Issued by	Date
Conceptual team	2022-01-24
Approved by	Classification
M Olofsson	Consortium Unclassified Released Public

NAVELINK

Navelink Industry Consortium

HOW-TO Issue Certificate in Navelink

DocId: 0189
Version: v1.0

Navelink.org

HOW-TO Issue Certificates in Navelink

Certificates can be issued from Navelink in different ways.

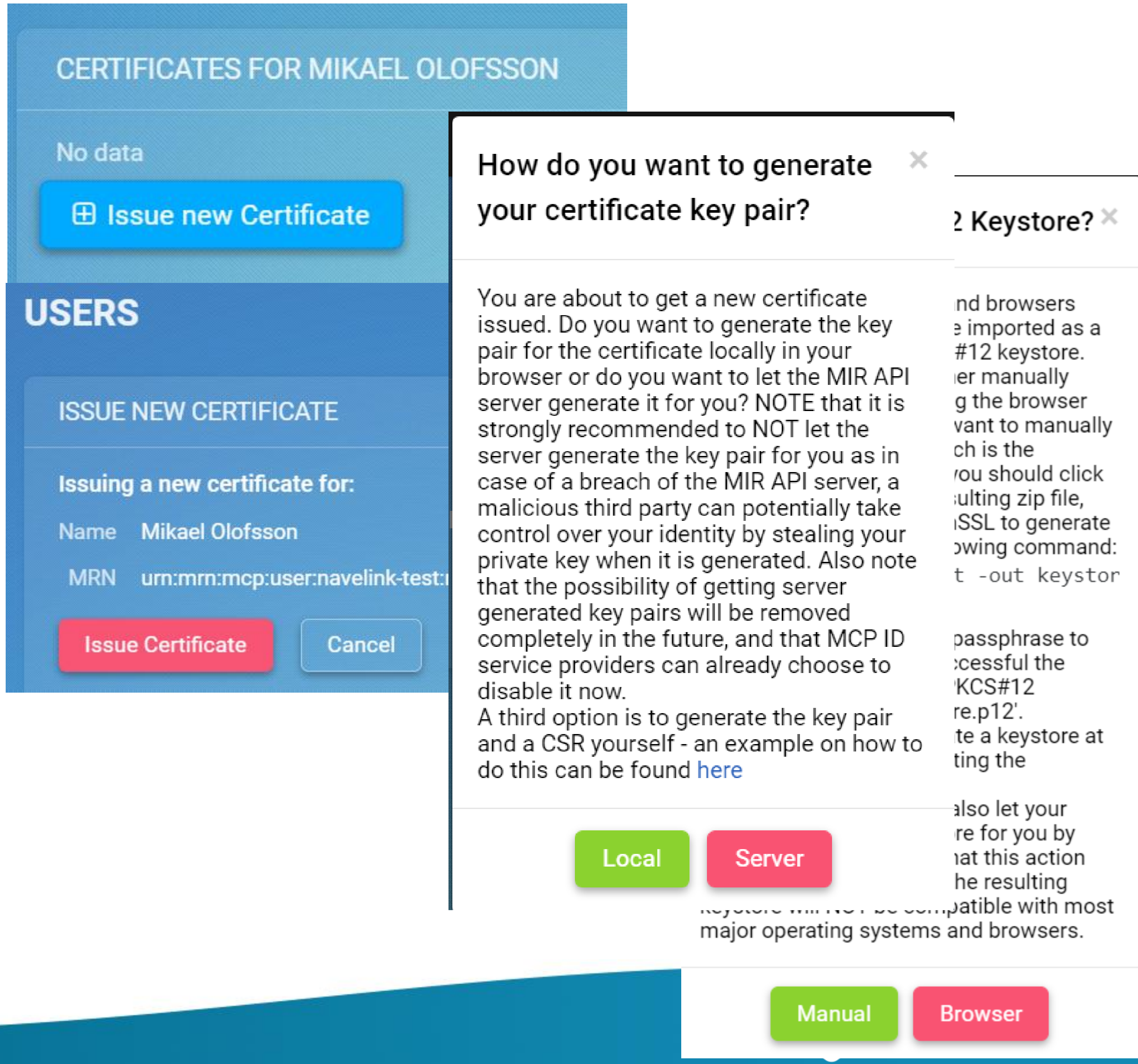
1) Manually through the Web Portal
2) With REST service calls to Navelink Identity Registry

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.

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 recommendation is to always select "Local" button. This means that the private key are 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 the signed public key, your public Certificate.



Issuing a new certificate for:

Name Mikael Olofsson

MRN urn:mrn:mcp:user:navelink-test:

How do you want to generate your certificate key pair?

You are about to get a new certificate issued. Do you want to generate the key pair for the certificate locally in your browser or do you want to let the MIR API server generate it for you? NOTE that it is strongly recommended to NOT let the server generate the key pair for you as in case of a breach of the MIR API server, a malicious third party can potentially take control over your identity by stealing your private key when it is generated. Also note that the possibility of getting server generated key pairs will be removed completely in the future, and that MCP ID service providers can already choose to disable it now.
A third option is to generate the key pair and a CSR yourself - an example on how to do this can be found here

? Keystore?

nd browsers
e imported as a
#12 keystore.
er manually
g the browser
vant to manually
ch is the
you should click
sulting zip file,
SSL to generate
owing command:

```
t -out keystor
```

passphrase to
ccessful the
KCS#12
re.p12'.
te a keystore at
ting the

also let your
re for you by
hat this action
he resulting
patible with most
major operating systems and browsers.

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.

This procedure can also be made through the Web portal (from v0.12).

Step 1 Generate keys

ECC

```
$ openssl ecparam -out privateKey.pem -name secp384r1 –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

NAVELINK

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

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"
```

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 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
```

Replace all yellow marked text with actual data.

```
$ openssl version
OpenSSL 1.1.1g  21 Apr 2020
```

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

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

NAVELINK

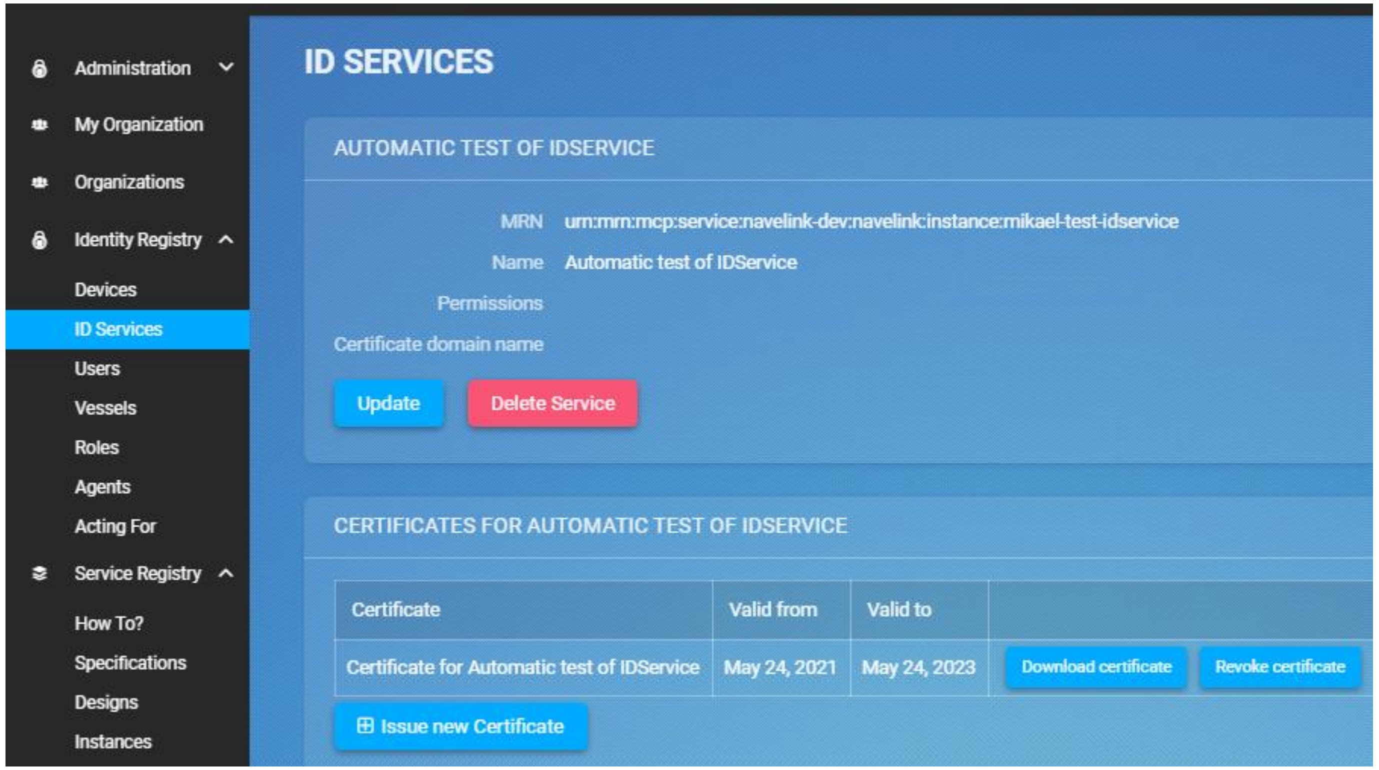
Check Certificate by comparison to downloaded Certificate

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.



ID SERVICES

AUTOMATIC TEST OF IDSERVICE

MRN urn:mrn:mcp:service:navelink-dev:navelink:instance:mikael-test-idservice

Name Automatic test of IDService

Permissions

Certificate domain name

CERTIFICATES FOR AUTOMATIC TEST OF IDSERVICE

Certificate	Valid from	Valid to	
Certificate for Automatic test of IDService	May 24, 2021	May 24, 2023	Download certificate Revoke certificate

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.

Certificate Content

The certificate contains signed data regarding the owner of the certificate, such as MRN identity. If the certificate is attached to a Vessel entity, the certificate also contains signed data regarding IMO and MMSI number of the ship.

The certificate also include the public keys that can be used to verify signature on received data.

Signed data means that data in the Certificate cannot be changed without notice. When checking the certificate, the signature is also checked which contains a checksum of the content. Thus the verification of the Certificate not only checks if it is valid and trusted, it also checks that no information has been changed, data integrity.

For more detailed information of Certificate content in Navelink and MCP, see MCP documentation of Identity Registry.
MCP Documents – Maritime Connectivity Platform