



**Input paper:** XXXX-n.n.n

**Input paper for the following Committee(s):**

- |  |                              |                              |
|--|------------------------------|------------------------------|
| <input type="checkbox"/> ARM             | <input type="checkbox"/> ENG | <input type="checkbox"/> PAP |
| <input checked="" type="checkbox"/> DTEC | <input type="checkbox"/> VTS |                              |

**Purpose of paper:**

- |   |
|---|
| <input type="checkbox"/> Input                  |
| <input checked="" type="checkbox"/> Information |

**Agenda item**

**Technical domain/ Task number**

**Author(s)/Submitter(s)**

DLR, and AIVeNautice

## THE DECENTRAL TRUST SYSTEM OF THE MARITIME CONNECTIVITY PLATFORM

### 1. INTRODUCTION

#### 1.1. SCOPE

This document is intended as an introductory outline to the Maritime Trust System (MTS) and how it will be both designed and implemented. This is intended to be the first draft for a future IALA guideline, thus, so any comments/feedback is welcome.

This document is intended to be a technical specification of MTS and we will refer to [33] for further details on the historical context and background understanding.

#### 1.2. RATIONALE

The Maritime Trust System (MTS) is to be considered a core component of the MCP, extending the authenticated identities as described in IALA G1183 [27].

### 2. SPECIFICATION

The Maritime Trust System (MTS) is a system within the Maritime Connectivity Platform (MCP) for delegating roles/policies to identities within the MCP. Note, that authentication of identities is carried out by the Maritime Identity Registry as described in IALA G1183 [27]. The MTS merely is intended to manage users trust relations and authorise that they have been granted said trust by other identities. Therefore complete checking of access to services is a joint effort between the MIR and the MTS.

The MTS will be transparent to most operations, with actors able to access whatever systems or services they have permission to do so. In order to do so, permissions would need to be granted ahead of time, presumably by the actors employer or owner depending on the nature of the actor.

There are three different aspects of the MTS:

- Trust definition, that being how the trust relations themselves are actually defined
- Trust verification, that being how a trust relation is verified
- Distribution, that being how the different identity and trust certificates are distributed across the system so that each identity can demonstrate the trust placed in it even in an offshore environment

### 3. THE ROLE OF THE MCP CONSORTIUM

MCP will be part of designing guideline specification, but invites everyone to participate in this through the relevant IALA committees.

Specific trust policies will be defined by IALA and other relevant organisations. Trust policies will be defined in the MTS trust policy language, but are otherwise agnostic.

MCP will not be party to enforcing or endorsing operational instances or specific certificates/trust policies.

### 4. MARITIME TRUST SYSTEM

The following will give an overall introduction to the Maritime Trust System (MTS).

#### 4.1. IDENTITIES AND TRUST RELATIONS

Any actor within the MCP must have an associated maritime identity [27]. This will primarily be comprised of a maritime resource name (MRN) [26], which will be issued by some entity within the MCP and accessible within the MIR. That MRN will have an associated X.509 certificate which can be presented to others to authenticate that actor is whom they say, as well as a public/private key-pair. This MRN with an associated X.509 certificate is what will be referred to here as their Identity. Identities themselves contain very little information, only that they relate to a particular actor though they will include the public key. Note, that the private key will need to be kept by an entity in order to sign any trust relations issued in their name.

Relations between Identities are used to express trust, that is the truster delegating a role to the trustee in some certain context. Identities and MRNs should not include any information on what that actor actually is or can do; Identities are only intended to define provide unique cryptographic guarantee of the existence and association to an MRN. It will be the job of the MTS to express roles and policies by the trust relations that an Identity is given. For instance, an engineer may have a logically identical identity to a pilot, but they would each have differing trust relations such as the ability to access an engine room or operate the helm in accordance with their role.

Delegations are defined as a term from one MCP Identity to a MCP MRN. Note that this means anyone can sign any trust relation for any party even without their knowledge. But this will not be meaningful as it would never involve any previously trusted trust anchor.

This is all shown in the Figure 1, with a MIR and two identities shown. Both of the MRN identities are issued by the MIR and so have their certificates signed for by the MIR. There are also two delegations, one from the MIR to MRN1 and one from MRN1 to MRN2. Each are signed by the MRNs private key. Also shown are what each MRN will present in order to demonstrate to another node what trust is placed in it, that being the certificates and signed delegations between the MIR and that MRN. This will be explained in more depth in the following sections.

MTS can be separated into three different abstraction layers, which each has distinct responsibilities.

**Delegation Policy.** In this layer users can define See Section 6 for details.

**Trust Verification.** See Section 7 for details.

**Delegation Distribution.** See Section 6 for details.

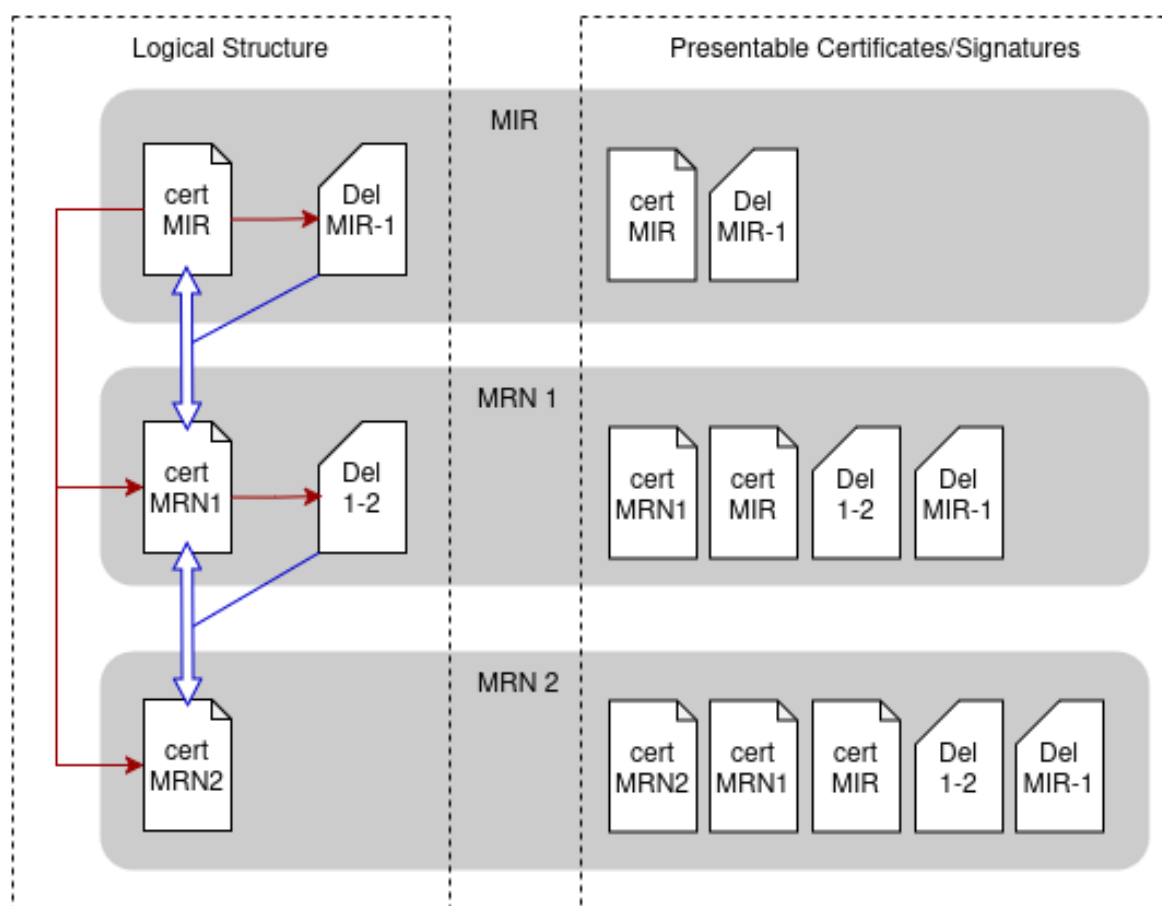


Figure 1: Trust relations and the certificates and signatures backing them up.

## 5. DELEGATION POLICY

### 5.1. SPECIFYING TRUST

Trust relations will be defined using a domain specific language (DSL). This DSL will somewhat resemble High-level Trust Policy Language (HTPL) which in EBNF can be expressed like so:

```

ident  id
tag ::= '<' ident '>'
spec ::= 'language' incls ('{' decls '}')?
incl  ::= tag ('with' along)? (',' incl)?
decls ::= decl *
decl  ::= tag ('::' type)? ('=' strucs)?
strucs ::= struc ('|' strucs)?
struc ::= atom | tag | '{' stree '}' | '*' | '_'
stree ::= struc ('[' stree ']')? ('.' stree)?
along  ::= atom | '{' tree '}' | '*' | '_'
tree  ::= along ('[' tree ']')? ('.' tree)?
type  ::= 'atomic' | 'tdns(' type ')'
```

This language has had some initial investigation into its suitability but may require further refinement before being finalised. This HTPL allows us to define specifications for how relations will be defined and assessed such as shown here:

```

language <op > , <area> (with {eu}) {
  <op> = { <object> . <action>}
  <object> :: atomic =
    gate | door

  <action> = open | close

  <eu_country> = denmark | sweden | norway
  <as_country> = japan | china | india

  <area> :: tdns(atomic)
    = { as . < as_country >}
      | { eu . < eu_country >}
}
```

This specification would allow us to specify permissions for opening or closing gates and doors within the certain countries in Europe or Asia. We can use this both as a format to grant Identities the appropriate permissions (such as an employee permission to open gates in Denmark and Sweden), and to format the tested-against permissions in order to perform such an operation (such as a gate to KU expecting others to have permission to open gates in Denmark in order to grant access).

The actual verification of these permissions can be evaluated by P3KI or equivalent. This allows for the comparison between a querying request, and defined requirement, by computing the net trust between the two.

### 5.2. DEFINING TRUST DELEGATIONS TO DEFINE A TRUST DELEGATION AN ENTITY WILL NEED

to construct a policy, alongside source and target mrns, those being who is trust whom. this delegation is stored as internal data to the MTS but signed by the issuers private key. As all delegations are point to

point, a unique ID for the policy can be constructed out of the mrn by concatenating the signer, source and target. Any preexisting delegation from the same source to the same target signed by the same issuer will be replaced by subsequent delegations between the same identities.

Testing an attempted connection between two actors in the MTS is very rarely expected to be direct from one to another, but are expected to instead be varyingly indirect. For instance, imagine a VTS system that only accepts requests from trusted actors. The VTS would not be expected to trust each permitted actor directly, but would instead trust some intermediate nodes. For instance, rather a VTS may wish to only accept communications from ships that are somehow appropriately registered. Rather than each VTS directly determining which ships are authorised and having a personal relationship to them directly it would be easier to determine which top-level nodes to trust, in this case those being some global trade association(s), of which there are few. At this scale it is still practical for a VTS system to determine which higher authorities it trusts. The VTS may not wish to trust these higher authorities with absolutely everything, but may be willing to trust them with anything related to any registering/owning maritime vessels.

Any global trade associations may in turn grant permissions to smaller shipowners associations, such as those belonging to individual nations or corporate groupings. These permissions would be for shipowners associations to grant permission to owners to register that they own ships. This can be seen in the diagram below along with other parallel relations we will get to momentarily. Each Actor in this chain of relations can delegate either everything trusted to it, or some subset to it. Eventually there is a complete loop of trust between the VTS and any vessel attempting to connect to it, which the VTS can test for to accept a communication.

This loop alone is not very useful, and several loops would presumably be operating in parallel, with a vessel only being accepted if it had numerous relations in combination as shown below. Any MTS must be capable of identifying any relation loops relevant to a particular actor and ensure that they are available to that actor if that actor is present in an offshore node. That is, without all necessary certificates to prove a loop of relation in an offline scenario, any transaction between two actors has the potential to fail even if they should be trusted.

Consider the below diagram showing some abstract relations within an MCP. Some actors A, B and C, are all part of the same blue organisation. A and B are onshore nodes, as shown by the double circles. C is some offshore node such as a ship that is granted permissions by A through node B. Similarly, N and M are some

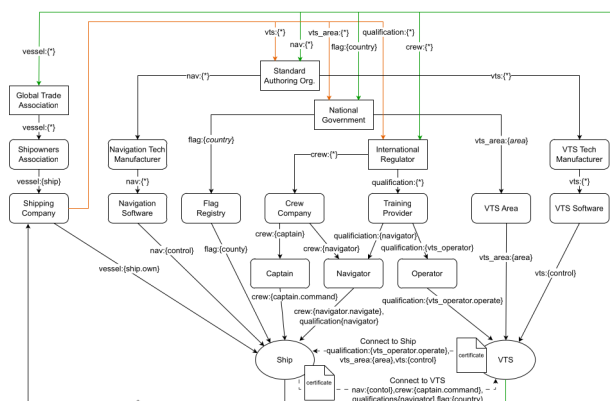


Figure 3: Wider trust relation for a VTS system.

other actors with N being an onshore node and M being offshore. This could be some navigation assistant on a buoy or some such. Let us imagine that C is attempting some operation on M, such as connecting to some service, and it requires some arbitrary permission  $x.y$ .

C and M have never met each other before, and as two actors at two endpoints within the network it is unreasonable to expect them to have an already existing one-to-one relation. However, it is not unreasonable to expect that if actor M has N as a ICA, then M could have already delegated any trust relations  $x$  to it. These relations will not be directly to other end nodes, but to other ICAs. For instance it would be unreasonable for all VTS services (M in this example) to individually trust every ship that could come by, but not unreasonable for each each VTS to delegate their trust to some national/transnational VTS manager (N) that in turn could trust some national/transnational shipping registries (A). As an additional slight complication we've added a third onshore node between A and C, that being B though this will not complicate this current logic.

Crucially, as pre-established relations we have M granting permission  $x$  to N, which in turn grants permission  $x$  to A. A grants the same in turn to B, but B only grants  $x.y$  to C. Now these two sides of the diagram are managed by two separate ICAs. A, B, and C are all have A as a ICA, while M and N have N as a ICA. This means that before C and M try and interact they are part of two separate trust hierarchies. As A, B and C all have A as their ICA, it is trivial for A, B and C to all have local copies of the Identity certificates for A, B and C as CRDTs from C. It is also trivial for each of them to have a copy of the relation certificates for the  $A > B(x)$ , and  $B > C(x.y)$ . Similarly, both M and N can have copies of the M and N Identity certificates, and the  $M > N(x)$ , and  $N > A(x)$  relation certificates. Due to their preexisting relation which implies some potential communication (more on this later) it is also not difficult to envision that A could have a copy of Ns Identity certificate and the  $N > A(x)$  relation certificate, and N could have a copy of As Identity certificate as well.

This means that when C tries to connect to M, C has access to a chain of proof at least from A to C of relation  $x.y$ , and that M has access to a chain of proof from M to A of relation  $x$ . As long as they can verify that they are talking about the same A, they should be able to verify that a chain of proof can be derived from M to C of relation  $x.y$ . Note that it might be that the relation  $N > A(x)$  with both the Identities of N and A being verified as a whole that might be needed as a link, but even in this more stringent scenario it should still be perfectly possible to verify this chain as both sides have access to their relevant proofs as discussed above.

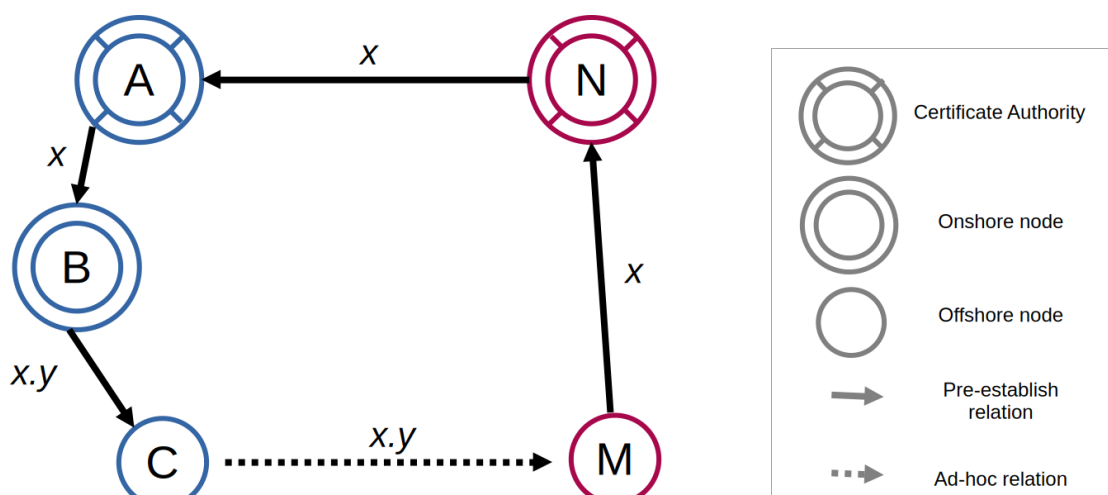


Figure 4: Abstract trust relations.

## 6. DISTRIBUTING DELEGATIONS (INCLUDING IDENTITIES AND RELATIONS)

For any verification of a relation to occur, at least one node will need to verify a chain of relations as discussed in the previous section. In all likelihood, two parties will need to separately verify the chain of relations, those being both the client and server of whatever interaction is taking place. In order to do so they will require at least the certificates for all intervening Identities and relations in the chain.

There are essentially two modes of operation within the MCP, onshore and offshore. Onshore distribution is done using the onshore network, and may be done with both onshore and offshore nodes. This would be when ships are in port, sailors are on land, or equipment is under maintenance and so reliable communication between them and the wider network can occur. Offshore distribution would be between any two nodes that at least one of is partitioned from the wider onshore network. Presumably only offshore nodes could be partitioned in such a way due to weather, unreliable communication, or sufficiently limited communication capacity.

Let us define four types of nodes within the MTS, where a node would be a local processes running at some particular location. Note that we may refer to an MTS instance. This is as many MTSs may be running within the same MCP. An MTS instance would be the MTS managing the delegations for one or more MIR instances.

**Root.** This would be a ‘main’ node, potentially running alongside the MIR but at the very least directly interacting with it. The root acts as the MTS’s link to the MIR. It is expected such a node would only ever be run in a location with reliable, internet based communication such as on land, in a server rack or such like. In practice there may be more than one Root but logically we shall treat it as a single node. The Root will maintain internal data structures such as a record of all delegations from MRNs controlled by its corresponding MIR. The Root and Distribution Nodes will together be referred to as the Distribution Layer.

**Distribution Node.** Mostly identical to the Root, except it does not directly interact with the MIR but delegates that to the Root. This should replicate all of the internal MTS data pertinent to its Root instance. The Distribution Node will act as a distributed access point to the Onshore and Offshore nodes. The

Root and Distribution Nodes will together be referred to as the Distribution Layer.

**Onshore Node.** A node running at a maritime location with reliable communications back to the Distribution Layer, such as a port, coastal communication station, or all but the most remote onshore locations. The Onshore Nodes and Offshore Nodes will together be referred to as the Implementation Layer.

**Offshore Node.** Mostly identical to the Onshore Node except without any permanent expectation of reliable communication, such as ships, buoys etc. It is still expected that there will be some timeframes where reliable communication can be attempted such as when docked in port or under repair, in which case any communication with the wider MTS will be done in this period to get up-to-date certificates. Communication may be attempted at unreliable timeframes, but should not be relied upon. The Onshore Nodes and Offshore Nodes will together be referred to as the Implementation Layer.

## 6.1. DISTRIBUTION LAYER

The Distribution layer acts as a link between implementation layer nodes, ensuring they have as up-to-date certificates as can be managed, and acting as a link between the implementation layer and the MIR(s). An internal record of the identity and delegation certificates used within the MTS instance will be kept and stored on a Merkle-CRDT. This will ensure eventual consistency between nodes whilst ensuring partition tolerance and availability.

A separate graph of delegations will be maintained on a per node basis. This will act as an easily navigable index of how the MRNs and their delegations relate to each other. It is expected that the 'top' of the graph will be the MIRs own MRN, with many delegations being issued by it to the identities it has issued, potentially directly or via a long chain of intervening MRNs. The distribution layer may also need to maintain a list of other MIRs and at what address they may be accessed.

## 6.2. IMPLEMENTATION LAYER AND CERTIFICATES

The implementation layer will not replicate the entire MTS instance data structure but will instead flag to the Distribution Layer which identities its nodes should be sent. It is assumed that a node on the Implementation layer can start with the MRNs for whatever identities it requires, either directly or by being sent them by some managing node. The implementation node will send the MRN to the Distribution layer, which can identify the controlling MIR as the mir name is stored within the MRN. It will then query that MIR for the corresponding ID certificate. The ID certificate is sent to implementation node. An MTS ID is also created within the distribution layer as an entry on the CRDT, with the ID certificate being recorded, along with which implementation nodes have requested it.

## 6.3. DELEGATION SIGNATURES

delegations can be created by the MTS between MRN certificates. They will be signed by an MRNs private key, which could be a 3rd party (e.g. not the source or target of the delegation), though this will only be a meaningful delegation if the signer has been separately trusted with an appropriate delegation as discussed in the Delegation chapter. Note it is the MRN holders responsibility to ensure that they have access to their own private key in order to sign delegations.

When a new delegation is created, the MTS will tie it to the sources MTS ID within the delegation layer. These delegations form edges within an MTS graph. When an implementation layer requests an ID, the graph is traversed to find all delegations to that MRN, along with the MRN certificate that presumably signed those delegations. Several layers of delegations to that MRN up to the MIR MRN are included in any return.



## 6.4. INTER-IMPLEMENTATION NODE COMMUNICATION

Having retrieved the ID certificate for an MRN, the implementation node should have said certificate, and then recursively from that MRN any delegations targeting it and the ID certificate of the delegation source MRN, up to the MIR MRN.

## 6.5. CERTIFICATE/SIGNATURE CACHING

Certificates and signatures are stored within the MTS alongside an identifying tag and hash of that certificate/signature. Communications for status updates between the Implementation and Delegation layers for certificates/signatures will therefore provide a hash of any local copy, so that an update is only sent if the hashes differ. cache entries must be given some time to live.

## 7. TRUST VERIFICATION

The MTS at a particular receiving node should be able to respond to a sending entity that presents its certificates and delegation signatures. Those certificates and delegation signatures should be a chain between the sender and its own trust root, that being its MIR. The receiver should combine the sent certificates and signatures with its own chain of certificates and signatures up to its own trust root, that being its own MIR. Note that in practice these may be the same MIR. The verification is therefore proving that a continuous chain of trust exists from the receiver to the sender. This chain is via delegations from identity to identity, each being signed by some identities. The chain also includes certificates for the identities of each MRN in each delegation, but also any identity certificate signers up to their root MIR, and being previously identified and therefore trusted. The chain will also include the signatures of the delegations from the chain of delegations.

## 8. DEFINITIONS

The definitions of terms used in this Guideline can be found in the International Dictionary of Marine Aids to Navigation (IALA Dictionary) and were checked as correct at the time of going to print. Where conflict arises, the IALA Dictionary should be considered as the authoritative source of definitions used in IALA documents.

## 9. ACRONYMS

## References

- [1] The specification of e-navigation technical services, 2018. <https://www.iala-aism.org/product/g1128-specification-e-navigation-technical-services/>, edition 1.2.
- [2] ISO/TC 22/SC 31. Road vehicles - security certificate management, July 2006. <https://www.iso.org/standard/41891.html>.
- [3] IEC 63173-2:2022. Maritime navigation and radiocommunication equipment and systems - data interfaces - part 2: Secure communication between ship and shore (secom). <https://webstore.iec.ch/publication/64543>.
- [4] D. Atkins, W. Stallings, and P. Zimmermann. PGP message exchange format, August 1996. <https://www.rfc-editor.org/rfc/rfc1991>.
- [5] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 164–173, 1996.
- [6] C. Bormann. Well-known URIs for the websocket protocol. <https://www.rfc-editor.org/rfc/rfc8307>.
- [7] S. Bradner. Key words for use in RFCs to indicate requirement levels. <https://www.rfc-editor.org/rfc/rfc2119>.
- [8] Tim Bray. The JavaScript object notation (JSON) data interchange format. <https://www.rfc-editor.org/rfc/rfc8259>.
- [9] J. Callas, L. Donnerhake, H. Finney, D. Shaw, and R. Thayer. OpenPGP message format, November 2007. <https://www.rfc-editor.org/rfc/rfc4880>.
- [10] David Chadwick, Gansen Zhao, Sassa Otenko, Romain Laborde, Linying Su, and Tuan Anh Nguyen. Permis: a modular authorization infrastructure. *Concurrency and Computation: Practice and Experience*, 20(11):1341–1357, 2008.
- [11] MCP Consortium. Maritime identity registry of the maritime connectivity platform. <https://maritimeconnectivity.net/>.
- [12] MCP Consortium. The mcp concept document – conceptual overview. <https://maritimeconnectivity.net/mcp-documents/>.
- [13] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL). <https://www.rfc-editor.org/rfc/rfc5280>.
- [14] C. Ellison. SPKI requirements, September 1999. <https://www.rfc-editor.org/rfc/rfc2692>.
- [15] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. Spki certificate theory, September 1999. <https://www.rfc-editor.org/rfc/rfcSDSI>.
- [16] S. Farrell and R. Housley. An internet attribute certificate profile for authorization, April 2002. <https://www.rfc-editor.org/rfc/rfc3281>.
- [17] I. Fette and A. Melnikov. The WebSocket Protocol. <https://www.rfc-editor.org/rfc/rfc6455>.
- [18] K. Hamilton-Duffy, R. Grant, and A. Gropper. Use cases and requirements for decentralized identifiers, March 2021. <https://www.w3.org/TR/did-use-cases/>.

- [19] Houshyar Honar Pajoo, Mohammad Rashid, Fakhru Alam, and Serge Demidenko. Multi-layer blockchain-based security architecture for internet of things. *Sensors*, (3), 2021.
- [20] Gregor Jehle. Cross-pki web-of-trust als enabler für zusammenarbeit, 21. September 2023. [https://www.teletrust.de/fileadmin/user\\_upload/07\\_TeleTrust-EBKA\\_PKI-WS\\_Jehle\\_P3KI.pdf](https://www.teletrust.de/fileadmin/user_upload/07_TeleTrust-EBKA_PKI-WS_Jehle_P3KI.pdf).
- [21] Michael Jones. JSON web algorithms (JWA). <https://www.rfc-editor.org/rfc/rfc7518>.
- [22] Michael Jones and J. Bradley. JSON web signature (JWS). <https://www.rfc-editor.org/rfc/rfc7515>.
- [23] Michael Jones, J. Bradley, and N. Sakimura. JSON web token (JWT). <https://www.rfc-editor.org/rfc/rfc7519>.
- [24] P. Leach, M. Mealling, and R. Salz. A Universally Unique Identifier (UUID) URN namespace. <https://www.rfc-editor.org/rfc/rfc4122>.
- [25] Google LLC. Protocol Buffers Documentation. <https://protobuf.dev/>.
- [26] Management of Maritime Resource Name Organization Identifiers. IALA guideline G1164, ed 1.1. <https://www.iala-aism.org/content/uploads/2022/09/G1164-Ed1.1-Management-of-Maritime-Resource-Name-Organisation-Identifiers-December-2021.pdf>.
- [27] IALA Guideline on the Provision of MCP Identities. IALA guideline G1183, ed 1.0, Jun 2024. <https://www.iala-aism.org/>.
- [28] E. Rescorla. The transport layer security (TLS) protocol version 1.3, 2018. <https://www.rfc-editor.org/rfc/rfc8446>.
- [29] Ronald L. Rivest and Butler Lampson. SDSI – a simple distributed security infrastructure. <http://people.csail.mit.edu/rivest/pubs/RL96.ver-1.1.html>, 1996.
- [30] Mohit Sahni. Online Certificate Status Protocol (OCSP) Nonce Extension, 2020. <https://www.rfc-editor.org/info/rfc8954>.
- [31] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 internet public key infrastructure online certificate status protocol - OCSP. <https://www.rfc-editor.org/rfc/rfc6960>.
- [32] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen. Decentralized identifiers (dids) v1.0, July 2022. <https://www.w3.org/TR/did-core/>.
- [33] Michael Kirkedal Thomsen and Gregor Jehle. Whitepaper: The decentral trust system of the maritime connectivity platform. 2025.
- [34] Charikleia Zouridaki, Brian L. Mark, Kris Gaj, and Roshan K. Thomas. Distributed ca-based pki for mobile ad hoc networks using elliptic curve cryptography. In Sokratis K. Katsikas, Stefanos Gritzalis, and Javier López, editors, *Public Key Infrastructure*, pages 232–245, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.