NAVELINK

Developer forum

25-05-2023

Navelink.org

Agenda

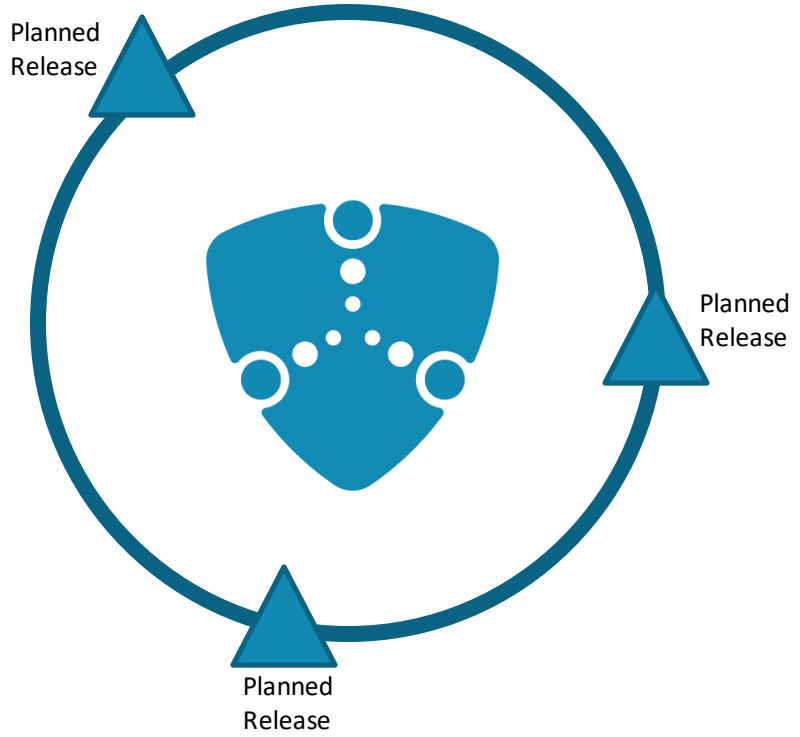
1) Navelink Platform status & update
2) Navelink Roadmap (Head of concept Navelink)
3) Service development discussions & information
   a) Forum service developers (Each developer)
   b) Forum security and interoperability (Each developer)
   c) Ongoing work within the STM-community (Trello) (Each developer)
4) Overview of Navelink usage
5) Q&A
   a) New questions (All)
6) Upgraded G1128 schema
7) Navelink HOW-Tos
8) GetPublicKey
9) Discussion: Navelink + REST + MMS + VDES
10) Closing remarks

1) Navelink Platform status & update

- Since the last meeting:
  - MIR version 1.2.0 tested in internal environment
    - *Awaiting official release before implementing to DEV, TEST and PROD environments.*
- Future
  - Create Navelink specific email adresses in each organization?
  - NCSR 10 meeting decisions?

Received questions

-



2) Navelink Roadmap

1-2 years

2-5 years

Support new Service Specifications and Designs

Increase VDES support

Add GetPublicKey

Add MMS support

Increase SECOM Compliance

Add SECOM Hotel

Add SecretKeyExchange

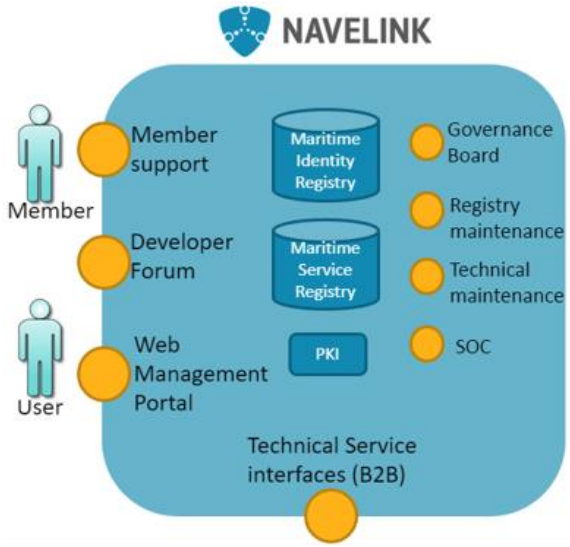
Add MRR usage

Add Service Ledger support

Enable subscription on Navelink technical notes

Enhance functionality to host payload formats

Add support for Service Payment

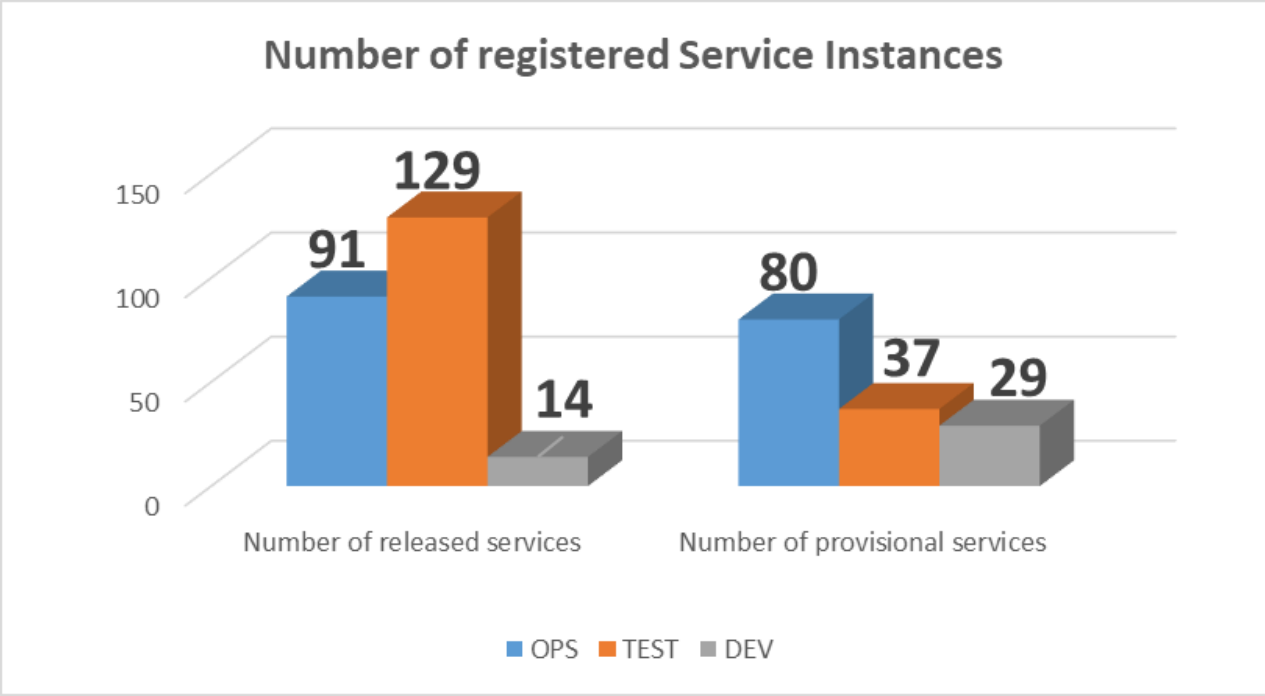


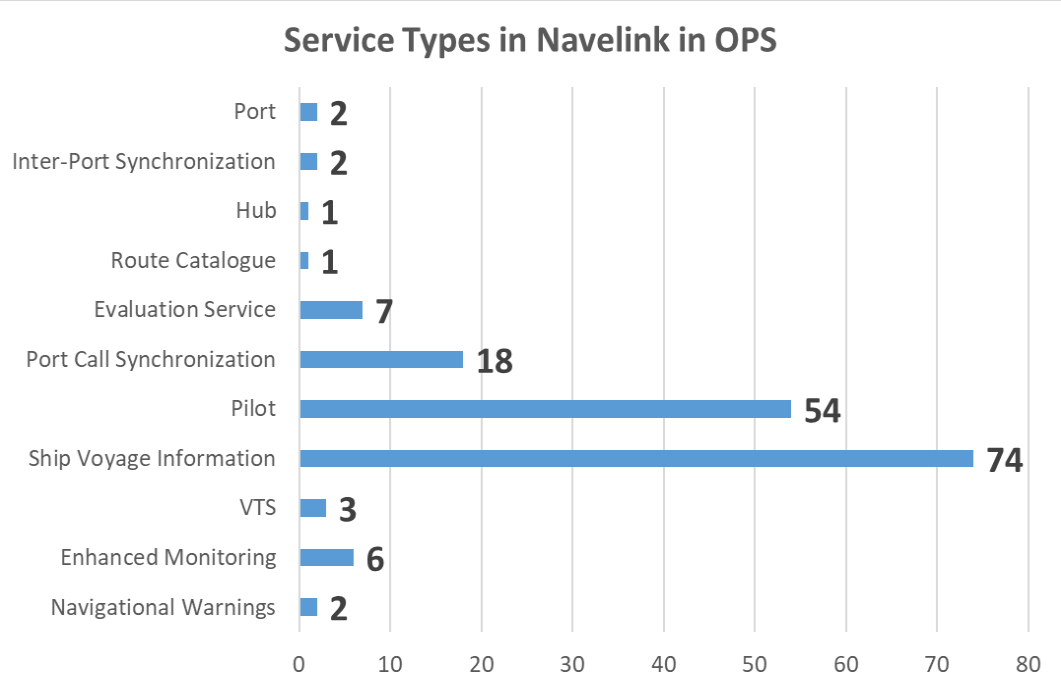
3) Service development discussions & information

- Forum service developers
  - Common discussions
- Forum Security and interoperability
  - Common discussions
- Ongoing work within the STM-community (Trello)
  - Trello check
  - Common standardization work: S-124, S-421, SECOM, General STM news

4) Overview on Navelink usage

2023-05-24





Events since last Dev Forum:

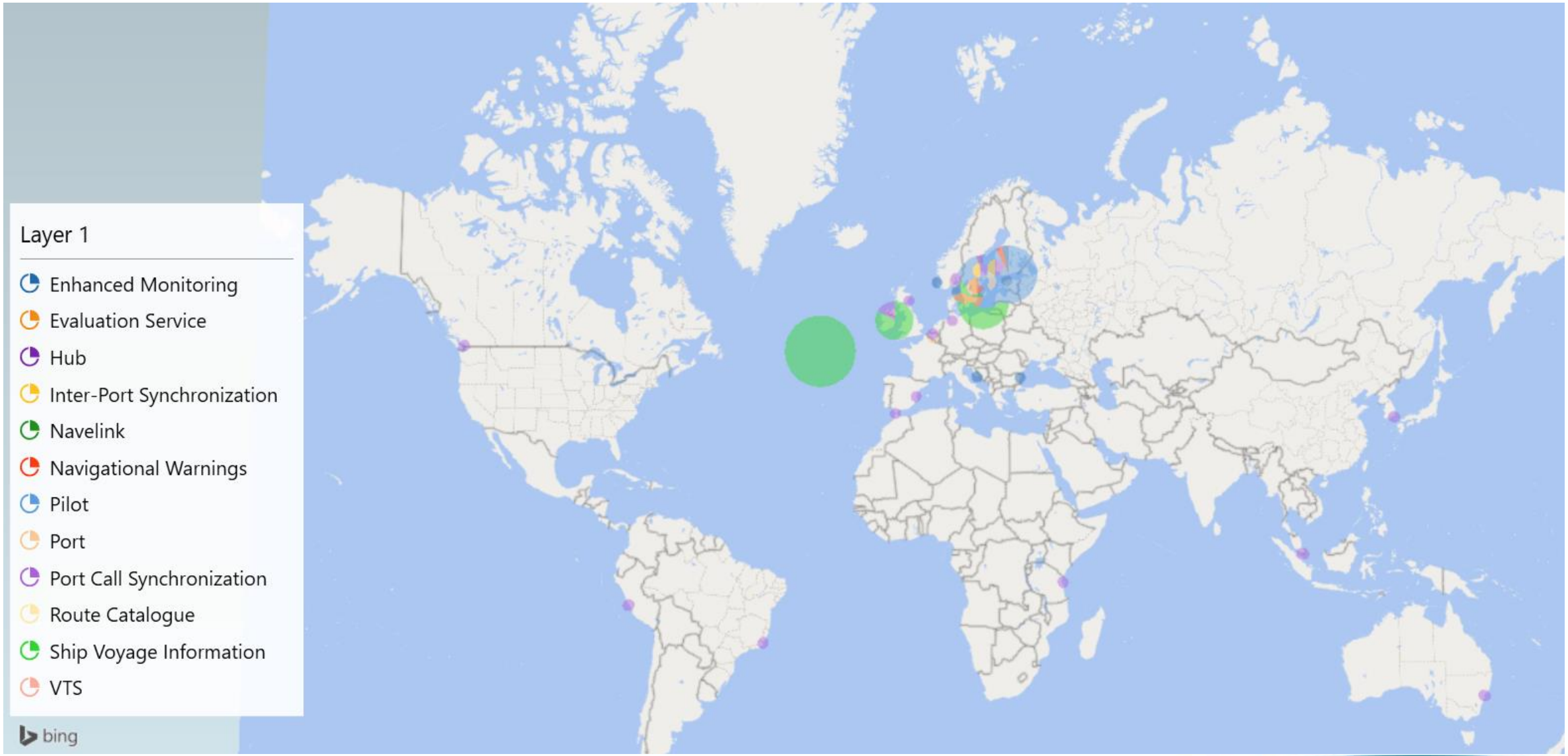
Minor changes

Navelink Operational environment Service Registrations

Service Specifications: 1 (Voyage Information Service v2.2)
Service Technical Design: 1 (Voyage Information Service Design v2.2)
Service Instances: 171

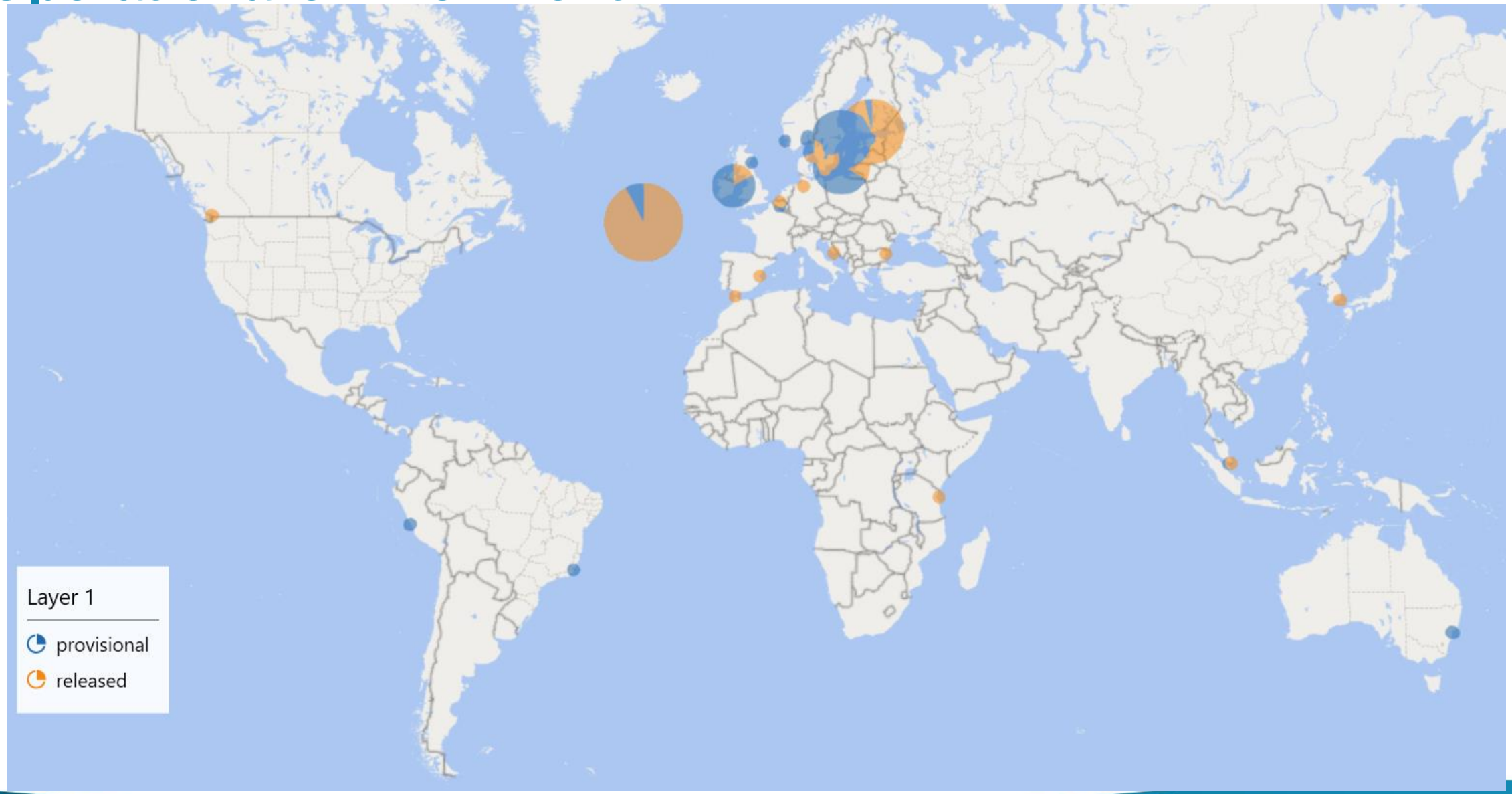
NAVELINK

Operational environment



NAVELINK

Operational environment



NAVELINK

5) Q&A

- Any Questions? The floor is open.

6) Upgraded G1128 Schema (September 2022)

- G1128 v1.0 "early"
- G1128 v1.3 (latest implemented in Navelink)

v1.0 "early"

```
<element name="id" type="ServiceSpecificationSchema:Ser
<element name="version" type="ServiceSpecificationSchem
<element name="name" type="string" minOccurs="1" maxOcc
<element name="status" type="ServiceSpecificationSchema
<element name="description" type="string" minOccurs="1"
<element name="keywords" type="string" minOccurs="1" ma
<element name="URL" type="string" minOccurs="1" maxOccu
<element name="MMSI" type="string" minOccurs="0" maxOcc
<element name="IMO" type="string" minOccurs="0" maxOccu
<element name="serviceType" type="string" minOccurs="0"
<element name="requiresAuthorization" type="boolean" mi
<element name="offersServiceLevel" type="ServiceInstanc
<element name="coversAreas" minOccurs="1" maxOccurs="1"
    <complexType>
        <sequence>
            <element name="coversArea" type="ServiceIns
            <element name="unLoCode" type="string" minO
        </sequence>
    </complexType>
</element>
<element name="implementsServiceDesign" type="ServiceIn
<element name="producedBy" type="ServiceSpecificationSc
<element name="providedBy" type="ServiceSpecificationSc
```

v1.3

```
<element name="id" type="ServiceSpecificationSchema:Service
<element name="version" type="ServiceSpecificationSchema:Ser
<element name="name" type="string" minOccurs="1" maxOccurs="
<element name="status" type="ServiceSpecificationSchema:Serv
<element name="description" type="string" minOccurs="1" maxO
<element name="keywords" type="ServiceSpecificationSchema:Ch
<element name="endpoint" type="string" minOccurs="1" maxOccu
<element name="MMSI" type="ServiceSpecificationSchema:MMSI"
<element name="IMO" type="ServiceSpecificationSchema:IMO" mi
<element name="serviceType" type="ServiceSpecificationSchema
<element name="requiresAuthorization" type="boolean" minOccu
<element name="offersServiceLevel" type="ServiceInstanceSche
<element name="coversAreas" minOccurs="1" maxOccurs="1" type
<element name="implementsServiceDesign" type="ServiceInstanc
<element name="producedBy" type="ServiceSpecificationSchema
<element name="providedBy" type="ServiceSpecificationSchema
```

NAVELINK

Changed Service Endpoint

- When changing the service endpoint for an instance, the certificate domain name is not automatically changed and need to updated as well.
- When updating the Certificate domain, remember to issue new certificates.

- To handle services in Navelink, the permission SERVICEADMIN must be added for the User.

Update of G1128 Service Documentation schemas

https://www.iala-aism.org/product/g1128/

The IALA G1128 contains guidelines/specifications for documentation and description of services. It contains both textual documentation, document templates and XML schemas for describing services in the 3 defined levels; Specification, Design and Instance.

In Navelink, the data model in Service Registry is based on G1128 XML schemas with some complementary attributes. Creating and registering a new service in Navelink Service Registry is mainly done by uploading an XML describing the service. This XML shall follow the G1128 XML Schema (XSD).

Current version in Navelink is G1128 v1.3 schemas but intends for now to be backward compatible with v1.0 "early".

G1128 v1.3

5.3. SERVICE INSTANCE DESCRIPTION XSD STRUCTURE

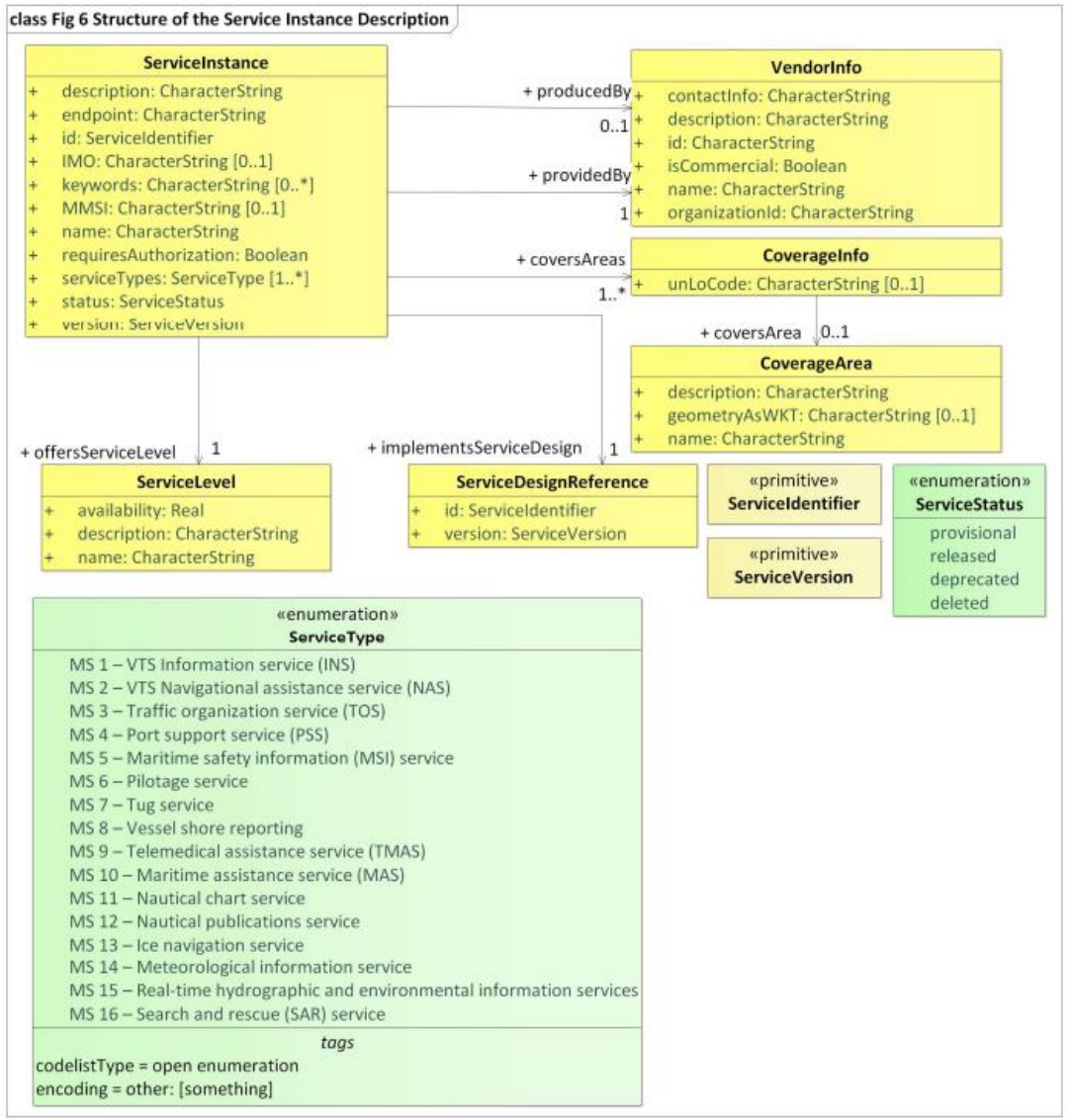


Figure 6 Structure of the Service Instance Description

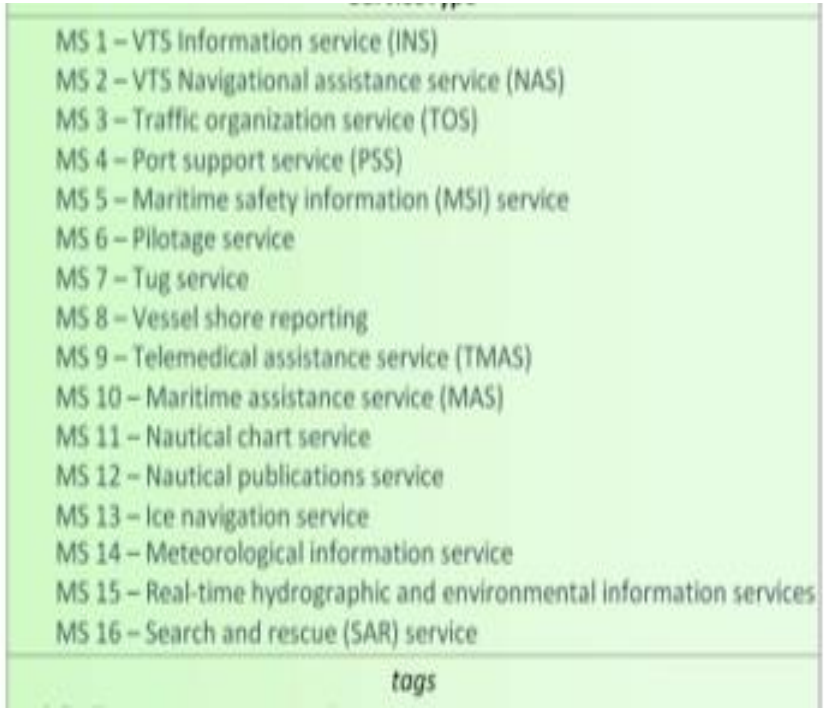
Differences

- serviceType: CharacterString -> ServiceType[1..*]
- URL->endpoint

NAVELINK

serviceType

Enumeration serviceType		
Value	**Acronym**	**Definition**
Enhanced Monitoring	EMS,VTS,STC,SC	Used for shore center services such as VTS
Evaluation Service	EVS	Used for evaluation purpose services only
Fleet Office	FOS	Used for fleet office services
Fleet Operation	FOS	Used for fleet operation services
Hub	HUB	Used for hub services, such as EF-Hub
Inter-Port Synchronization	IPS	Used for inter-port synchronization services (port-port)
Navigational Warnings	NW,NWS	Used for navigation warning services
Pilot	PPU,Pilot	Used for pilot services, such as using Pilot Portable Unit (PPU)
Port	PORT,ETA	Used for port information services
Port Call Synchronization	PCS,PORTCALL	Used for port call synchronization services (ship - port)
Route Catalogue	RC,PRS	Used for route catalogue services, such as Pilot Routes, Reference routes
Route Crosscheck	RCS	Used for route cross check services
Route Optimization	ROS	Used for route optimization services
SAR	SAR	Used for search and rescue services, such as MRCC, JRCC
Ship Voyage Information	SHIP	Used for ships exchanging voyage and route information (today VIS for ship)
Winter Navigation	WNS	Used for winter navigation services
Area Management		TBD
Flow Management		TBD
UKCM	UKCM	Used for under keel clearance management services
Charts	CHART	Used for chart services
Reporting		



- Need to decide and harmonize use of field serviceType

7) Navelink HOW –TO create keys and get them signed

- How to create keys – and the removed function in v1.2
- How to get keys signed by Navelink (CSR=Certificate Signing Request)
- Certificate renewal and certificate revocation
- Validation of certificate
- Certificate revocation list

HOW-TO Issue Certificates in Navelink

Certificates can be issued from Navelink in different ways.

1) Manually through the Web Portal

~~2) With REST service calls to Navelink Identity Registry~~

Certificates in Navelink are formatted as X.509 Certificates (RFC 5280) and are based on private – public key pair. The Certificate is in this context the signed public key. The private key belongs to the creator only.

To avoid transferring the private key on the internet, it's strongly recommended to create the private-public key pair locally, and then transfer only the public part of the certificate to Navelink to be signed and stored as valid certificate attached to the specific identity.

From v1.2 of MCP and Navelink the function to create private key on Navelink server is removed. The key can still be created through Web Management Portal on local browser.

Guidelines for issuing certificate through Web Portal

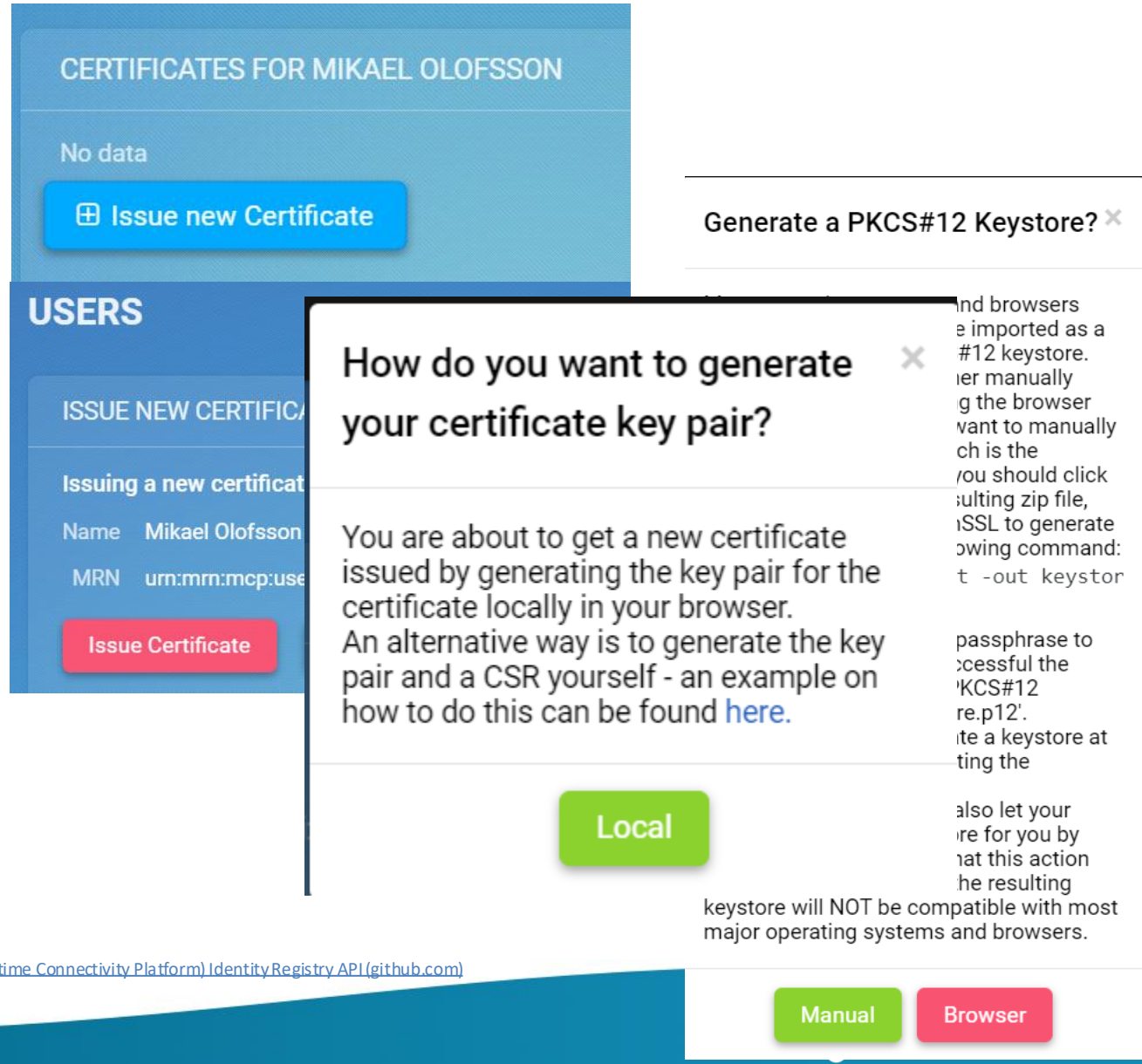
1) Login to the Web Portal for your target environment (each environment in Navelink has its own Root Certificate)

2) Select the entity in focus for the certificate. If the entity is a Service Instance, the entity can be selected either in Identity Registry as ID Service, or the Service Instance in Service Registry part in the portal.

3) Press button "Issue new Certificate"
If you don't see the button, you don't have the right permissions.

4) Follow the guidelines. The only choice now is to select "Local" button. This means that the private key is created in your local browser and is not transferred on the internet.

5) Follow the guidelines. If you don't need a keystore file, press "Manual" and you will receive a ZIP-file with Private key, Public key and Certificate.



maritimeconnectivity/IdentityRegistry: MCP (Maritime Connectivity Platform) Identity Registry API (github.com)

HOW-TO Get keys signed with CSR

Reference: MCC description on GitHub
https://github.com/maritimeconnectivity/IdentityRegistry

The MIR supports signing of PEM encoded PKCS#10 certificate signing requests. It is usually generated for the entity where the certificate will be stored/owned and contains the entity's information such as the organization name, common name (domain name), locality, and country, which will be overwritten by the corresponding information stored in MIR. A CSR also contains the public key that will be included in the certificate. A private key is usually created at the same time that you create the CSR, and is expected to be stored and treated securely.

The algorithm and bit-length pairs of CSR that MIR supports are *RSA:>=2048, DSA:>=2048, ECC:>=224, and EdDSA:256*.

The rationale to use CSR is to protect your private key and never have it in transit on Internet.

Step 1 Generate keys

ECC

$ openssl ecparam -out privateKey.pem -name secp384r1 –genkey
$ openssl ecparam -out privateKey.pem -name secp256r1 –genkey

RSA

$ openssl genrsa -out privateKey.pem 2048

DSA

$ openssl dsaparam -genkey 2048 | openssl dsa -out privateKey.pem

EdDSA

$ TBD

Protect your private keys with passphrase
$ openssl ec -aes256 –in privateKey.pem -out protectedPrivateKey.pem

$ openssl rsa –aes256 -in privateKey.pem -out protectedPrivateKey.pem

RSA:>=2048, DSA:>=2048, EC:>=224, and EdDSA:256

Step 2: Generate CSR

$ openssl req -new -key privateKey.pem -out request.csr

This will prompt you to fill in the attributes of the certificate. For this you can just use dummy data as they in the end will be replaced with data from the MIR database.

$ openssl version
OpenSSL 1.1.1g 21 Apr 2020

Step 3: Send CSR to MIR for signing

Navelink MIR URL
DEV: https://api-x509.dev.navelink.org
TEST: https://api-x509.test.navelink.org
OPS: https://api-x509.navelink.org

Navelink ORG MRN
DEV: urn:mrn:mcp:org:navelink-dev
TEST: urn:mrn:mcp:org:navelink-test
OPS: urn:mrn:mcp:org:navelink

CSR for Service certificate

Navelink DEV

$ curl.exe -i -v -k --output “csr.output” --key “<MIR_PrivateKey.pem>" --cert “<MIR_Certificate.pem>" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.dev.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-dev:navelink/service/<serviceMRN>/<serviceVersion>/certificate/issue-new/csr" -T “request.csr”

Navelink TEST

$ curl.exe -i -v -k --output “csr.output” --key “<MIR_PrivateKey.pem>" --cert “<MIR_Certificate.pem>" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.test.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-test:navelink/service/<serviceMRN>/<serviceVersion>/certificate/issue-new/csr" -T “request.csr”

Navelink OPS

$ curl.exe -i -v -k --output “csr.output” --key “<MIR_PrivateKey.pem>" --cert “<MIR_Certificate.pem>" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink:navelink/service/<serviceMRN>/<serviceVersion>/certificate/issue-new/csr" -T “request.csr”

Replace all yellow marked text with actual data.

NB! Often absolute paths are required by CURL for the keys.

NAVELINK

Step 3: Send CSR to MIR for signing

Navelink MIR URL
DEV: https://api-x509.dev.navelink.org
TEST: https://api-x509.test.navelink.org
OPS: https://api-x509.navelink.org

Navelink ORG MRN
DEV: urn:mrn:mcp:org:navelink-dev
TEST: urn:mrn:mcp:org:navelink-test
OPS: urn:mrn:mcp:org:navelink

CSR for Device certificate

Navelink DEV

$ curl.exe -i -v -k --output “csr.output” --key “<MIR_PrivateKey.pem>" --cert “<MIR_Certificate.pem>" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.dev.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-dev:navelink/device/<deviceMRN>/<version>/certificate/issue-new/csr" -T “request.csr”

Navelink TEST

$ curl.exe -i -v -k --output “csr.output” --key “<MIR_PrivateKey.pem>" --cert “<MIR_Certificate.pem>" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.test.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-test:navelink/device/<deviceMRN>/<version>/certificate/issue-new/csr" -T “request.csr”

Navelink OPS

$ curl.exe -i -v -k --output “csr.output” --key “<MIR_PrivateKey.pem>" --cert “<MIR_Certificate.pem>" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink:navelink/device/<deviceMRN>/<version>/certificate/issue-new/csr" -T “request.csr”

Replace all yellow marked text with actual data.

NB! Often absolute paths are required by CURL for the keys.

NAVELINK

Step 3: Send CSR to MIR for signing

Example

$ curl.exe -i -v -k --output "NLP-DEV_csr.output" --key "C:\Users\MikaelOlofsson\Documents\Navelink\Certificates\Private\NLP-DEV_PrivateKey_Mikael_Olofsson.pem" --cert "C:\Users\MikaelOlofsson\Documents\Navelink\Certificates\Private\NLP-DEV_Certificate_Mikael_Olofsson.pem" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.dev.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-dev:navelink/service/urn:mrn:mcp:service:navelink-dev:navelink:instance:mikael-test-idservice/1/certificate/issue-new/csr" -T "request.csr" 2>"NLP-DEV_csr.error"

```
curl.exe -i -v --output "NLP-DEV_signedKeys_EC256.output"  -k --key "C:\Navelink\Certificates\NLP-DEV_PrivateKey_Mikael_Olofsson_22Nov04.pem" --cert "C:\Navelink\Certificates\NLP-DEV_Certificate_Mikael_Olofsson_22Nov04.pem" --header "Accept: application/pem-certificate-chain;application/json;charset=UTF-8" --header "Content-Type: text/plain" --http1.1 -X POST "https://api-x509.dev.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-dev:navelink/device/urn:mrn:mcp:device:navelink-dev:navelink:vdes:vdes-1/certificate/issue-new/csr" -T "ec256-key-pair.csr" 2>csrEC256.error
```

The result from the CSR request be a certificate chain containing of the signed certificate followed by the intermediate CA that signed it, looking like this:

-----BEGIN CERTIFICATE-----
....
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
....
-----END CERTIFICATE-----

Store the first certificate in a file e.g. Certificate.pem, this is your signed public key. The Certificate can also be downloaded through the web portal.

You have now a Private-Public key pair that can be used for signing of data and authentication.

Verify the certificate

The CSR request gives you the signed public key and the intermediate certificate used for the signing.

The certificate can be verified by

1) Check the certificate with OCSP
   openssl ocsp -issuer navelink-test-ca-chain.pem -cert <Certificate.pem> -text -url
   http://api.test.navelink.org/x509/api/certificates/ocsp/urn:mrn:mcp:ca:navelink-test:navelink-idreg

   the yellow markings need to be adjusted depending on which environment you want to check against.

2) Download the public certificate from portal and compare (can only be done within your own organization)

3) REST call using the certificate as verification of certificate validity
   1) REST call to Navelink MIR
   2) REST call to another service (e.g. VIS instance)

Check Certificate with OCSP

Navelink MIR URL
DEV: https://api-x509.dev.navelink.org
TEST: https://api-x509.test.navelink.org
OPS: https://api-x509.navelink.org

Navelink ORG MRN
DEV: urn:mrn:mcp:org:navelink-dev
TEST: urn:mrn:mcp:org:navelink-test
OPS: urn:mrn:mcp:org:navelink

Navelink DEV

$ openssl ocsp -issuer navelink-dev-ca-chain.pem -cert <Certificate.pem> -text -url http://api.dev.navelink.org/x509/api/certificates/ocsp/urn:mrn:mcp:ca:navelink-dev:navelink-idreg

Navelink TEST

$ openssl ocsp -issuer navelink-test-ca-chain.pem -cert <Certificate.pem> -text -url http://api.test.navelink.org/x509/api/certificates/ocsp/urn:mrn:mcp:ca:navelink-test:navelink-idreg

Navelink OPS

$ openssl ocsp -issuer navelink-ops-ca-chain.pem -cert <Certificate.pem> -text -url http://api.navelink.org/x509/api/certificates/ocsp/urn:mrn:mcp:ca:navelink:navelink-idreg

The trusted Public Root Certificate for Navelink DEV environment is found on
https://api.dev.navelink.org/trust-chain.pem
https://api.test.navelink.org/trust-chain.pem
https://api.navelink.org/trust-chain.pem

Replace all yellow marked text with actual data.

$ openssl version
OpenSSL 1.1.1g 21 Apr 2020

NAVELINK

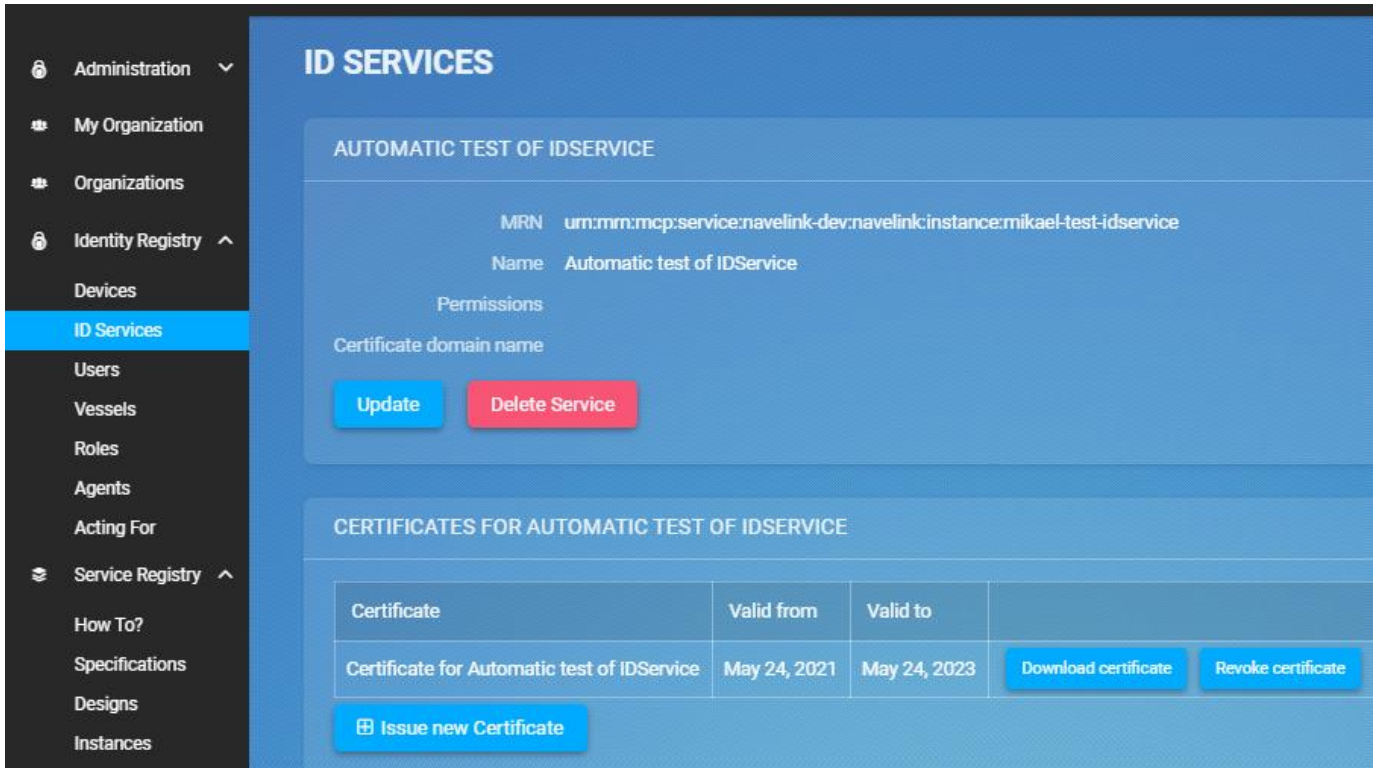
Check Certificate by comparison to downloaded Certificate

Log on to Web portal for the environment

Find your service in **ID Services**

Download the public certificate

The file will be single line with the public certificate.
Compare this file to the first certificate received in the CSR request response.



NAVELINK

Make a REST call using the certificate

If you retrieve the service from Service Registry you made a CSR request for, the public certificate is also received together with revocation status etc.

$ curl.exe -i -v -k --key <privateKey.pem> --cert <certificate.pem> --header "Accept:application/json" --header "Content-Type:application/json" --http1.1 -X GET https://api-x509.dev.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-dev:navelink/service/<ServiceMRN>

Replace all yellow marked text with actual data.

Example:

$ curl.exe -i -v -k --key C:\Users\MikaelOlofsson\Documents\Navelink\Examples\CSR\NLP-DEV_S1_privateKey.pem --cert C:\Users\MikaelOlofsson\Documents\Navelink\Examples\CSR\NLP-DEV_S1_certificate.pem --header "Accept:application/json" --header "Content-Type:application/json" --http1.1 -X GET https://api-x509.dev.navelink.org/x509/api/org/urn:mrn:mcp:org:navelink-dev:navelink/service/urn:mrn:mcp:service:navelink-dev:navelink:instance:mikael-test-idservice

Troubleshooting

- HTTP 406 when using the certificate
  You don’t have the permission. The permission will follow the permissions set on the entity in focus.
- Please be careful to copy and paste from PDF since characters in the examples may be different.

Information in Certificates

MIR entities

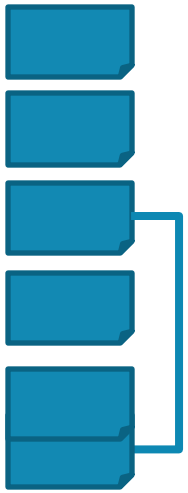
ORG

USER

VESSEL

DEVICE

SERVICE



Service can be linked to Vessel and the details from Vessel i stamped into Service Certificate

MCP Certificate

MCP can issue X.509 certificates for the users which can then be used for authentication. Service providers relying on X.509 certificate authentication must obtain and install MCP root certificate into their webservice.

Please note that the use of *MCP MRN* and *MRN* is based on MCP namespace excluding *Subsidiary MRN*.

The standard information present in an X.509 certificate includes:

- **Version** – which X.509 version applies to the certificate (which indicates what data the certificate must include)
- **Serial number** – A unique assigned serial number that distinguishes it from other certificates
- **Algorithm information** – the algorithm used to sign the certificate
- **Issuer distinguished name** – the name of the entity issuing the certificate (MCP)
- **Validity period of the certificate** – the number of months that the certificate is valid
- **Subject distinguished name** – the name of the identity the certificate is issued to
- **Subject public key information** – the public key associated with the identity

The Subject distinguished name field will consists of the following items:

Field	User	Vessel	Device	Service	MMS	Organization
CN (CommonName)	Full name	Vessel name	Device name	Service Domain Name	MMS name	Organization Name
O (Organization)	Organization MCP MRN					
OU (Organizational Unit)	"user"	"vessel"	"device"	"service"	"mms"	"organization"
E (Email)	User email					Organization email
C (Country)	Organization country code					
UID	MCP MRN					

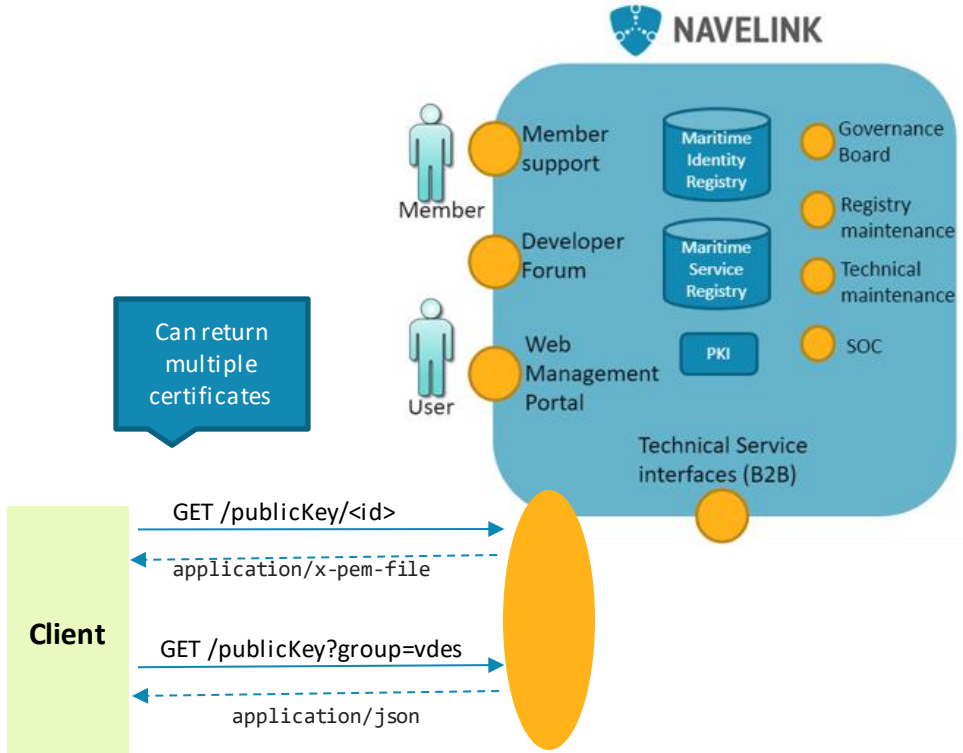
Name	Object Identifier (OID)	Used by
Flagstate	*2.25.323100633285601570573910217875371967771*	Vessel, Service
Callsign	*2.25.208070283325144527098121348946972755227*	Vessel, Service
IMO number	*2.25.291283622413876360871493815653100799259*	Vessel, Service
MMSI number	*2.25.328433707816814908768060331477217690907*	Vessel, Service
AIS shiptype	*2.25.107857171638679641902842130101018412315*	Vessel, Service
Port of register	*2.25.285632790821948647314354670918887798603*	Vessel, Service
Ship MRN	*2.25.268095117363717005222833833642941669792*	Service
MRN	*2.25.271477598449775373676560215839310464283*	Vessel, User, Device, Service, MMS
Permissions	*2.25.174437629172304915481663724171734402331*	Vessel, User, Device, Service, MMS
Subsidiary MRN	*2.25.133833610339604538603087183843785923701*	Vessel, User, Device, Service, MMS
Home MMS URL	*2.25.171344478791913547554566856023141401757*	Vessel, User, Device, Service, MMS
URL	*2.25.245076023612240385163414144226581328607*	MMS

NAVELINK

8) GetPublicKey

Use Cases and needs

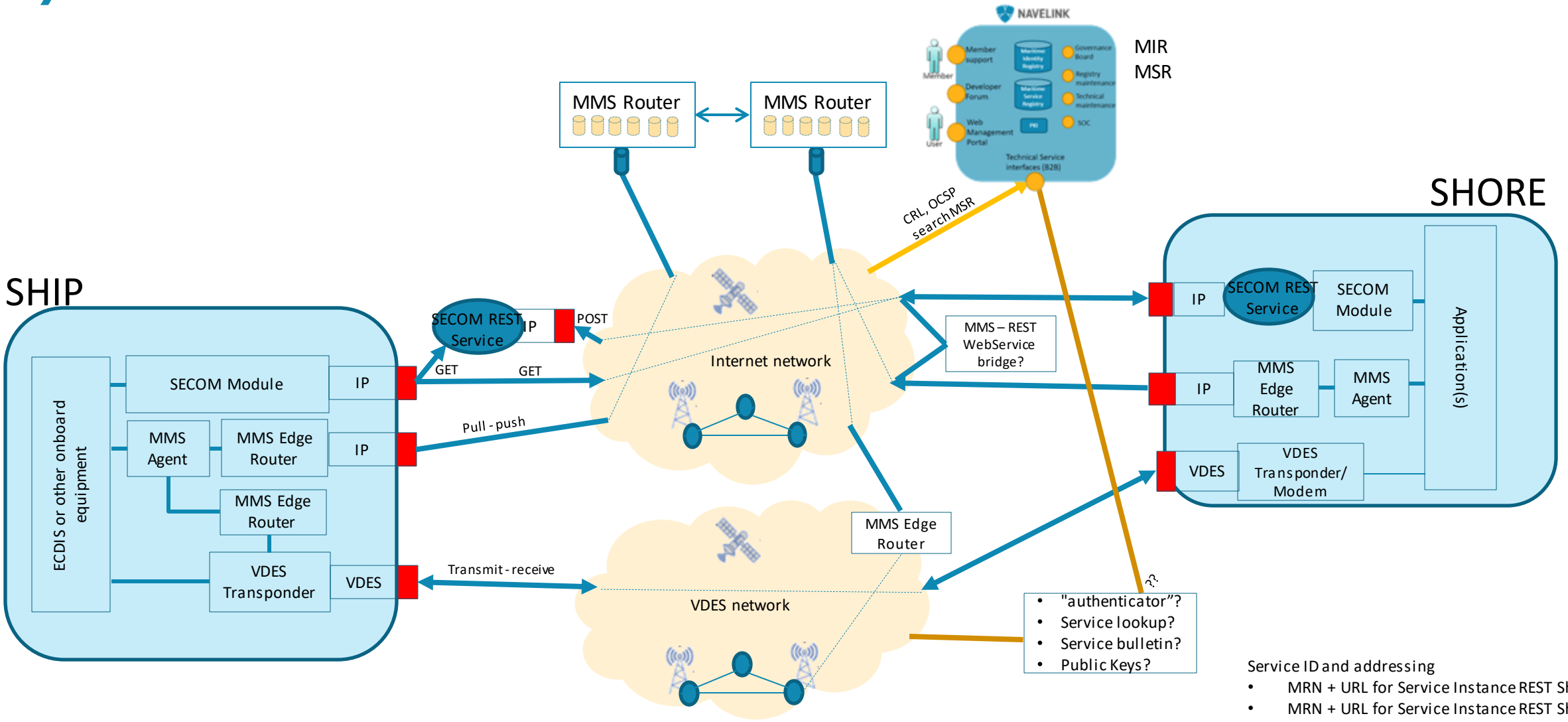
- Create offline list of known public keys
  - e.g. ship will be without internet connection but may receive signed data by other communication means
- Get opposite public key to exchange secret data performing a Diffie-Hellman procedure
  - e.g. HMAC keys, other symmetric encryption keys, private keys, smaller messages



Questions

- Are the identity or identifier always known?
- Need for "Get All Public keys"?
  "Get All Public Keys for VDES devices/group"?
  "Get All Public Keys for Services of type VTS?
- Are there standardised format for returning ID + PEM in a list?
  Own JSON format?

9) Discussion: Navelink + REST + MMS + VDES



Service ID and addressing

- MRN + URL for Service Instance REST Ship
- MRN + URL for Service Instance REST Shore
- MRN for MMS Queue Ship
- MRN for MMS Queue Shore
- MRN + URL for MMS Router

NAVELINK

10) Closing remarks

- Pause for Summer Vacation
  - July & August
- Next Developer Forum at 22/06-2023

Meeting notes (1/2)

- Service registry as an open registry contains today contact mails to persons, but it would beneficial here to use functional mails instead of personal mails. A risk with common mailboxes is they can slip under the radar, but it might still be good to have them. SMA already has one such mailbox used in the Service Registry.
  - If you have a function email that you want us to use for general emails, Dev. Forum and Service registry, please let us know at info@navelink.org
- NCSR (Navigation Communication Search and Rescue) is a subcommittee of IMO. On the May agenda was SMAs proposal for mandating route exchange using S-421 and the proposal was approved. We need to note that although it was accepted, and it was decided that it should be implemented in ECDIS from now on it is NOT a regulation as of now. S-421 route plan, will follow same introduction process as other ECDIS standards, so it will be recommended from 2024 and mandatory in 2029.
  - SMA currently has two project ongoing, one of which is Navigational Assistance, which will use Navelink for publishing the services.
  - Thinking of making the SECOM compliance much more accessible with raspberry pi:s acting as support systems.
    - *Security is an issue and VDES may be one way to move away from the internet and the risks it poses, discussions are currently ongoing.*
- Should we continue with the Trello or should we drop it? We will pause it and move any issues to other platforms.
- We have different ways of communicating VDES, MMS etc. One thing that has been an issue is the distribution of these functionalities with current APIs to the ships and this is an ongoing discussion. How can we distribute the capabilities of receiving messages through SECOM?
  - We return to this point in later notes as the image of the communication is in a later slide from where this discussion arouse.
- Last September Navelink was upgraded to G1128 v.1.3 schema where one of the changes is that the XML field "endpoint" was previously called "URL" and many use "URL" and the old XML schema for registration. Currently you can register services with both the old version and the new version of G1128 schemas.
  - Should we continue supporting "URL" or should we just use "endpoint" and change current registrations? Will Search and GET instances be affected? Mikael will check until the next Developer forum. Peter checked quickly and as long as you parse the JSON instead of xml you should not have any problems. Important viewpoint: More important to keep backwards compatibility in Search and Get than on registration of new instances.

Meeting notes (2/2)

- Navelink plan to provide Public keys for the VDES testbed before the summer.
- In MIR there is a Vessel entity where you can add more information about the vessel. That Vessel information will be visible in certificates issued on the vessel entity. You can also attach that Vessel to a Service and the vessel infromation will also be stamped into the Service Certificate. So you can use either service certificate or vessel certificate to sign the data.
  - We recommend using the service entity and certificate as the most generic entity and most informative content in the certificate, especially on ship side. Optionally the service can be registered in Service Registry, but that would be for the reason to publish the service, not for the reason to sign data and announce who I am for authentication.
- Please send us feedback on the HOW-TO and which functions you would like us to go through! PowerPoint Presentation (navelink.org)
- Mikael believe it is time to create SECOM services for S-421 Route Plan and other S-products, and plan for the phase out of VIS services using RTZ.
- There is uncertainty regarding what formats will be transferred over the VDES and MMS. Stefan Pielmeier is leading a group for standardization of VDES by RTCM.
  - Specification for MMS can be found at the github for those in the MCC MMS Work group. Participation is free. If you want access and joining, please contact us or MCC.
- Next meeting 2023-06-22

NAVELINK

Navelink.org