ID: MCP IDsec 1

Version: 2.0

MCC Identity Management and Security: General Approach and Basic Requirements

1 GENERAL

1.1 Summary

The goal of this document is twofold. The first goal is to define the general approach of the Maritime Connectivity Platform (MCP) with respect to identity management and security. The second goal is to define a set of basic requirements for governing and operating instances of the MCP. Both, approach and requirements, build on the analysis, design choices, and experience with the testbed implementations during the EU projects *EfficienSea2* and *STM Validation Project* and the *SMART Navigation* Project funded by the Republic of Korea. The record of this can be found in the previous white paper "Identity Management and Cyber Security" of the MCP [1]. The current state of the testbed can be taken from the MCP Developer's Guide [2].

In the remainder of this section we describe structure, functionality, and governance of the MCP with respect to identity management and security. This is to take into account that the MCP is currently adapting to include governing, integrating and harmonizing several operational MCP instances in addition to providing reference implementations and a testbed. The remainder of this document is then structured as follows. In Section 2 we address Identity Management, in Section 2 we focus on Public Key Infrastructure (PKI), and Section 3 is about Authentication and Authorization for Web Services. Altogether, we derive a first set of requirements for MCP instances, which we collect into a profile in Section 4.

1.2 Structure and Functionality

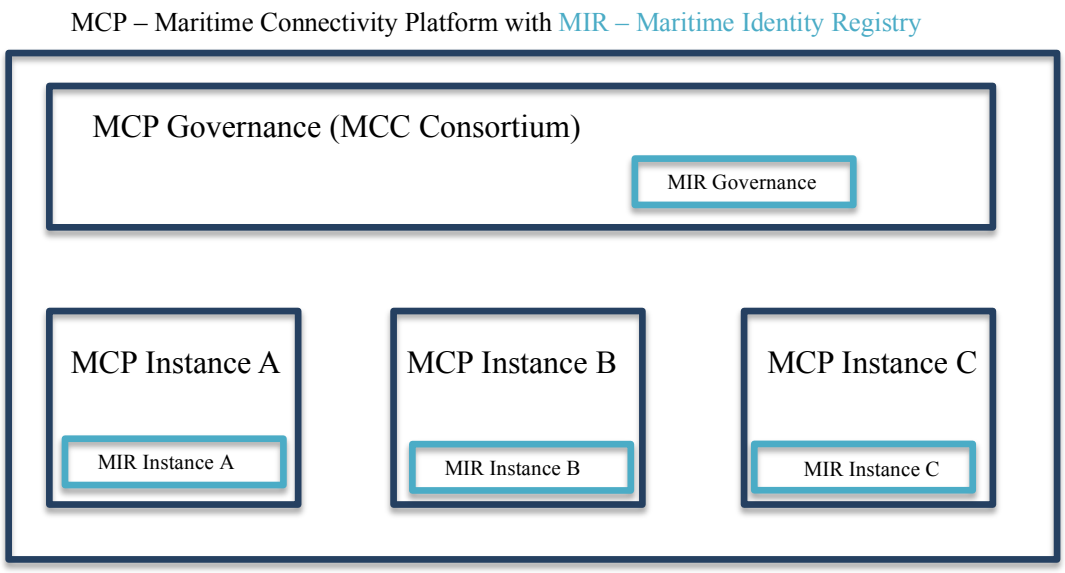

Figure 1: Structure of MIR within MCP

The MCP specifies three core components and their interoperability: the Maritime Identity Registry (MIR), the Maritime Service Registry, and the Maritime Messaging Service. The MIR is responsible for identity management and providing security functionality to the other components. As shown in Fig. 1 the MIR consists of MIR governance and several MIR instances. In summary, MIR governance and instances together typically provide the following functionality:

1. **Identity Management:** The MIR enables that each maritime entity (such as a device, human, organization, service, or ship) can be registered as a participant of the MCP and be equipped with a unique identity. The identity is given in terms of a MRN (Maritime Resource Name). While MIR governance harmonizes the MRN namespace governed by the MCC and sets out criteria for the registration process it is up to the MIR instances to implement and have certified concrete identity registries. We use the following terminology:
    - MCP entity: An entity registered at some MIR instance.
    - MCP namespace: The subspace of the MRN namespace that is governed by the MCC.
2. **Public Key Infrastructure (PKI):** The MIR enables that each MCP entity holds a cryptographic identity in terms of a public/private key pair and a certificate bound to their ID within the MCP. While the cryptographic identity of a MCP entity can change over time (due to updates of key material) the MIR ensures that each MCP entity holds only one *valid* cryptographic identity at any point in time bound to their ID within the MCP. MIR governance provides criteria as to the use and management of cryptographic identities but, similarly to above, it is up to the MIR instances to implement and have certified concrete PKIs.
3. **Authentication and Authorization for Web Services:** The MIR enables that MCP entities benefit from login, single sign-on, and authorization for API access of web services, as well as secure integration of web services based on the widely used standards OAUTH 2.0 and OpenID Connect. To this end MIR governance provides criteria as to interoperability and configurations while the MIR instances deliver concrete OAUTH 2.0/OpenID Connect platforms.

1.3 Governance and Profiles

The focus of the MCP is currently changing from only providing reference implementations and a testbed to including governing several operational MCP instances as well as ensuring their interoperability. At the time of writing there are two emerging operational instances: one is evolving from the STM project; the second is being deployed by the SMART project of the Republic of Korea. Hence, the MCP has to strike a balance between laying down criteria according to which the emerging deployments can be endorsed as MCP instances while remaining open to both, ongoing refinements of the first set of requirements (e.g. with respect to security) as well as new developments and technologies the MCP might wish to utilize (e.g. with respect to distributed PKI). This is why the MIR adopts the following approach of profiles.

The MCP will not develop a single set of criteria that every MIR instance has to comply with but rather allow several *MIR profiles* to coexist. Each MIR profile contains a set of requirements that define what MIR instances have to guarantee to be compliant with the profile. In addition, a profile will typically contain requirements that define what MIR governance is supposed to guarantee (e.g. to maintain operability and overall security). Each MCP instance can choose which of the current MIR profiles it aims to fulfill. While the

MCC is not able to carry out assessments as to whether a MIR instance adheres to a profile itself (in particular with respect to security) it will endorse organizations that can provide this.

Two distinct MIR profiles can either be compatible in that one is a refinement of the other, or they can be non-compatible. To allow non-compatible profiles ensures that the MCP can evolve into different branches. This is to enable that an MCP instance or a cluster of MCP instances may adopt new developments without having to ensure downwards compatibility. As usual downwards compatibility entails the risk of being forced to carry over security vulnerabilities or simply being bogged down by obsolete technology. Therefore the approach of coexisting profiles is also meant to ensure that the MCP can evolve as a whole. The WG IDSec will formulate requirements that will pin down how the profiles are managed and harmonized.

2 IDENTITY MANAGEMENT

2.1 The MCP Namespace

We first describe the MCP namespace. As explained above it is a subspace of the *Maritime Resource Name (MRN)* space [3], which is an official URN namespace. The syntax definitions below use the Augmented Backus-Naur Form as specified in [RFC5234].

The syntax for a MRN is as follows [3]:

```
<MRN> ::= "urn" ":" "mrn" ":" <OID> ":" <OSS>
          [ rq-components ]
          [ "#" f-component ]
<OID> ::= (alphanum) 0*20(alphanum / "-") (alphanum)
<OSS> ::= <OSNID> ":" <OSNS>
<OSNID> ::= (alphanum) 0*32(alphanum / "-") (alphanum)
<OSNS> ::= pchar *(pchar / "/")
```

The rules for alphanum and pchar are defined in [RFC3986].
The optional rq-components and f-component are specified in [RFC8141].

"mrn" specifies that the URN is within the MRN namespace. The *Organization ID (OID)* refers to an organization that is assigned a subspace of MRNs such as IMO, IALA, or the MCP. Syntactically, it is a string that must be unique across the "mrn" scheme. The *Organization Specific String (OSS)* is specified and managed by the governing organization in a consistent way conform to the definitions of the MRN namespace. In particular, each organization must structure the OSS into two parts: the *Organization Specific Namespace ID (OSNID),* and the *Organization Specific Namespace String (OSNS)*. The OSNID identifies a particular type of resource (uniquely within the governing organization), while the OSNS identifies the particular resource (uniquely for its type within the governing organization). Altogether, this ensures that the resulting URN is globally unique.

For a MRN governed by the MCC the OID reads "mcp", and the OSNID specifies one of the following types used within the MCP: device, organization, user, vessel, service, mir, mms, and msr. The latter three types are to be used for entities of the three MCP components MIR, Maritime Messaging Service, and Maritime Service Registry respectively. Moreover, the definition of the OSNS takes into account the distributed structure of the MCP: identities can be provided and managed by several identity providers. In detail, the syntax of a *MRN governed by the MCC* (short: *MCP MRN* or *MCP name*) is as follows:

```
<MCP-MRN> ::= "urn" ":" "mrn" ":" "mcp" ":" <MCP-TYPE> ":" <IPID> ":" <IPSS>
<MCP-TYPE> ::= "device" | "org" | "user" | "vessel" | "service" |
               "mir" | "mms" | "msr"
<IPID> ::= <CountryCode> | (alphanum) 0*20(alphanum / "-") (alphanum)
```

```
<IPSS> ::= pchar *(pchar / "/")
```

"mcp" specifies that the governing organization is the MCC. The next element is *MCP-TYPE*. As explained above this pins down one of the types currently used within the MCP. The *Identity Provider ID (IPID)* refers to a national authority or other kind of organization that acts as an identity provider within the MCP. If the identity provider is a national authority then the IPID must be a country code as defined by ISO 3166-1 alpha-2. Otherwise it will be a string of the same syntax as that for OIDs. The IPID must be unique across the urn:mrn:mcp namespace. The *Identity Provider Specific String (IPSS)* can be defined and managed by the respective identity provider in a way that is consistent and conforms to the definitions of the MRN namespace and requirements laid down by the MCC. In particular, the identity provider must ensure that the IPSS identifies a particular resource uniquely for its type within the domain of the identity provider. Altogether, this will ensure that the resulting URN is globally unique.

Examples:

- urn:mrn:mcp:user:dma:alice - valid MCP MRN for a user, where dma specifies the ID Provider, and the subsequent IPSS string is defined to give the username.
- urn:mrn:iala:aton:gb:sco:6789-1 - valid MRN for a marine aid to navigation (AtoN), where gb stands for United Kingdom, sco for Scotland, and the number is the scottish asset identifier. The example is from [4]. This is *not* a MCP MRN.
- urn:mrn:mcp:device:mirX:aton:gb:sco:6789-1 - valid MCP MRN for the same AtoN, where mirX specifies the ID Provider, and the subsequent IPSS string is defined to first specify the type of the device, and then to follow the country-specific convention of the IALA scheme.

The following requirements pin down that and how the MCP namespace can be managed decentrally.

ID1 The MCC can delegate the assignment of part of the MCP namespace to other organizations that act as identity providers. More concretely, this means that the organization, say X, must hold an IPID, say string "nameofx", and is then responsible for the namespace with the prefix "urn:mrn:mcp:<MCP-TYPE>:nameofx".

ID1.1 The MCC must ensure that each IPID refers to at most one identity provider.

ID1.2 Each Identity Provider must ensure to respect all syntax prescribed in the MRN specification. Moreover, each Identity Provider must ensure that each IPSS of their name space refers to at most one entity of their domain.

ID1.3 The MCC can give recommendations on how to structure the IPSS, e.g. to harmonize the syntax for particular types of entities. These recommendations will not be binding. However, the MCC reserves the right that a particular syntax can be binding with respect to conformance to certain profiles.

Note that ID1.1 and the second part of ID1.2 together ensure uniqueness: one MCP MRN is assigned to at most one entity. This is a general requirement for any URN. ID1.3 allows us to harmonize the IP specific strings while not principally restricting the governance of an IP provider over its namespace.

Example:

> Say there are two ID providers, MIR X and MIR Y. Assume the MCC assigns the IPID "mirx" to MIR X, and "miry" to MIR Y respectively. The MCC must ensure that the strings "mirx" and "miry" are not assigned to any other MIR. MIR X is responsible for the namespace "urn:mrn:mcp:<MCP-TYPE>:mirx:*", and MIR Y is responsible for the namespace "urn:mrn:mcp:<MCP-TYPE>:miry:*" respectively. They might decide to employ the same syntax for the IP specific string, and make this part of a profile they both adhere to. Other ID providers are not bound to use the same syntax. However, if they do not comply to it they cannot be compliant to that profile.

Finally, the following is to ensure a good practice of transparency and interoperability:

ID2 Every Identity Provider shall publish the syntax that describes their name space as well as provide a reference implementation that recognizes the strings of their namespace.

1.3.1 Further Requirements for a Strong Notion of Maritime Identity

The vision of the MCP is to enable a strong concept of digital maritime identity. Hence, we put down requirements that go beyond what is commonly required of URNs. Firstly, we require that every MCP entity must have a name within the MCP namespace. This gives a clear concept of MCP entity: those entities that are registered under an MCP MRN name. Secondly, we require that one MCP entity cannot have several MCP MRNs. For example, this supports law enforcement: When a maritime entity gets discovered and blacklisted for "bad behaviour" (e.g. fake emergency signalling) then it cannot simply revert to another MCP identity and participate as usual.

ID3 Every entity of the MCP shall hold exactly one MCP MRN (i.e. MRN governed by the MCP). This does not exclude that a MCP entity can hold other MRNs, but these must be within namespaces governed by other organizations (e.g. IMO). Also, we will formulate exceptions concerning legacy MRNs within the MCP namespace.

Hence, the AtoN in the example above can be identified by its IALA MRN, or its MCP MRN respectively. However, Requirement ID3 rules out that the AtoN can be referred to by a second MCP MRN. The following requirements implement ID3 in a decentral manner.

ID3.1 Each Identity Provider shall ensure that each entity they register holds at most one MCP MRN within their namespace.

ID3.2 Each holder of a maritime entity shall ensure that this entity is registered with at most one MCP identity provider.

Note that practically it won't be possible to avoid that a "bad player" will seek to register their entity at several different Identity Providers and thereby obtain several MCP identities for it. However, ID3.1 ensures that they can obtain at most as many identities as there exist Identity Providers. And ID3.2 ensures that when it is discovered that an entity holds several MCP MRNs of different providers then it is clear that they have violated a rule (and action can be tied to this).

3 PUBLIC KEY INFRASTRUCTURE (PKI)

In addition to a unique ID in the form of a MCP MRN each MCP entity is provided with a cryptographic identity. This consists of a public/private key pair and a certificate for the public key bound to their ID. In the following, we describe the concept of the PKI that enables this, and a first set of requirements for it. We also identify issues that need to be addressed and refined in the future.

We proceed as follows. In Section 3.1 we explain the MCP core concepts of cryptographic identity and decentral PKI. In Section 3.2 we specify the requirements on cryptographic keys and mechanisms. In Section 3.3 the format of MCP certificates is described. Moreover, in Section 3.4 we show how a service can use an intermediary level of service certificates. For example, this is necessary if a service comes with cryptographic requirements that do not allow the direct use of the MCP ID credentials. Finally, in Section 3.5 we identify further aspects to be considered.

3.1 Core Concepts

1.3.2 Cryptographic Identity

The cryptographic ID of a MCP entity consists of a public/private key pair and a certificate bound to their MRN. The certificate must be issued by the identity provider responsible for the entity. The latter is clearly defined by the IPID string within the MRN of the entity.

Given an entity with MRN A (short: entity A), and its identity provider P, we use the following notation:

- pk_A is the public key of A, and pr_A is the private key of A respectively,
- $cert_P(A, pk_A, V)$ is the certificate of A signed by its identity provider P. The certificate contains the MRN A, the public key of A, and the validity period V of the certificate. (The precise format is provided in Section 3.3.)

The key pair is for use with a digital signature scheme. Hence, each MCP entity A can be verified by another party B to be the originator of a message or other data. As usual this involves the following steps:

1. Entity A signs the message, say M, using its private key pr_A. The result is a cyphertext C.
2. Entity A makes available its certificate $cert_P(A, pk_A, V)$, and transmits the signed message M||C.
3. Entity B obtains the certificate, and receives the signed message.
4. Entity B validates the certificate. As a result B trusts that pk_A is the valid public key of the MCP entity with MRN A. (Necessary requirements on certificate validation will be specified.)
5. Entity B uses pk_A to verify whether the ciphertext C is indeed the digital signature of M. If the verification is successful then B has assurance that M indeed originates from A. (Note that without the fourth step B only has assurance that M originates from the holder of the private key counterpart of pk_A.)

Note that B does not necessarily need to be an MCP entity.

At the time of writing the MCC does not prescribe a policy on how to use ID credentials. They could be used as long-term credentials to obtain short-term credentials for use for a service, or they could be directly used as working credentials. However, the IDSec working group will provide more guidelines on usage soon. Also see Section 3.4 on a related issue.

1.3.3 Decentral PKI

One of the principles of the MCP is to make do without a global notion of trust: in the international context of the MCP we cannot expect that all parties trust each other and each other's security management uniformly. Rather the goal of the MCP is to provide the transparency that enables organizations to decide on whom to trust in which context, and to provide the technical framework to translate such decisions into executable policies. For the PKI we put forward the following three principles:

1. A security breach within the realm of one identity provider's PKI instance shall not enable an attacker to impersonate an entity within the realm (i.e. namespace) of another identity provider.
2. A security breach within an organization or set of organizations predefined by the MCC to carry out some task shall not enable an attacker to impersonate any MCP entity, unless the identity provider of the entity coincides with (one of) the organization(s). In short this means identity providers' PKI instances can always remain secure independently from any central or distributed management by the MCC.
3. It is possible for everyone to obtain assurance as to the security level of any identity provider's PKI instance.

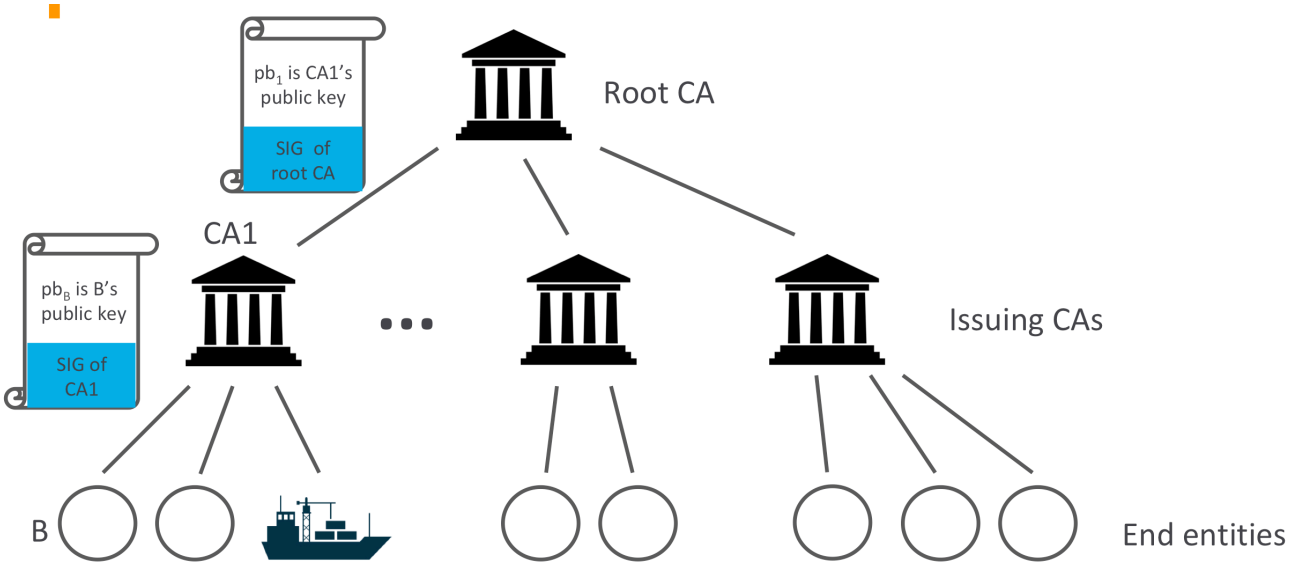

Figure 2: Hierarchical X.509 PKI Structure

The first two principles immediately imply that a classical hierarchical PKI with a root CA hosted by the MCC won't do. We illustrate this by giving examples of impersonation attacks. Assume a hierarchical PKI structure with MCC root CA as shown in Fig. 2. To verify a MCP certificate $cert_P(A, pk_A, V)$ a receiving party has to verify the signature of the identity provider P with the public key of P provided in an intermediary certificate $cert_{MCC}(P, pk_P, V)$ issued by the MCC root CA. Further, to verify the intermediary certificate the receiving party has to verify the signature of the MCC with the public key provided in the MCC root certificate $cert_{MCC}(MCC, pk_{MCC}, V)$. The latter provides the trust anchor accepted by the receiving party.

Examples:

a. *Single point of attack:* Assume the MCC root key pr_{MCC} is compromised. This will allow an attacker to impersonate any MCP entity A. Say P is the identity provider responsible for A. First, the attacker generates a key pair $pk_{I(P)}$, $pr_{I(P)}$ and generates a fake certificate for P with his own key $pk_{I(P)}$: $cert_{MCC}(P, pk_{I(P)}, V)$. This is possible since the attacker knows pr_{MCC}. Second, the attacker generates a key pair $pk_{I(A)}$, $pr_{I(A)}$ and generates a fake certificate for A and his own key $pk_{I(A)}$: $cert_{I(P)}(P, pk_{I(A)}, V)$. This is possible since he knows $pr_{I(P)}$. Altogether, the attacker can now present a valid certificate chain that establishes $pk_{I(A)}$ to be the public key of A while he knows the private counterpart $pr_{I(A)}$. Hence, he can impersonate A. Altogether this violates principle 2 above.

b. *Weakest Link I:* Say the attacker wishes to impersonate entity A of identity provider P. Note that in classic X.509 certificate validation it is only verified that there is a certificate chain up to a trusted root certificate. Say the attacker can easily obtain fake certificates signed by another identity provider P', perhaps, because the attacker is a state actor and P' is under his governance. Then, analogously to above, he only needs to generate his own key pair $pb_{I(A)}$, $pr_{I(A)}$ and generate a fake certificate for A and $pb_{I(A)}$ signed by P': $cert_{P'}(A, pb_{I(A)}, V)$. Since there is no check whether P' is indeed the identity provider of A this gives the attacker a valid certificate chain and corresponding private key, with which he can impersonate A. This violates principle 1 above.

c. *Weakest Link II:* Assume P is an identity provider of low security level, e.g. with a vetting procedure that can easily be undermined. Assume an attacker aims to join the MCP under a false identity so that he is able to inject fake messages without the risk of being traced. The attacker will simply choose P as the identity provider from whom to obtain his false identity. Without principle 3 in place a receiving party has no way to consider the low security level of P when processing the information within the message.

This motivates the following requirements:

PKI1.1 (PKI Structure) There shall be no root CA at the top level of the MCC. Every identity provider that hosts a PKI instance is to provide their own root CA.

PKI1.2. (Validation of IPID) When a receiving party verifies a MCP certificate, say $cert_P(A, pk_A, V)$, it must verify that the certificate is indeed signed by the identity provider responsible for A. The identity provider responsible for A can be read by the receiving party from the IDIP string within the MRN A.

The following requirements ensure that information on root certificates and security levels are made publicly available.

PKI1.3. Every identity provider is to publish their currently valid root certificate in a suitable fashion. For example, this can be made accessible via their web page, or they can commission a generally accepted authority or assurer to do so.

PKI1.4. Every identity provider is to make their security level publicly accessible in a suitable fashion. For example, they can provide a link from their web page to the database of their certification body.

PKI1.5. The MCC will provide a list of identity providers, links to obtain their root certificates, and security levels. However, the MCC won't provide any warranty for this information.

1.3.4 Security Requirements and Profiles

Security requirements to be defined will fall into the following categories:

1. Requirements on vetting. This can be specified similarly to classes such as EV (extended validation).
2. Requirements on certificate revocation
3. Requirements on the validity period of certificates
4. Requirements on security of keys and origin of signing - CA side (including requirements on HSMs)
5. Requirements on security of keys and origin of signing - MCP entity side (including requirements on HSMs)

The requirements will be dependent on the currently emerging profiles:

1. MCP entities generate their ID key pair themselves and in own responsibility and provide this to the responsible CA for certification.
2. The CA (perhaps together with a manufacturer) provisions the initial ID key pair and certificate securely within HSMs (for/within endpoints) to be distributed to the MCP entities.

3.2 Cryptographic Requirements

The cryptographic mechanism approved for ID digital signatures is the Elliptic Curve Digital Signature Algorithm (ECDSA) [FIPS 186-3] with the appropriate hash algorithm from the SHA-2 family [FIPS 180-3]. The approved elliptic curve domain parameters are specified by reference to standardized curves. Currently the following combinations are approved:

ECDSA Key Size (bits)	Hash Algorithm	Elliptic Curve Domain Parameters
384	SHA-384	P-384 [FIPS 186-3] (= secp384r1)
256	SHA-256	P-256 [FIPS 186-3] (= secp256r1)

Note: The reference implementation currently uses SHA-256 for 384 bits key size. This will be changed.

Future extensions:

- Requirements on key pair generation and checks for key pair validity will be given by reference to standards. Also we will check whether there are relevant recommendations in the last version [FIPS 186-5].
- Currently the only approved curve parameters are the NIST recommended curves. It will be checked whether this needs to be extended with regards to cryptographic recommendations of member states' security agencies (e.g. BSI and brainpool curves). Also, if a curve is found to be weak in the future it will be good to have an alternative curve per key size already approved.
- We will also consider matters of crypto agility.

3.3 Certificate Format

We now specify the format of the MCP ID certificates. The format is based on the X.509 standard1. 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** – start/end date and time
- **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 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 MRN					
OU (Organizational Unit)	"user"	"vessel"	"device"	"service"	"mms"	"organization"
E (Email)	User email					Organization email
C (Country)	Organization country code					
UID	Entity MRN					Organization MRN

Example: The following gives an example of the Subject distinguished name field for a vessel with identity provider idp1:

[1] https://www.itu.int/rec/T-REC-X.509

C=DK, O=urn:mrn:mcp:org:idp1:dma, OU=vessel, CN=JENS SØRENSEN, UID=urn:mrn:mcp:vessel:idp1:dma:jens-soerensen

In addition to the information stored in the standard X.509 attributes listed above, the X509v3 extension SubjectAlternativeName (SAN) extension is used to store extra information. There already exists some predefined fields for the SAN extension, but they do not match the need we have for maritime related fields. Therefore the "otherName" field is used, which allows for using an Object Identifier (OID) to define custom fields. The OIDs currently used are not registered at ITU, but are randomly generated using a tool provided by ITU (see http://www.itu.int/en/ITU-T/asn1/Pages/UUID/uuids.aspx). See the table below for the fields defined, the OIDs of the fields and which kind of entities that use the fields.

Field	OID	Used by
Flagstate	2.25.323100633285601570573910217875371967771	Vessels, Services
Callsign	2.25.208070283325144527098121348946972755227	Vessels, Services
IMO number	2.25.291283622413876360871493815653100799259	Vessels, Services
MMSI number	2.25.328433707816814908768060331477217690907	Vessels, Services
AIS shiptype	2.25.107857171638679641902842130101018412315	Vessels, Services
Port of register	2.25.285632790821948647314354670918887798603	Vessels, Services
Ship MRN	2.25.268095117363717005222833833642941669792	Services
MRN	2.25.271477598449775373676560215839310464283	Vessels, Users, Devices, Services, MMS
Permissions	2.25.174437629172304915481663724171734402331	Vessels, Users, Devices, Services, MMS
Subsidiary MRN	2.25.133833610339604538603087183843785923701	Vessels, Users, Devices, Services, MMS
Home MMS URL	2.25.171344478791913547554566856023141401757	Vessels, Users, Devices, Services, MMS
URL	2.25.245076023612240385163414144226581328607	MMS

Encoding of string values in certificates must follow the specifications defined in RFC 5280[2], and where possible it is highly recommended to use UTF-8.

3.4 Service Certificates

Several maritime services come with requirements concerning cryptography and/or certificate formats that might make it impossible to employ MCP ID credentials directly. For example, if an identity provider issues certificates for ECDSA with 384 bits key size this will not meet the real-time requirements and low bandwidth conditions of AIS and VDES [TODO: ref Gareth's paper]. While the service must then provide its own CA the service CA can automatically issue its service certificates based on MCP ID credentials. We provide an

[2] https://tools.ietf.org/html/rfc5280

example of how this can be done based on the concept of *certificate signing requests (CSRs)*, also known as *certification requests*. The most common format for CSRs is defined by the PKCS#10 standard [RFC 2986].

Example: In the following we show the steps carried out by an MCP entity to request a service certificate, and the steps performed by the service CA to issue the certificate respectively. The example follows the implementation of the Haptik CA from the project Haptik[3]. This functionality will also be embedded in a web service and secured by the MCP OpenID Connect/OAuth 2.0 framework.

The MCP entity ...

1. generates a fresh key pair for use with the service
2. builds a X.500 name for use in the service certificate
3. builds a corresponding PKCS#10 CSR
4. signs the CSR with their private MCP ID key
5. sends the CSR together with their MCP ID certificate to the service CA.

On receipt the service CA ...

1. checks whether the CSR is valid
2. builds a X.509v3 certificate according to the CSR and additional information provided by the CA such as issuer, serial number and validity period
3. signs this with their CA private key
4. sends the new certificate to the requesting MCP party.

Note: This pattern is also applicable when the MCP ID keys are mainly used as enrolment keys to obtain shorter lived "working keys".

3.5 Integration of Other PKI Systems

In the spirit of decentrality the PKI shall remain open for PKI systems other than X.509, and also agile for update of certificate formats. Care will be taken to accommodate the necessary flexibility when defining usage of certificates. More to this point is provided by example of the P3KI approach.

4 AUTHENTICATION AND AUTHORIZATION FOR WEB SERVICES

For authentication and authorization in web services the MCP offers users to be able to use OpenID Connect[4] (OIDC). OpenID Connect is a token based authentication protocol that is built as a layer on top of the OAuth 2.0[5] authorization protocol.

A service provider can choose to use the MCP based OpenID Connect authentication for their web service. This means that when a user wants to use the service they must first login with their MCP credentials and receive an OIDC token from MCP that contains information about the user and what organization they belong to. The user can then use this token to authenticate themselves with the web service. The service provider can also choose to implement authorization based on the information in the token.

The table below shows the claims that an MCP OpenID Connect token must contain:

[3] https://haptik.io
[4] https://openid.net/connect/
[5] https://oauth.net/2/

Attribute	Description
preferred_username	The username of the user in the parent organization.
email	The email of the user.
given_name	First name of the user.
family_name	Last name of the user.
name	Full name of the user.
org	The Maritime Resource Name of the organization the user is a member of.
permissions	List of permissions for this user assigned by the organization the user is a member of.
mrn	The Maritime Resource Name of the user.
roles	List of roles that the user has in the role hierarchy.

To learn more about how to use MCP based OIDC we refer to the MCP developer's guideline[6].

Future Tasks:

- Formulate adequate risk containment principles analogously to those for the MCP PKI, and translate them into requirements for MCP OIDC instances. For example, if it was up to a central entity to issue authorization tokens then this central entity constitutes a single point of attack for MCP web authorization (even when authentication itself is delegated to identity providers).
- Specify secure OIDC configurations recommended for use.

5 PROFILE "BASIC REQUIREMENTS"

The profile "Basic Requirements" V1.0 consists of the following requirements:

1. Identity Management:
   - MCP MRN syntax as specified in Section 2.1
   - ID1, ID1.1 - ID1.3: Decentral Management of MCP MRNs
   - ID2: Transparency of Syntax
   - ID3, ID3.1 - ID3.2: Strong Notion of MCP Entity
2. PKI:
   - PKI1.1 - PKI1.5: Decentral PKI Concept
   - The cryptographic requirements as specified in Section 3.2
   - The certificate format as specified in Section 3.3

6 https://developers.maritimeconnectivity.net/identity/index.html#brokered-user-federation

ANNEX A REFERENCES

[1] Identity Management and Cyber Security: White Paper of Maritime Cloud Development Forum, Input Paper to ENAV19

[2] MCP Developers' Guideline: https://developers.maritimeconnectivity.net/identity/index.html

[3] MRN Specification: **https://www.iana.org/assignments/urn-formal/mrn**

[4] G1143 Unique Identifiers for Maritime Resources, IALA Guideline C69-11.2.6, Edition 1.0, June 2019

ANNEX B SPECIFICATION OF SMART MRN SYNTAX

will be revised

KOR MRN namespace and its conversion to MCP MRN which is used in SMART-Navigation Project are given as an example of how an identity provider can utilize their own MRN namespace in the context of MCP. The string “KOR” states *Republic of Korea*, the governance body of KOR MRN and the *IPID* of MCP MRN. The KOR MRN is expected to govern the digital identity of maritime resources and related entities in a national level, enabling the use MCP services developed by the SMART-Navigation Project. In the context of MCP, the Republic of Korea will be an organization entity that provides identities through KOR and MCP MRNs. In conformance to this document, the SMART-Navigation Project uses the MCP MRN for every interaction with MIR by using the “mrn" attribute in the certificate profile and the KOR MRN for national identity management which is stored to the “mrnSubsidiary” attribute, where one-to-one mapping between two namespaces provides. The following description will more focus on the one-to-one mapping of two MRN namespaces rather than explaining the details of each types of entities. The syntax definitions below use the Augmented Backus-Naur Form (ABNF) as specified in [RFC5234].

The syntax for a KOR-MRN used in SMART-Navigation Project is as follows:

<KOR-MRN> ::= "urn" ":" "mrn" ":" "kor" ":" <KOR-TYPE> ":" <ISID> ":" <ISSS>
<KOR-TYPE> ::= "vessel" | "device" | "user" | "service" ":" <SST> | "system" | "mcp"
<SST> ::= “instance" [":" <SIT>] | "specification" | "design"
<SIT> ::= "web" | "app"
<ISID> ::= (alphanum) 0*20(alphanum / "-") (alphanum)
<ISSS> ::= pchar *(pchar / "/")

The *OID* of KOR-MRN is "kor"and the *OSNID* starts from one of the eight types, *KOR-TYPE,* currently used within the KOR context: “vessel", “device", “user”, “service", “system", and “mcp". Note that the absence of the type "*org*" compared to the *MCP-TYPE* indicates the organization is the one and only, Republic of Korea, in the context of SMART Navigation Project. The “service” type has *Service SubType (SST)* as following sub-element which corresponds to the documentation types of the IALA's G1128 e-Navigation technical service specification guideline. The KOR MRN defines *Service Instance Type (SIT)* for the “instance” subtype to specify the target terminal of the service and locates it to the end of the “instance” *SST* as a hierarchy.
The *Identification System ID (ISID)* refers to an external or internal identification system that governs a unique identifier of an entity for its own purpose. The SMART Navigation Project governs and restricts the *ISID* for each type. The *Identification System Specific String (ISSS)* is specified and managed by the governing identification system in a consistent way. Taking both into account an example of a vessel is given as “imo:8814276”, where “imo” and the actual imo number of the vessel “8814276” are represented in *ISID* and *ISSS* respectively, so as to make the vessel’s full KOR MRN to “urn:mrn:kor:vessel:imo:8814276”. The “mcp” type utilizes the *ISID* to indicate the MCP components where the “mms” is only one used in the project for the time of writing.

In order to establish the interoperability SMART Navigation Project uses the *IPSS* of the MCP MRN to build the mapping between the KOR MRN and the MCP MRN. In detail, the syntax of a MCP MRN of the SMART Navigation project, *KOR-MCP-MRN*, is as follows:

<KOR-MCP-MRN> ::= "urn" ":" "mrn" ":" "mcp" ":" <MCP-TYPE> ":" "kor" ":" <KOR-IPSS>
<KOR-IPSS> ::= [<SST> ":" | <DST> ":"] <ISID> ":" <ISSS> | <ISSS>
<SST> ::= "instance" [":" <SIT>] | "specification" | "design"
<SIT> ::= "web" | "app"
<DST> ::= “system"
<ISID> ::= (alphanum) 0*20(alphanum / "-") (alphanum)
<ISSS> ::= pchar *(pchar / "/")

Note that "kor" represents both the *IPID* and an *organization* entity in the MCP type for the sake of reducing the redundancy, i.e., “kor:kor”. Thus the MIR implementation will have the ability to interpret this context as a configurable option. For the KOR MRN types which corresponds to those are in MCP MRN in terms of name, definition, and purpose in use, the *ISID* and the *ISSS* afterward take the place of *IPSS*, namely *KOR-IPSS*. The *KOR-IPSS* can optionally take *Service SubType (SST)* or *Device SubType (DST)* from beginning to represent the same subtypes of the KOR MRN, where a “instance” subtype can have *Service Instance Type (SIT)* at the end in the same manner with the KOR MRN. The *DST* is employed to embrace the “system” of the KOR MRN as the subtype of “device” of the MCP MRN. Please note that the actual use of the *SST* and the *DST* should be constrained to specific MCP types, i.e., the *SST* for services and the *DST* for device, but is formulated here in a simple manner. The identity provider, by restricting the identification systems, should guarantee that the *ISID* does not conflict to the value of either *SST*, *DST*, or *SIT*. The “mcp” type in the KOR MRN is converted by locating the “mcp” to the *OID* of the MCP MRN, the “mms” to the *MCP-TYPE* by meaning of the *ISID*, and *ISSS* at the end which is from the *ISSS* of the KOR MRN for the MMS, e.g., “urn:mrn:mcp:mms:kor:smart001”. As the SMART-Navigation Project proceeds and elaborates the use of MRNs in reality, the presented MRN syntax and its mapping can be changed.