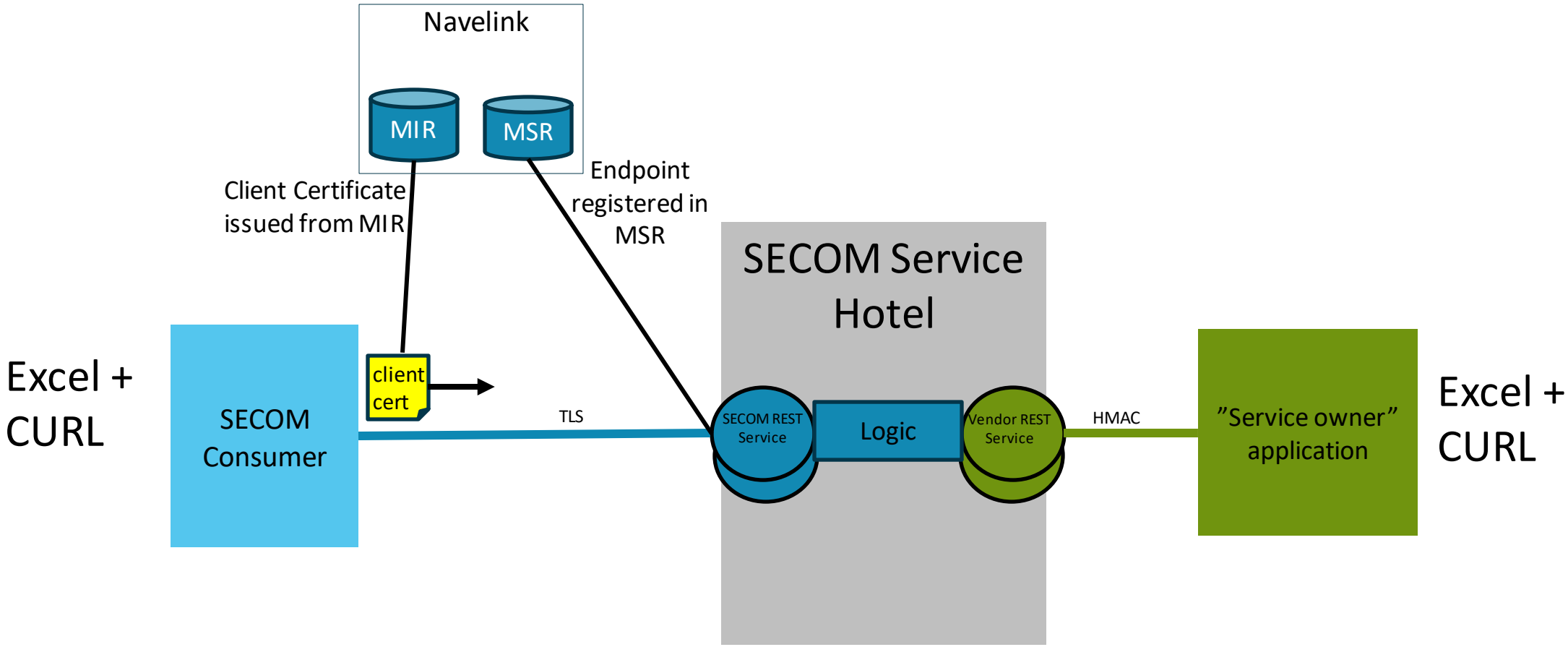
Scenario examples

Example of scenario

1) Service provides downloadable information for consumer to pull
    1) Ship provides the monitored route plan to be downloaded by authorized actors, such as VTS's and Ports
    2) Coastal state provides Pilot Routes to be downloaded
    3) Coastal state provides Navigational Warnings or NoGo areas to be downloaded
    4) Coastal state provides Aids to navigate items to be downloaded
    5) Ports provides information to be downloaded, e.g. berths position and status
2) Service waits for any actor to upload (push) information to process
    1) VTS has an open channel where ships can push their Route Plan, Port Call request, Traffic Clearance request, etc.
    2) Route Optimizer or Weather Router has an open channel where authorized actors can push their Route Plan to be optimized
    3) Coastal state has an open channel where ship can puch their Route Plan and receive all Navigational Warnings along their comp lete route
3) Subscribe on information
    1) Subscription requested by consumer
        1) VTS asks/requests to subscribe on ships Route Plan
        2) Ship asks/requests to subscribe on Navigatinal Warnings
    2) Subscription forced by provider
        1) Ship forces the VTS to be subscribing on their Route Plan
        2) VTS forces Ship to be subscribing on their Traffic Clearance
4) Request access
    1) VTS asks to get access to the Ships Route Plan
5) Supporting operations (Ping, Capability, GetPublicKey, CallService)
    1) Ship/VTS pings the service to see if it is alive
    2) Ship/VTS requests the capabilities of the service; Which payload formats can it handle? Which operations are avilable?
6) Encrypted data
    1) Information Owner decides to encrypt the data and sends encryption key to reciever
7) Linked data
    1) Shore stores larger data objects (e.g. maps, AtoN sets, PDF, pictures, movies…) and sends link to Ship where to get it

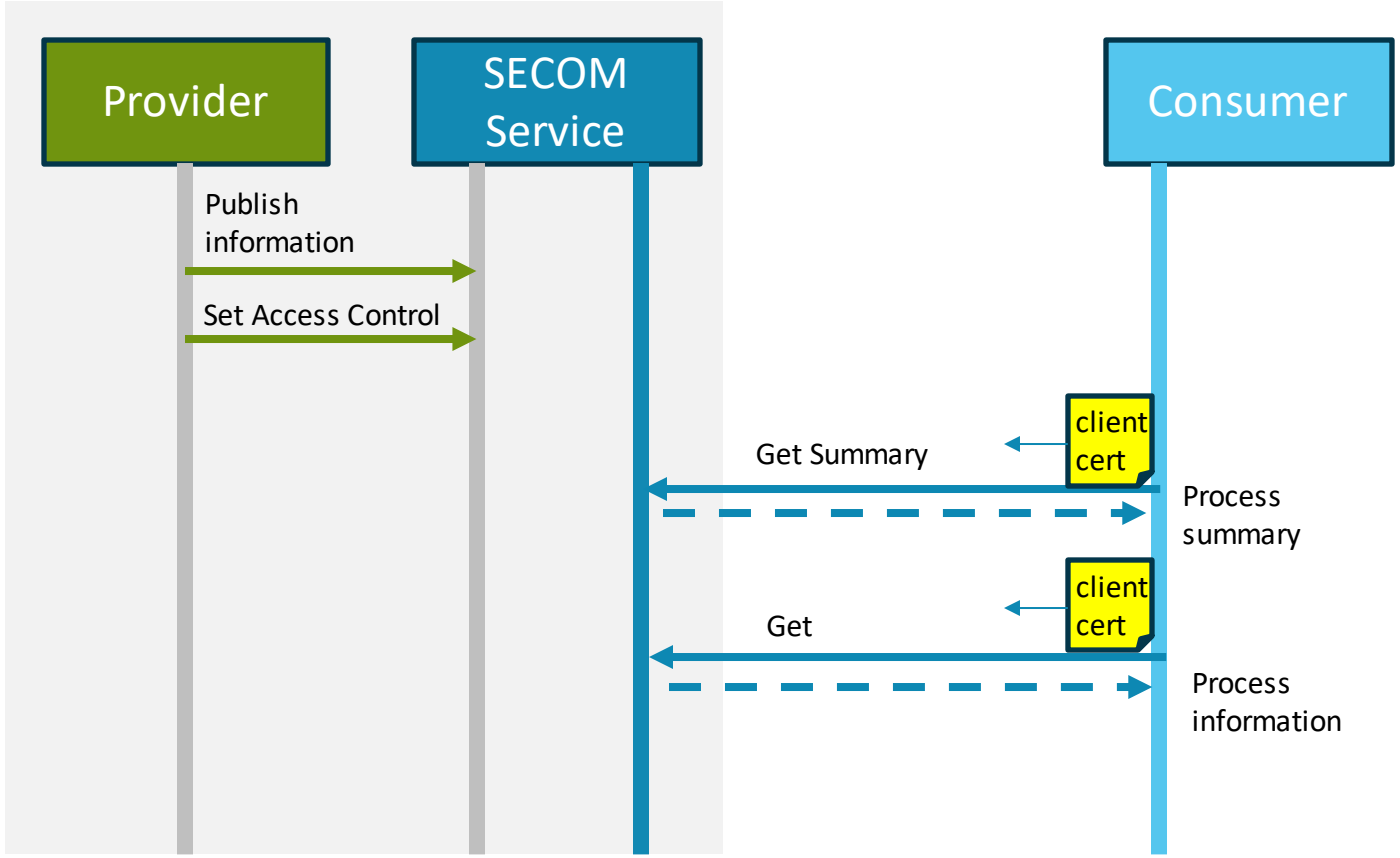
Desktop demo, some details and some lessons learned

Demo setup



Desktop demo, some details and some lessons learned

Scenario 1: Pulling data from a SECOM Service



Get Summary

Request: https://secom.intdev.navelink.org/INT-NLP001/v1/object/summary

+ filtering parameters

Search for data, by
- containerType
- dataProductType
- productVersion
- geometry
- unlocode
- validFrom
- validTo

Response:

{"summaryObject": [{

"dataReference": "43f63047-22ea-439d-8b1c-e632054b29ba",
"dataProtection": false,
"dataCompression": false,
"containerType": 0,
"dataProductType": 24,
"info_identifier": "urn.mrn.stm.voyage.id.operator.44-18_cd032bed-c689-4915-803e-2f82d759accf",
"info_name": "Typical monitored route",
"info_status": "Monitored",
"info_description": "Test data for a typical monitored route",
"info_lastModifiedDate": null,
"info_productVersion": "0.1",
"info_size": 26
},

{"dataReference": "1fb2ec14-eea6-4e59-b8d9-a2edb217e686",
"dataProtection": false,
"dataCompression": true,
"containerType": 1,
"dataProductType": 9,
"info_identifier": "aton.uk.temp_cork_hole_aton",
"info_name": "Temp Cork Hole Test",
"info_status": null,
"info_description": "Aids to Navigation Changes",
"info_lastModifiedDate": null,
"info_productVersion": null,
"info_size": 23
}

],"pagination": {
"totalItems": 5,
"maxItemsPerPage": 100}

Translated this becomes
- dataReference is used to download the data
- dataProtection=false indicates the data is not encrypted
- dataCompression=false indicates the data is not compressed zip
- containertype=0 indicates S100 DataSet
- dataProductType=24 indicates S-421 Route Plan

Metadata of the data object
Info_size indicates the amount of kbyte to download the complete data object

Translated this becomes
- dataReference is used to download the data
- dataProtection=false indicates the data is not encrypted
- dataCompression=true indicates the data is compressed zip
- containertype=1 indicates S100 ExchangeSet
- dataProductType=9 indicates S-125 Aids to Navigate

Metadata of the data object
Info_size indicates the amount of kbyte to download the complete data object

NAVELINK

Get data

Request: https://secom.intdev.navelink.org/INT-NLP001/v1/object? dataReference= 43f63047-22ea-439d-8b1c-e632054b29ba
+ filtering parameters

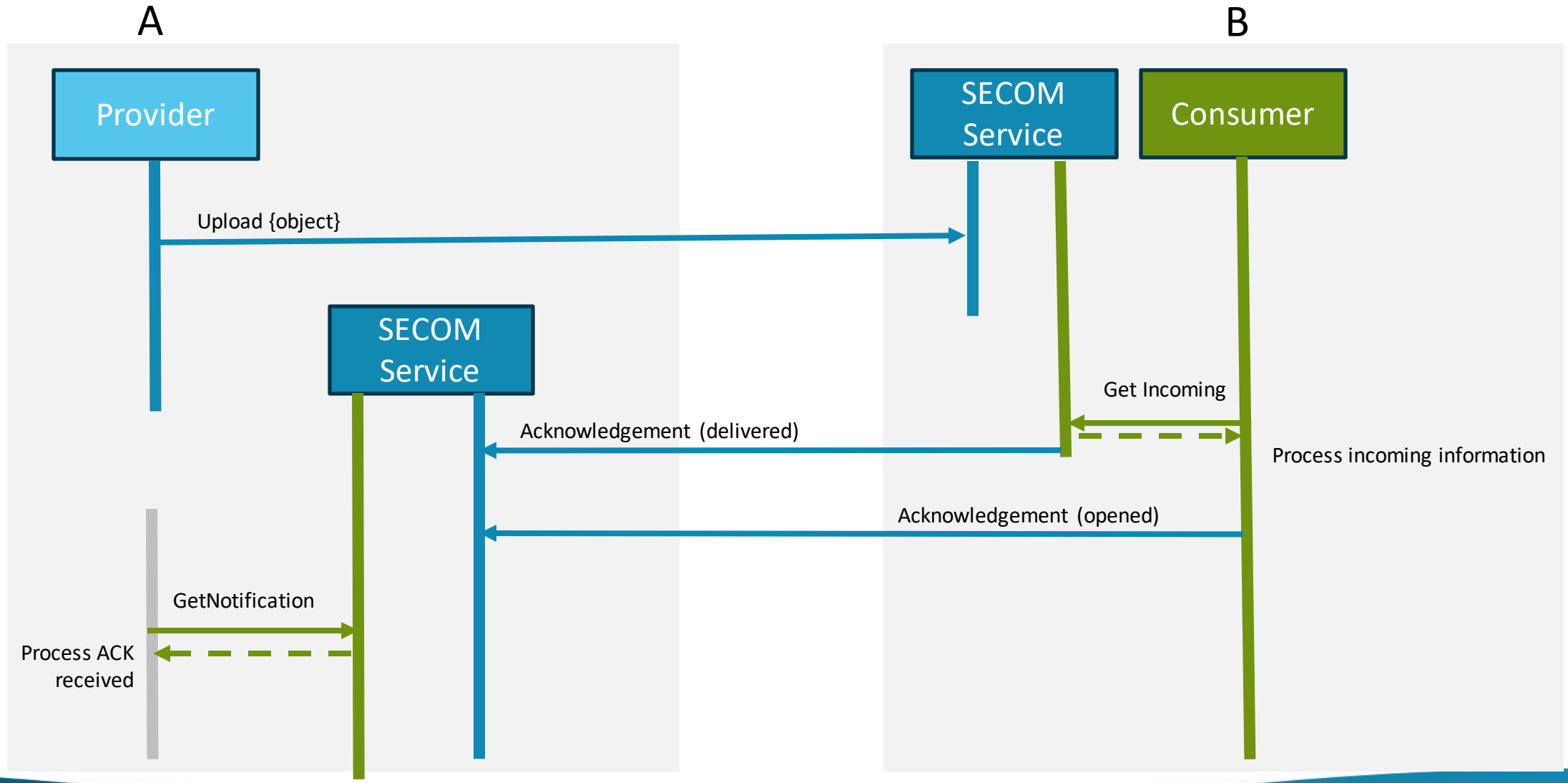
Response:

Request for specific data: by dataReference

Search for data, by
- containerType
- dataProductType
- productVersion
- geometry
- unlocode
- validFrom
- validTo

Desktop demo, some details and some lessons learned

Scenario 2: Service waits for any actor to upload information, processes the information and responds upon



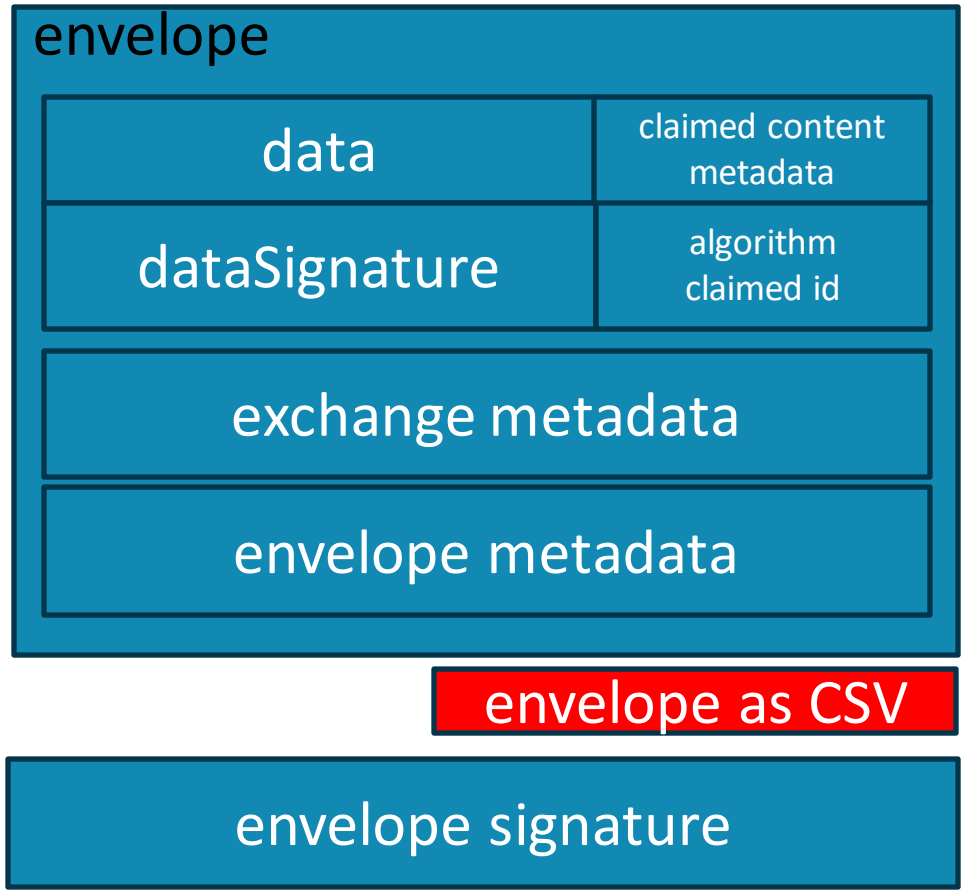
NAVELINK

Upload object

Request: POST https://secom.intdev.navelink.org/INT-NLP001/v1/object{body}

body:

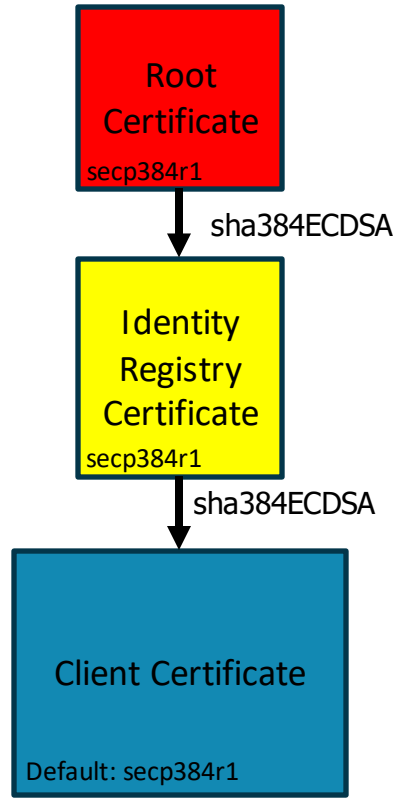

```
{"envelope":
        {"data": "PD94bWTZXQ+",
        "containerType": 0,
        "dataProductType": 24,
        "exchangeMetadata": {
        "dataProtection": false,
        "protectionScheme":"SECOM",
        "digitalSignatureReference":"ECDSA",
        "digitalSignatureValue": {
        "publicRootCertificateThumbprint": "fd4d1c0bc29db9614737e18b4c9bbae4dc8e8c50",
        "publicCertificate": "MIIEuHk=",
        "digitalSignature": "3066870758C4E2191"},
        "compressionFlag": false},
        "fromSubscription": false,
        "ackRequest": 3,
        "transactionIdentifier": "22f70994-0631-4583-9ee7-9d6c6d363206",
        "envelopeSignatureCertificate":"MIIEbHk=",
        "envelopeRootCertificateThumbprint": "fd4d1c0bc29db9614737e18b4c9bbae4dc8e8c50",
        "envelopeSignatureTime": "2024-02-22T07:17:22Z"},
"envelopeSignature":"3067522E65DB370EB10755E00"}
```

Lessons Learned and findings

- **Signing the envelope**
  - **Algorithm to use**; DSA is recommended by NIST to **NOT** use. The recommendation is to use ECDSA instead.
    ECDSA also fits the Navelink default eliptic keys which is EC384bit.
    ECDSA using SHA1 or SHA256 (or SHA384, SHA512)? Currently the tests uses SHA256 but EC384bit keys (which becomes SHA384…?)
    There is also just reference to the algorithm used for the data signature, not explicitly for the envelope signature. These may be different if different keys are used for signing data and signing envelope.
  - **CSV to sign**; Lower case on som fields, others not. CR+LF or not? Currently we remove all CR/LF from the CSV before signing it. The whole procedure depends on **EXACT** same construction of the CSV by the producer and the consumer to verify the integrity of the envelope.
  - **Keys and ID** (certificate) to use.
    The publicCertificate shall contain just the leaf certificate, not a complete trustchain. When downloading certificate as PEM from Navelink you get both the clent certificate and the Identity Registry certificate that has signed the client certificate.
    But when minimised, it can be difficult/impossible to restore and separate the two certificates.

    The publicRootCertificateThumbprint can be sent in different formats, but currently we send it as SHA1.



Lessons Learned and findings

- **TransactionIdentifier versus DataReference**
  - It may seem easy to differentiate between the transaction and the data, but in reality they are easily mixed up. In some cases they seem almost mixed up in the standard as well, but may also be interpretation error.
- **Logic in the service and the dependency to the application behind the service**
  - Several features of the SECOM Service depends on certain logic in the service itself or by the user application behind the service, especially all asynchronous calls, such as Acknowledgement, Access Request and Subscription
- **Callback Endpoint URI lookup in MSR**
- Error messages…

Summary and Other finding related to SECOM

It can be implemented!

Currently tests are running internally and the plan is to release it for DEV and TEST in March

Contents

One major concern by many is that the ”last mile” is not normative in the IEC 63172-2 standard.

Ships consuming other services (pushing and pulling data), this is not an issue. In this case it is always the ship that connects to the other service.

This however affects Ships that want to expose a SECOM Service for others to consume. For different reasons it may not possible to deploy a stable SECOM Service onboard. This also includes Ships participating in a publish-subscribe pattern and Acknowledgement patterns. The solution described in the standard is to deploy the SECOM Service outside ship and then poll the Ship SECOM Service for data, as a mailbox.

7) Closing remarks

- Break for easter/ meeting month that is march
- Next Developer Forum at 25/04-2024

Meeting notes (1/2)

- Navelink is working on the finishing touches with the SECOM Hotel which will be implemented and available on DEV and TEST in a couple of weeks (March).
- MMR and MSR. There are discussion to separate Service Registry so that it contains only (mainly) the service instances, and the service design and specifications are hosted by a Maritime Resource Registry (MRR), such as IALA MRR.
- Mikael Olofsson (Navelink) gave a presentation regarding the SECOM Service and Lessons Learned (se slides 9-21)
  - The SECOM Standard contains three parts, this presentation will only be regarding the SECOM Rest Service part.
  - SECOM services include operations for pulling data as well as pushing data.
    You can ask to be given access to certain data as well as ask for the service's capability (what operations are active and what data can it process)
  - In the SECOM Service Hotel you have one private side and one public side, where the private side is for your own interaction with your own service and the public side for other actors to consume.
  - The SECOM Hotel is built for several purposes; It can be used as reference, it can be used as counterpart to your service development and it can be used as operational service for your own applications
  - In a SECOM Service, the data is signed so that the receiver can always verify the integrity of the data and authenticate the creater.
  - For more information and examples you can contact us at info@navelink.org
  - Question about the envelope: Is it defined in the standard?
    - Yes, it is included in the json body to the POST request and is defined in the SECOM Standard. It is also used in the POST acknowledgement and POST EncryptionKey.
  - The basic idea is if you use SECOM for one step in the data transfer you may remove the envelope when it has been exchanged between two SECOM services, but the data should never be separated from the data signature to enable integrity check.
  - Do you see any benefits of having this extra layer?
    - Yes, to ensure the integrity of the data and to ensure the integrity of the metadata as well and confirm that nothing has been changed. Whether it is worth it or not depends on the risk and the information that is exchanged. The CSV creation and the signature algorithm is what we have the most interoperability issues with. It needs to be very exact to enable verification of the signature.
- If you have any further questions about SECOM or you want a specific demo you can contact us at info@navelink.org

NAVELINK

Meeting notes (2/2)

- In April there will be an IEC meeting regarding the S421 Route plan and SECOM. If you have any more feedback regarding S421 or SECOM, either speak with your national IEC representative or send it onwards to us at info@navelink.org. We can also provide the findings back to you as an information
- Next meeting 2024-04-25

NAVELINK

Navelink.org