

THE NATIONAL  
MARITIME DOMAIN AWARENESS  
ARCHITECTURE PLAN  
APPENDICES

---

THIS PAGE INTENTIONALLY LEFT BLANK

---

---

# APPENDIX A - IMPLEMENTATION GUIDE

---

---

This implementation guide is copied from the online implementation guide at <http://mise.mda.gov>. Refer to the web site for the latest version.

The Implementation Guide contains the following Sections:

1. [Introduction](#)
2. [Process Flows for Security, Publish/Update, Delete, Search, and Retrieve](#)
3. [Data Mapping](#)
4. [Code Overview](#)
5. [User Stories for Search](#)
6. [Interfacing with the Security Services](#)
7. [Interfacing with the Publication Service](#)
8. [Interfacing with the Delete Service](#)
9. [Interfacing with the Search Service](#)
10. [Interfacing with the Retrieve Service](#)
11. [Testing on the Test Service Platform](#)
12. [Going Live on the Integration Platform](#)

## 1. Introduction

Maritime security is a national priority that depends on the ability to efficiently, effectively, and appropriately share and safeguard information among trusted maritime partners within the Global Maritime Community of Interest (GMCOI). The Maritime Information Sharing Environment (MISE) as defined in the National Maritime Architecture Plan enables secure, standardized sharing of unclassified maritime information among a wide variety of federal, state and local agencies as well as international participants. MISE employs NIEM-M exchange models, representational state transfer (REST) services for publishing/consuming, and attribute-based access control to facilitate information sharing and safeguarding with non-provisioned users in a dynamic environment.

The purpose of this implementation guide is to provide practitioners with guidance and specific examples for interfacing with MISE. Specifically, it shows how to create messages that conform to the [National Information Exchange Model \(NIEM\)](#) Maritime IEPD formats, how to

implement security to successfully access the environment, and how to interface with the services to publish and consume messages from the environment.

For new practitioners it is recommended you start with [Process Flows for Security, Publish/Update, Delete, Search, and Retrieve](#) and proceed in order through the implementation guide.

## 1.1. NIEM-M EXCHANGE MODELS

To learn more about using the NIEM-M exchange models, where they can be downloaded, and how to produce messages to adhere these standards review the section on [Data Mapping](#).

## 1.2. SERVICE INTERFACES

To learn more about the service interfaces to share information via the MISE start with the section on [Process Flows for Security, Publish/Update, Delete, Search, and Retrieve](#), and then visit the sections on [Interfacing with the Publication Service](#), [Interfacing with the Delete Service](#), [Interfacing with the Search Service](#), and [Interfacing with the Retrieve Service](#).

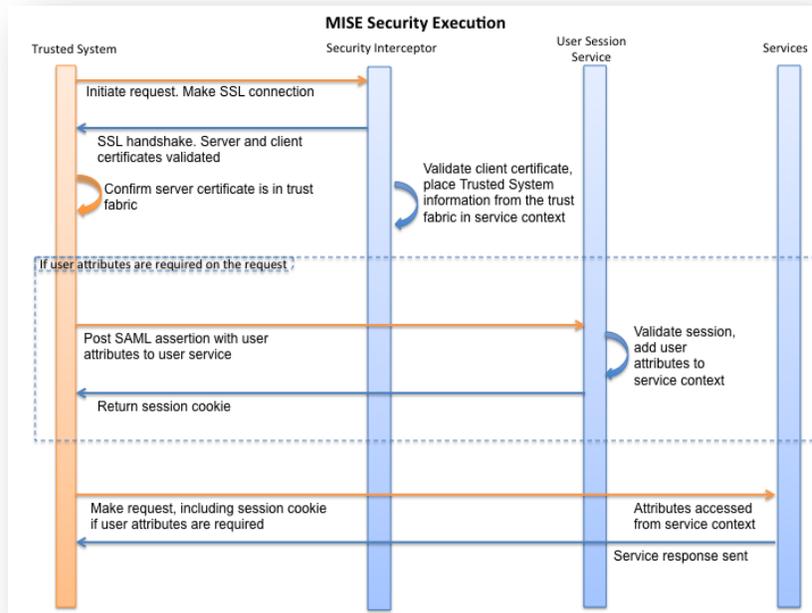
### 1.1. SECURITY SERVICES

To learn more about interfacing with the MISE security services see the sections on [Process Flows for Security, Publish/Update, Delete, Search, and Retrieve](#) and [Interfacing with the Security Services](#).

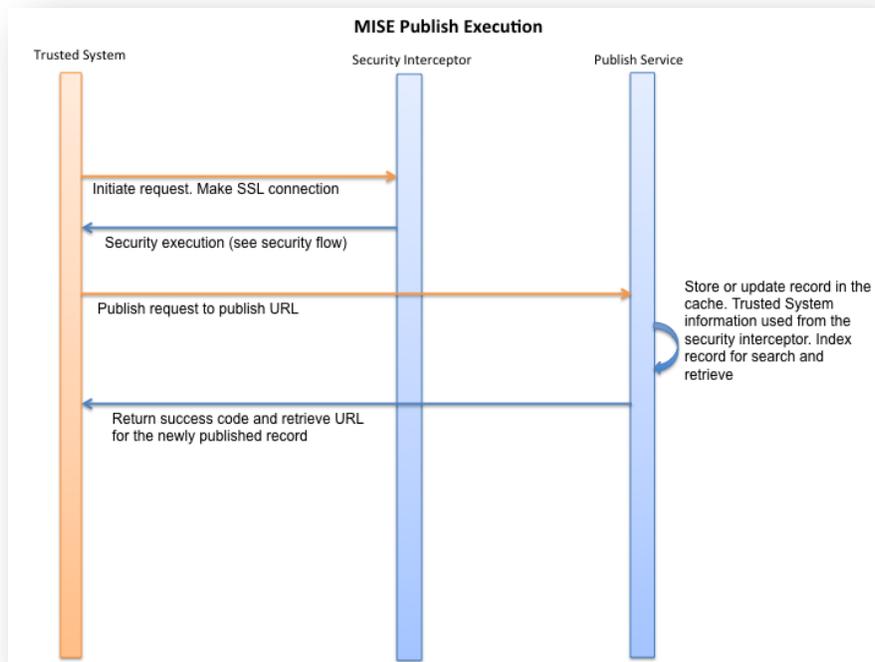
# 2. Process Flows for Security, Publish/Update, Delete, Search, and Retrieve

This section shows graphical representations of the major data provider and data consumer interactions with the MISE services.

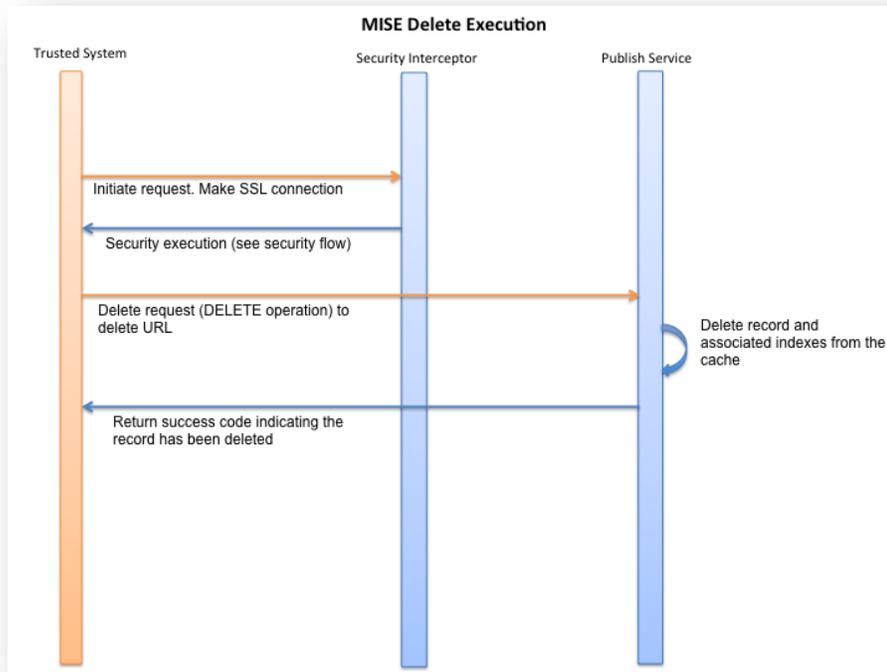
## 2.1. SECURITY



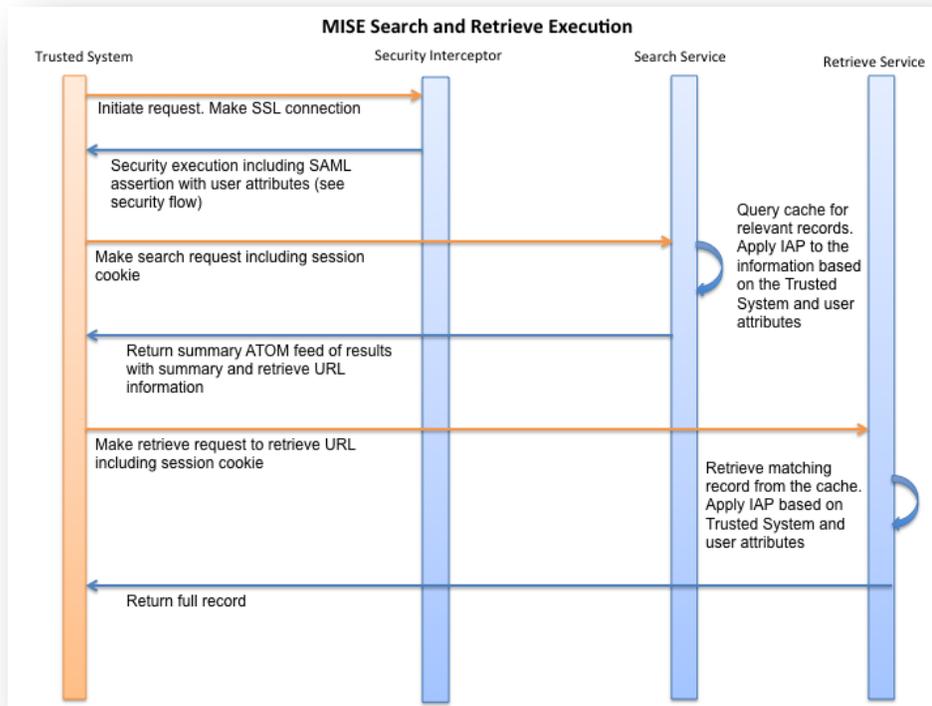
## 2.2. PUBLISH/UPDATE



## 2.3. DELETE



## 2.4. SEARCH AND RETRIEVE



## 3. Data Mapping

### 3.1. OBTAIN THE LATEST NIEM-M MODELS

The NIEM-M Maritime Domain Awareness (MDA) Enterprise Information Exchange Model (EIEM) and Information Exchange Package Documentation (IEPD) are registered and available for download from the [NIEM IEPD Clearinghouse](#) and the DoD Metadata Registry.

Quick links to download the artifacts:

- [Download MDA Enterprise Information Exchange Model \(EIEM\)](#)
- [Download Notices of Arrival IEPD](#)
- [Download Indicators and Notifications IEPD](#)
- [Download Vessel Positions IEPD](#)

### 3.2. HOW TO MAP DATA TO NIEM MARITIME

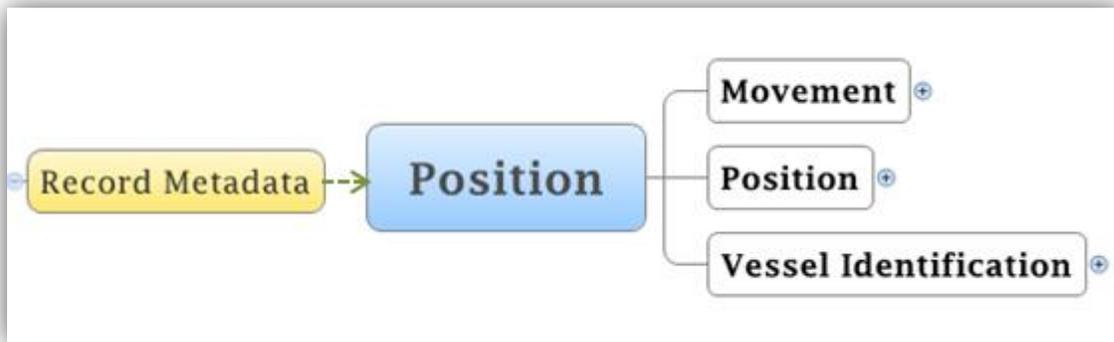
Suppose the following is a native vessel position report format. Data elements include ship's identification, GPS position, course, speed, navigational status, and time-stamp. Sample XML syntax is included below.

```

1 <Track>
2   <DateTime>2011-01-13T13:17:02.325Z</DateTime>
3   <CountryMID>303</CountryMID>
4   <RegionID>27</RegionID>
5   <TrackInfo>
6     <TrackInfoVessel>
7       <VesselId>
8         <MMSI>367354000</MMSI>
9       </VesselId>
10      <VesselAIS>
11        <VesselDataDynamic>
12          <Coordinate>
13            <LAT>53.8782833</LAT>
14            <LON>-166.538633</LON>
15          </Coordinate>
16          <COG>178</COG>
17          <SOG>0.1</SOG>
18          <HDT>228</HDT>
19          <ROT>0</ROT>
20          <NavStatus>Engine</NavStatus>
21          <PosAcc>Low</PosAcc>
22        </VesselDataDynamic>
23      </VesselAIS>
24    </TrackInfoVessel>
25  </TrackInfo>
26 </Track>

```

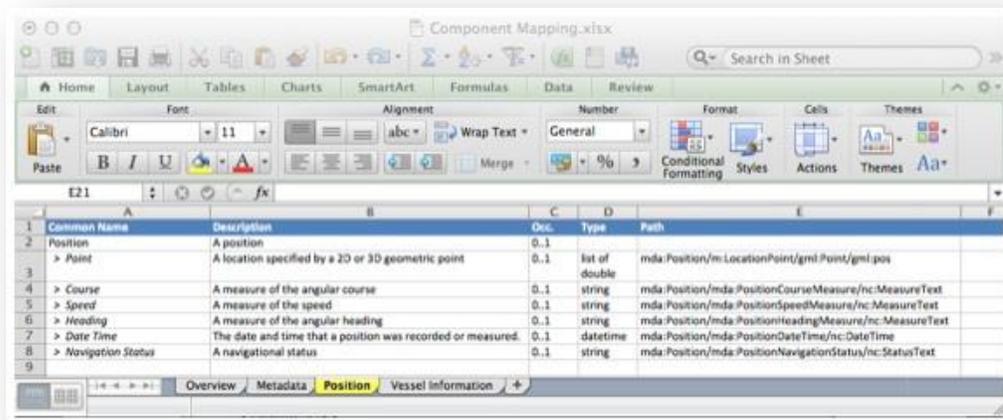
As depicted in the logical diagram for the Position IEPD, Movement, Position, and Vessel Identification are the three primary logical blocks included in a position report. Record Metadata is included in every message type.



The first step is to use the Mapping Spreadsheet from the IEPD. In this example, we are translating a track message into the NIEM format so we will be using the [Position IEPD](#). Find and open the component mapping spreadsheet.

The mapping spreadsheet provides the path for each element in the XML. Each tab corresponds to a "block" in the logical diagram.

An example of the Mapping Spreadsheet is captured below:



Use the Position tab of the Mapping spreadsheet to determine the xml elements to represent the position data in the position message type. For the example message the position is represented as follows:

```

1 <mda:Position>
2   <m:LocationPoint>
3     <gml:Point gml:id="tp1">
4       <gml:pos>53.8782833 -166.538633</gml:pos>
5     </gml:Point>
6   </m:LocationPoint>
7   <mda:PositionSpeedMeasure>
8     <nc:MeasureText>0.1</nc:MeasureText>
9     <nc:SpeedUnitCode>kt</nc:SpeedUnitCode>
10  </mda:PositionSpeedMeasure>
11  <mda:PositionCourseMeasure>
12    <nc:MeasureText>178</nc:MeasureText>
  
```

```

13         <m:AngleUnitText>deg</m:AngleUnitText>
14     </mda:PositionCourseMeasure>
15     <mda:PositionHeadingMeasure>
16         <nc:MeasureText>228</nc:MeasureText>
17         <m:AngleUnitText>deg</m:AngleUnitText>
18     </mda:PositionHeadingMeasure>
19     <mda:PositionNavigationStatus>
20         <nc:StatusText>Engine</nc:StatusText>
21     </mda:PositionNavigationStatus>
22     <mda:PositionDateTime>
23         <nc:DateTime>2011-01-13T13:17:02.325Z</nc:DateTime>
24     </mda:PositionDateTime>
25 </mda:Position>

```

If an element does not map to the NIEM format, it can be included in the expansion text if desired.

To complete the message, we used the Vessel Information tab of the Mapping Spreadsheet to complete translation. Vessel is represented as follows:

```

1 <mda:Vessel>
2     <m:VesselAugmentation>
3         <m:VesselMMSIText>367354000</m:VesselMMSIText>
4     </m:VesselAugmentation>
5 </mda:Vessel>

```

Below is the full NIEM conformant Position Message.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--Sample Position instance corresponding to the Position version 3.2 IEPD -->
3 <posex:Message
4     xsi:schemaLocation="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2">http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2</a>
5     ..\XMLSchemas/exchange/3.2/position-exchange.xsd"
6     xmlns:m="http://niem.gov/niem/domains/maritime/2.1</a>" xmlns:mda="http://niem.gov/niem/domains/maritime/2.1/mda/3.2</a>"
7     xmlns:posex="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2</a>"
8     xmlns:nc="http://niem.gov/niem/niem-core/2.0</a>" xmlns:gml="http://www.opengis.net/gml/3.2</a>"
9     xmlns:ism="urn:us:gov:ic:ism" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance</a>"
10     mda:securityIndicatorText="LEI" mda:releasableNationsCode="USA"
11     mda:releasableIndicator="true">
12     <nc:DocumentCreationDate>
13         <nc:Date>2011-12-01</nc:Date>
14     </nc:DocumentCreationDate>
15     <nc:DocumentExpirationDate>
16         <nc:Date>2012-01-01</nc:Date>
17     </nc:DocumentExpirationDate>
18     <nc:DocumentCreator>
19         <nc:EntityOrganization>
20             <nc:OrganizationName>Example Organization</nc:OrganizationName>
21         </nc:EntityOrganization>
22     </nc:DocumentCreator>

```

```
22 <mda:RecordIDURI>00000001</mda:RecordIDURI>
23 <mda:MessageStatusCode>Initial</mda:MessageStatusCode>
24 <mda:MessageSourceSystemName>Track Source</mda:MessageSourceSystemName>
25 <mda:ICISMMarkings ism:classification="U"
26   ism:ownerProducer="USA" />
27 <mda:Vessel>
28   <m:VesselAugmentation>
29     <m:VesselMMSIText>367354000</m:VesselMMSIText>
30   </m:VesselAugmentation>
31 </mda:Vessel>
32 <mda:Position>
33   <m:LocationPoint>
34     <gml:Point gml:id="tp1">
35       <gml:pos>53.8782833 -166.538633</gml:pos>
36     </gml:Point>
37   </m:LocationPoint>
38   <mda:PositionSpeedMeasure>
39     <nc:MeasureText>0.1</nc:MeasureText>
40     <nc:SpeedUnitCode>kt</nc:SpeedUnitCode>
41   </mda:PositionSpeedMeasure>
42   <mda:PositionCourseMeasure>
43     <nc:MeasureText>178</nc:MeasureText>
44     <m:AngleUnitText>deg</m:AngleUnitText>
45   </mda:PositionCourseMeasure>
46   <mda:PositionHeadingMeasure>
47     <nc:MeasureText>228</nc:MeasureText>
48     <m:AngleUnitText>deg</m:AngleUnitText>
49   </mda:PositionHeadingMeasure>
50   <mda:PositionNavigationStatus>
51     <nc:StatusText>Engine</nc:StatusText>
52   </mda:PositionNavigationStatus>
53   <mda:PositionDateTime>
54     <nc:DateTime>2011-01-13T13:17:02.325Z</nc:DateTime>
55   </mda:PositionDateTime>
56 </mda:Position>
57 </posex:Message>
```

## 4. Code Overview

The MISE implementation team provides a client toolkit for interface to the MISE REST services for Publish, Update, Delete, Search, and Retrieve. All operations are simply REST operations to the correct endpoint.

The client toolkit is primarily designed to make interfacing with the security services easier. The client toolkit is implemented in accordance with the following specifications. More information about each of the specifications can be found in the links below.

- [National MDA Architecture Attribute Specification](#)
- [National MDA Architecture Security Specification](#)
- [National MDA Architecture Publish Specification](#)
- [National MDA Architecture Search/Retrieve Specification](#)

The client toolkit can be downloaded from the MDA Architecture [tools page](#). The tools contain a simple java project that demonstrates how to connect and make a GET request against the ISI services. Two JAR files are included. The first is the MDAUtils JAR, which contains the MDA

Architecture security implementation, the client REST toolkit, and all the necessary dependencies. Additionally, the project also requires the included commons-io JAR, which provides file handling utilities for reading and writing files.

The client toolkit is used in all of the following examples:

- [Interfacing with the Security Services](#)
- [Interfacing with the Publication Service](#)
- [Interfacing with the Delete Service](#)
- [Interfacing with the Search Service](#)
- [Interfacing with the Retrieve Service](#)

Please note that the current base URL for the MISE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

## 5. User Stories for Search

Prior to reading this section, read [Process Flows for Security, Update, Delete, Search, and Retrieve](#) for a basic understanding of how information flows between the provider and consumer.

This set of user stories calls out specific examples for search and retrieve for the information products provided by the MISE. Examples for publish and delete are contained in the sections that discuss those operations, as they are simple, one-time operations.

As noted in the [National MDA Search/Retrieve Specification](#), these operations return the Atom summary feeds of each of the information products. The full message for any of the summaries would be accessed via a retrieve operation.

In each of the examples below, the URL is relative to the mise.mda.gov base path for MISE service access. The placeholder \$value is used in the place of query values.

### 5.1. POSITION

Return vessel position summary messages based on a geospatial area and time window, to see last known positions of all vessels within that geospatial area.	/search/pos?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value&start=\$value&end=\$value
Retrieve a full vessel position message from the URL in a position summary for full details on a specific vessel.	/retrieve/pos?entityid=\$eid&recordid=\$posid
Retrieve the most recent vessel position summary data for each vessel that has updated in the last 30 seconds in the geographic area of interest.	/search/pos?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value&start=\$value&end=\$value start, end should be the last 30 seconds

### 5.2. INDICATORS AND NOTIFICATIONS

Retrieve a summary of IANs about all vessels in a specific	/search/ian?ulat=\$value&ulng=\$value&llat=\$value&llng=
--	--

geospatial area.	\$value &start=\$value&end=\$value
Retrieve a summary message for vessels with a specific threat level within a geospatial area. Note that \$threat in the following example must align with one of the threat values available in the IAN schema.	/search/ian?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value&\$threat=\$value

### 5.3. NOTICE OF ARRIVAL

Retrieve a summary message of all vessels inbound to a specified port.	/search/noa?PortCodeText=\$value&start=\$value&end=\$value
Retrieve a summary message of all pending notices of arrival for a specified vessel.	/search/noa?VesselNameText=\$value&VesselMMSIText=\$value &VesselIMONumberText=\$value
Retrieve a summary of IANs of all vessels in a specific geospatial area that are carrying certain dangerous cargo (CDC).	/search/ian?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value &start=\$value&end=\$value&VesselCDCCargoOnboardIndicator=true
Retrieve a summary message based on a specified vessel and port for which any data element has been updated in the last 10 minutes.	/search/noa?start=\$value&end=\$value&PortCodeText=\$value&VesselNameText=\$value &VesselMMSIText=\$value&VesselIMONumberText=\$value start, end should be the last 10 minutes

\*Note that any combination of MMSI, IMO, and Name can be used, all three are not required.

### 5.4. LEVELS OF AWARENESS

Retrieve a summary message for all vessels in a specified geospatial area.	/search/loa?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value&start=\$value&end=\$value
Retrieve a summary message for vessels with a specific threat level within a geospatial area. Note that \$threat in the following example must align with one of the threat values available in the LOA schema.	/search/loa?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value&\$threat=\$value
Retrieve a summary of a specific vessel in a geospatial area.	/search/loa?ulat=\$value&ulng=\$value&llat=\$value&llng=\$value &VesselNameText=\$value&VesselMMSIText=\$value&VesselIMONumberText=\$value

\*Note that any combination of MMSI, IMO, and Name can be used, all three are not required.

## 6. Interfacing with the Security Services

All interactions to publish and consume data within the MISE are secured interactions over SSL between trusted systems. As a prerequisite to understanding the security implementation examples in this section, it is highly recommend you first read the following documents:

- [National MDA Architecture Plan](#) for an overview of the MISE security approach.
- [National MDA Architecture Security Specification](#) for the details of how trusted systems securely connect to the ISI.
- [National MDA Architecture Attribute Specification](#) for an explanation of the common attributes used for entitlement management.

## 6.1. OBTAINING X.509 CERTIFICATES

Numerous tools and processes are available for creating key pairs and X.509 certificates. The exact process chosen by a trusted system will vary depending on the platform the trusted system implementation is based upon, agency procedures, and the chosen root CA.

In some cases a trusted system may need to generate a keypair and a certificate signing request (CSR) internally using a tool such as OpenSSL or Java's keytool, and submit the CSR to a root CA for signing. The following sections provide steps for generating the private key and public Certificate Signing Request (CSR).

## 6.2. USING OPENSSL TO GENERATE A PRIVATE KEY AND PUBLIC CERTIFICATE SIGNING REQUEST (CSR)

Issue the following command to create private key and CSR

```
openssl req -new -nodes -keyout myserver.key -out server.csr -newkey rsa:2048
```

This creates two files. The file myserver.key contains a private key; do not disclose this file to anyone. Carefully protect the private key. In particular, be sure to back up the private key, as there is no means to recover it should it be lost. The private key is used as input in the command to generate a Certificate Signing Request (CSR).

1. You will now be asked to enter details to be entered into your CSR. What you are about to enter is what is called a Distinguished Name or a DN. Use the FQDN as Common Name (CN). The fields email address, optional company name and challenge password can be left blank for your SSL certificate.

```
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:San Diego
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Agency One
Organizational Unit Name (eg, section) []:IT
Common Name (eg, YOUR name) []:agencyone.gov
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
-----
```

2. Your CSR has been created. You can open the server.csr in a text editor to view it.
3. Follow the CA-specific instructions for submitting the CSR to the CA to generate your SSL certificate.

## 6.3. USE JAVA'S KEYTOOL TO GENERATE A PRIVATE KEY AND PUBLIC CERTIFICATE SIGNING REQUEST (CSR)

1. Using the java keytool command line utility, the first thing you need to do is create a key store and generate the key pair. Do this with the following command:

```
keytool -genkey -keysize 2048 -keyalg RSA -alias agencyone -keystore
mykeystore
```

1. Tip: The 2048 in the command above is the key bit length. MISE requires a key bit length of 2048.
2. You will be prompted for a password for the key store. The key password must be at least 6 characters long.
3. Tip: Make a note of the password. If lost it cannot be retrieved.
4. You will be asked for several pieces of info which will be used by the MISE Test Certificate Authority to create your new SSL certificate. When it asks for your first and last name, make sure you enter the FQDN of your server that will make the connection to the MISE. Here is an example:

```
What is your first and last name?
[Unknown]: http://agencyone.mda.gov
What is the name of your organizational unit?
[Unknown]: IT
What is the name of your organization?
[Unknown]: Agency One
What is the name of your City or Locality?
[Unknown]: San Diego
What is the name of your State or Province?
[Unknown]: California
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=http://agencyone.mda.gov, OU=IT, O=Agency One, L=San Diego,
ST=California, C=US correct?
[no]: yes
```

5. You will be prompted for a password for the private key within the key store. If you press enter at the prompt, the key password is set to the same password as that used for the key store from the previous step. The key password must be at least 6 characters long.
6. Tip: Make a note of the passwords. If lost they cannot be retrieved.
7. Now generate the Certificate Signing Request (CSR) from the private key generated above using the following command:

```
keytool -certreq -alias agencyone -file agencyone.gov.csr -keystore mykeystore
This creates a CSR and stores it in a file named agencyone.gov.csr.
```

8. Follow the CA-specific instructions for submitting the CSR to the CA to generate your SSL certificate.

Below is an example of what your CSR will look like. This is an example only and cannot be used to generate your SSL certificate.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICtTCCAZOCAQAwcDELMAkGA1UEBhMCMVVMxEZARBgNVBAgTCmNhbGlmb3JuaWEeEjAQBgNVBAcT
CXNhbWVnbzETMBEGA1UEChMKYVdlbWlbnN5IG9uZTELMaKGA1UECzMCSVQxRjAUBgNVBAMTDWFn
ZW5jeW9uZS5nb3YwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCwjMqyx0R2OqhlTjXJ
4tH7275YW01+6K6kLOc/yFmfDLK8oSynDoTq4PzO2z1BAHq/8CgkPTs/tHgNphWTlw0c5WpaR487
bVh0dyJwvz3EI6hLmPAYqTvAB2C2aW0zcLzGTSxR8rAhHoX7oOgA3E9xKmoYVMVIZLfn63Tn/F6M
T5NdFdTbkoRzcxpkVmh6o60Vv6jGTI+zUpdyC7W8QRm/kshQgtjXLeYLACXuLvaKzn69p1TLcss
knRCsOsLjHhJrBmPK3upD9HRZ2bbe+Pp1/QUUVztiHDhnwPyEV6Iq2ZF0AuIF4otQU05DLQMsI28
EPu52GmfDMkPuXZFmYYDAgMBAAGgADANBgkqhkiG9w0BAQUFAAOCAQEAJTWTApJiggCgSxE48+Wi
ATNHe3rHJYPLzFmTRupM0tReLQWA+246g+ZGFH0wRv2VO90mMW/MivoxAnoyyP5J708MNsHo1LmN
```

```
bW9kyUuZK22T0lpO3t6BDDix8NWLg5cEGm08sI20iptesemlKq4E/2TxFdEmLpiARD9768WGVtbX
8EB08V2U78A8s5hTMly7hnaywfOm4ezpW1lktU1EuzVxGLHkBj7H5CEKpjH02/AZRNYJRYWrzcdO
YS/gUqs/cvqL77QrwoXWjrCEjSKYtibaXNlSbjEnDKbkoKJl0UsKRLAhMs8NI/HvalV1o8J8/ftc
1J1xTgHFYyxRJluV6w==
-----END NEW CERTIFICATE REQUEST-----
```

## 6.4. REGISTRATION OF TRUSTED SYSTEM IN TRUST FABRIC

Once the necessary X.509 certificate is obtained, your trusted system must be registered in the trust fabric document by the MISE Management team. You will have an entry in the trust fabric for each role for which your system is authorized, i.e. data provider and/or data consumer.

Following is an example entry for a provider trusted system:

```
1 <md:EntityDescriptor entityID="<a
  href="https://mise.agencythree.gov/">https://mise.agencythree.gov/</a>">
2   <md:RoleDescriptor
  protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
3     xsi:type="mise:MISEProviderDescriptorType">
4     <md:KeyDescriptor use="signing">
5       <ds:KeyInfo xmlns:ds="<a
  href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>">
6         <ds:X509Data>
7           <ds:X509Certificate>
8             <!-- Base 64 encoded certificate embedded here
9              This is the client certificate which the trusted
10             system will present during SSL connection handshake.
11             The private key matching this certificate will also
12             be used by this trusted system for signing SAML
13             assertions.
14             -->
15           </ds:X509Certificate>
16         </ds:X509Data>
17       </ds:KeyInfo>
18     </md:KeyDescriptor>
19   </md:RoleDescriptor>
20   <md:ContactPerson contactType="technical">
21     <md:Company>Trusted Federal Systems, Inc.</md:Company>
22     <md:GivenName>Eric</md:GivenName>
23     <md:SurName>Jakstadt</md:SurName>
24     <md:EmailAddress><a
  href="mailto:eric.jakstadt@trustedfederal.com">eric.jakstadt@trustedfederal.com</a
  ></md:EmailAddress>
25     <md:TelephoneNumber>404-806-8143</md:TelephoneNumber>
26   </md:ContactPerson>
27 </md:EntityDescriptor>
```

Now an example entry for a consumer trusted system. Notice in the consuming system entry in the trust fabric, the trusted system is assigned the appropriate indicator attributes used to make authorization decisions on queries.

```
1 <md:EntityDescriptor entityID="<a
  href="https://mise.agencyone.gov/">https://mise.agencyone.gov/</a>">
2   <md:Extensions>
3     <gfipm:EntityAttribute FriendlyName="COIIndicator"
4       Name="mise:1.2:entity:COIIndicator"
5       NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
6       <gfipm:EntityAttributeValue
7         xsi:type="xs:string">True</gfipm:EntityAttributeValue>
8     </gfipm:EntityAttribute>
9     <gfipm:EntityAttribute FriendlyName="LawEnforcementIndicator"
```

```

8         Name="mise:1.2:entity:LawEnforcementIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
9         <gfipm:EntityAttributeValue
xsi:type="xs:string">True</gfipm:EntityAttributeValue>
10        </gfipm:EntityAttribute>
11        <gfipm:EntityAttribute FriendlyName="PrivacyProtectedIndicator"
12        Name="mise:1.2:entity:PrivacyProtectedIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
13        <gfipm:EntityAttributeValue
xsi:type="xs:string">True</gfipm:EntityAttributeValue>
14        </gfipm:EntityAttribute>
15        <gfipm:EntityAttribute FriendlyName="OwnerAgencyCountryCode"
16        Name="mise:1.2:entity:OwnerAgencyCountryCode"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
17        <gfipm:EntityAttributeValue
xsi:type="xs:string">USA</gfipm:EntityAttributeValue>
18        </gfipm:EntityAttribute>
19    </md:Extensions>
20    <md:RoleDescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
21        xsi:type="mise:MISEConsumerDescriptorType">
22        <md:KeyDescriptor use="signing">
23            <ds:KeyInfo xmlns:ds="a
href="http://www.w3.org/2000/09/xmldsig#"></a>">
24                <ds:X509Data>
25                    <ds:X509Certificate>
26                        <!-- Base 64 encoded certificate embedded here
27                        This is the client certificate which the trusted
28                        system will present during SSL connection
handshake.
29                        The private key matching this certificate will
also
30                        be used by this trusted system for signing SAML
31                        assertions.
32                        -->
33                    </ds:X509Certificate>
34                </ds:X509Data>
35            </ds:KeyInfo>
36        </md:KeyDescriptor>
37    </md:RoleDescriptor>
38    <md:Organization>
39        <md:OrganizationName xml:lang="en">Agency One</md:OrganizationName>
40        <md:OrganizationDisplayName xml:lang="en">Agency
41        One</md:OrganizationDisplayName>
42        <md:OrganizationURL xml:lang="en"><a
href="http://www.agencyone.gov"></md:OrganizationURL"></a></md
:OrganizationURL</a>>
43        </md:Organization>
44        <md:ContactPerson contactType="technical">
45            <md:Company>Trusted Federal Systems, Inc.</md:Company>
46            <md:GivenName>Eric</md:GivenName>
47            <md:SurName>Jakstadt</md:SurName>
48            <md:EmailAddress><a
href="mailto:eric.jakstadt@trustedfederal.com">eric.jakstadt@trustedfederal.com</a
></md:EmailAddress>
49            <md:TelephoneNumber>404-806-8143</md:TelephoneNumber>
50        </md:ContactPerson>
51    </md:EntityDescriptor>

```

## 6.5. DOWNLOAD THE TRUST FABRIC DOCUMENT

As discussed in the [Security Specification](#), the trust fabric contains public keys for all trusted systems that interact with the MISE. The trust fabric endpoint requires HTTPS but does not require a client certificate or any other method of authentication.

Retrieve the trust fabric document by any standard means, including viewing in any browser, at the MISE server at `/miseresources/TrustFabric.xml`

The trust fabric document for the MISE test environment is available at:  
<https://107.23.66.168:9443/miseresources/TrustFabric.xml>

## 6.6. VALIDATE THE TRUST FABRIC SIGNATURE PROGRAMMATICALLY

The following sample code (written in Java) taken from [client toolkit](#) demonstrates validating of the signed trust fabric document. Since the trust fabric document is a SAML metadata file with a few simple extensions, this sample code is able to leverage the open source OpenSAML project to simplify implementation. Trusted system implementations not written in Java, or which already include other SAML implementations, may also be able to simplify implementation by relying on existing SAML metadata implementations.

The following code snippet shows how the trust fabric document may be loaded into a DOM object so the signing certificate can be parsed and the signature on the document validated.

```

1 // read in the trust fabric from a local file location
2   FileInputStream fis = new FileInputStream("/local/path/TrustFabric.xml");
3   m_domFactory = DocumentBuilderFactory.newInstance();
4   m_domFactory.setNamespaceAware(true);
5   Element domElement =
6     m_domFactory.newDocumentBuilder().parse(fis).getDocumentElement();
7 // cryptographic validation of signature
8 X509Certificate signedByCert = verifyXMLSignature(domElement);
9 System.out.println(String.format("Signature validation %s", signedByCert == null ?
10 "FAILED" : "SUCCEEDED"));

```

The following snippet takes the trust fabric as a DOM object and returns the signing certificate if it is valid.

```

1 public static X509Certificate verifyXMLSignature(Element target) throws Exception {
2   // Validate the signature -- i.e. SAML object is pristine:
3   NodeList nl = target.getElementsByTagNameNS(XMLSignature.XMLNS, "Signature");
4   if (nl.getLength() == 0)
5     return null;
6
7   KeyValueKeySelector kvs = new KeyValueKeySelector();
8   DOMValidateContext context = new DOMValidateContext(kvs, nl.item(0));
9
10  // Create a DOM XMLSignatureFactory that will be used to unmarshal the
11  // document containing the XMLSignature
12  String providerName = System.getProperty("jsr105Provider",
13  "org.jcp.xml.dsig.internal.dom.XMLDSigRI");
14  XMLSignatureFactory fac = XMLSignatureFactory.getInstance("DOM", (Provider)
15  Class.forName(providerName).newInstance());
16  DOMXMLSignature signature = (DOMXMLSignature)
17  fac.unmarshalXMLSignature(context);
18  if (!signature.validate(context))

```

```

17         return null;
18
19         return kvs.getUsedCertificate();
20     }
    
```

## 6.7. IMPLEMENTING MISE SECURITY ATTRIBUTES

As detailed in the [National MDA Attribute Specification](#), entitlement management within the MISE relies on the use of a common set of entity, user, and data attributes to make run-time authorization decisions as to whether a trusted system and requesting user are authorized to access a requested information resource.

There are three categories for attributes defined for the National MDA Architecture:

1. Entity Attributes: Attributes that pertain to a trusted system within the MISE.
2. User Attributes: Attributes that pertain to a human user.
3. Data Attributes: Attributes that pertain to data.

Currently data is grouped by LE sensitive (LEI), privacy protected (PPI) and the rest of the community (COI). The security indicators defined in the attribute specification map to these groups, i.e. Law Enforcement Indicator, Privacy Protected Indicator, and COI Indicator. Additionally, there is a one-to-one relationship between the security indicators assigned to data (data attributes) by information providers to convey sharing restrictions and the indicators assigned to information consumer trusted systems (entity attributes) and users (user attributes) to convey their respective privileges.

## 6.8. APPLYING DATA ATTRIBUTES ON PUBLISH

As a Data Provider Trusted System you must tag messages before publishing to the ISI with metadata to convey any restrictions on the data.

Attribute Name	Possible Values	Description
SecurityIndicatorText	“LEI”   “PCII”   “PPI”   “SBU”   “FSLT”   “PSO”   “COI”	Indicates the level of access required to access the data. LEI for Law Enforcement Sensitive Information, PPI for Privacy Protected Information, Protected Critical Infrastructure Information, Sensitive but Unclassified, Federal State Local and Tribal, Private Sector Only or COI for the rest of the community.
ReleasableIndicator	“true”   “false”	Marks data as releasable to the public domain under the restrictions of the associated security indicator.
ReleasableNationsCode	Space-delimited list of 3-letter country codes, ex. “CAN USA FRA”	Indicates data can only be released to those nations identified by the country codes. Default value is “USA”.

For example to publish a vessel position messages that is law enforcement sensitive, not publically releasable, and shareable with only US, these fields will be added to the exchange element of the publish message as depicted in the example below.

```

1 <message
  xmlns:posex="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2"
  mda:securityindicatortext="LEI" mda:releasablenationscode="USA"
  mda:releasableindicator="false"></message>
    
```

## 6.9. SUPPLYING USER/ENTITY ATTRIBUTES FOR SEARCH/RETRIEVE

### 6.9.1. MAP LOCAL USER PRIVILEGES TO MISE SECURITY ATTRIBUTES

Use the MISE security attributes as defined in the [Attribute Specification](#) to assert citizenship and the access level for the user associated with a query. Citizenship is conveyed using the CitizenshipCode attribute, `mise:1.4:user:CitizenshipCode`, with a value equal to the ISO 3-letter country code.

The access level is conveyed using one or more attributes defined as indicators.

Indicator	Attribute Name	Description
Law Enforcement Indicator	<code>mise:1.4:user:LawEnforcementIndicator</code>	User requires and qualifies for access to law enforcement information in accordance with all appropriate statutes and legislation.
Privacy Protected Indicator	<code>mise:1.4:user:PrivacyProtectedIndicator</code>	User requires and qualifies for access to privacy protected information in accordance with all appropriate statutes and legislation.
Community of Interest Indicator	<code>mise:1.4:user:COIIndicator</code>	Minimum access level assigned to user that requires access to information shared by the MISE community.

### 6.9.2. FORMING SAML USER ASSERTION

The following code snippet provides an example of building the user assertions and adding them to the context of the request. The full example is shown in the section on [Interfacing with the Search Services](#).

```

1 //Form the user assertion
2     String assertingPartyID = "test.client";
3     AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
4     builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION,
5 10*60); // valid for 10 minutes
6     builder.addAttribute("ElectronicIdentityId",
7 "gfipm:2.0:user:ElectronicIdentityId", "<a
8 href='mailto:testuser@testsystem.gov'>testuser@testsystem.gov</a>");
9     builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T.
10 User");
11
12     Attribute attr = builder.addAttribute("CitizenshipCode",
13 "mise:1.4:user:CitizenshipCode", "USA");
14     builder.addAttribute("LawEnforcementIndicator",
15 "mise:1.4:user:LawEnforcementIndicator", "true");
16     builder.addAttribute("PrivacyProtectedIndicator",
17 "mise:1.4:user:PrivacyProtectedIndicator", "true");
18
19     builder.signUsingPkcs12(assertingPartyID,
20 FilenameUtils.separatorsToSystem(keystorePath), keystorePass);

```

```
13         Assertion assertion = builder.getAssertion();
14
15         //Important to not use SAMLUtils.asPrettyXMLString(object) as it will
cause the signature validation to fail
16         HttpResponse response = m_client.post("/MDAUserSessionService/login",
null, SAMLUtils.asXMLString(assertion),
```

## 7. Interfacing with the Publication Service

The publication service provides the interface to publish information products to the MISE. The complete interface description for publish is in the [Code Overview](#) page for the code download and library details.

This example walks through publishing a single Position instance to the position interface, assuming that the publishing system can already interface with the [security services](#), including registering the publishing system as a trusted system in the Trust Fabric.

The instance file that might be published via this operation can be downloaded [here](#). This file contains a NIEM-M Position instance containing three position reports for the MV Example.

The actual XML for a publish operation would normally be assembled from a database or some other storage location. The example below just reads the XML from a file.

```
1 package test;
2
3 import gov.mda.trustfabric.TrustFabric;
4 import gov.mda.util.RestServiceClient;
5
6 import java.io.File;
7 import java.io.FileInputStream;
8 import java.security.cert.CertificateFactory;
9 import java.security.cert.X509Certificate;
10
11 import javax.xml.parsers.DocumentBuilderFactory;
12 import javax.xml.xpath.XPathFactory;
13
14 import org.apache.commons.io.FileUtils;
15 import org.apache.commons.io.FilenameUtils;
16 import org.apache.http.HttpResponse;
17 import org.apache.http.entity.ContentType;
18
19 public class TestPublishClient {
20
21     /**
22      * @param args
23      */
24     public static void main(String[] args) {
25         RestServiceClient m_client;
26
27
28         /* Strongly recommend that these be loaded from a configuration file
dynamically in production code */
29
30         String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public
certificate for the MISE
31         String trustFabricUrl = "<a
href="https://mise.mda.gov/miseresources/TrustFabric.xml">https://mise.mda.gov/mise
resources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
32         String trustFabricBackupPath =
```

```

"C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location
for a cached version of the trust fabric
33     String serverScheme = "https";
34     String serverHost = "mise.mda.gov";
35     String serverPort = "9443";
36     String serverBasePath = "/services";
37     String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore
which contains the certificate and private key for this trusted system
38     String keystorePass = "password";
39
40
41
42     try {
43         FileInputStream isCert = new
FileInputStream(FilenameUtils.separatorToSystem(miseCert));
44         CertificateFactory certFactory =
CertificateFactory.getInstance("X.509");
45         X509Certificate cert = (X509Certificate)
certFactory.generateCertificate(isCert);
46
47         TrustFabric.initializeFromURL(trustFabricUrl,
trustFabricBackupPath, cert);
48
49         m_client = new RestServiceClient(serverScheme, serverHost,
Integer.valueOf(serverPort), serverBasePath);
50
51         m_client.setClientCert(FilenameUtils.separatorToSystem(keystorePath),
keystorePass);
52
53         String body = null;
54         FileUtils.readFileToString(new File("SamplePosition.xml"), body);
55
56         HttpResponse response = m_client.put("/publish/pos/id", "", body,
ContentType.APPLICATION_XML); //perform the request
57     } catch(Exception e) {
58         e.printStackTrace();
59     }
60
61
62 }
63
64 }

```

Note again that all the configuration parameters should not be hardcoded as strings in production code, but should be loaded dynamically from a configuration file or configuration database.

Examining the code in detail, the following 4 lines load the SSL certificate for the MISE from the file system, and initialize the Trust Fabric. The Trust Fabric code attempts to load the MISE-hosted Trust Fabric from the MISE endpoint first, and can fall back to a local copy if the MISE endpoint cannot be accessed.

```

1  FileInputStream isCert = new
2  FileInputStream(FilenameUtils.separatorToSystem(miseCert));
3  CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
4  X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);
5
6  TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);

```

The next section of the code creates the MISE Rest Client and initializes it with the client key store. The client key store must contain the private key corresponding to the certificate registered for the publishing system in the Trust Fabric.

```
1 m_client = new RestServiceClient(serverScheme, serverHost,
  Integer.valueOf(serverPort), serverBasePath);
2
3 m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
  keystorePass);
```

Finally, the last few lines read the XML from a file and publish it to the MISE. The /id on the publish URL must actually be a unique ID for this Position message in the publishing system.

```
1 String body = null;
2 FileUtils.readFileToString(new File("SamplePosition.xml"), body);
3
4 HttpResponse response = m_client.put("/publish/pos/id", "", body,
  ContentType.APPLICATION_XML); //perform the request
```

In production, the publishing, response-handling, and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the publish operation.

Please note that the current base URL for the MSIE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

## 8. Interfacing with the Delete Service

The delete interface on the MISE is an extension of the publish interface. To delete a previously published information product, the publishing system need only issue a DELETE HTTP request with the same parameters as the original publish message. The complete interface description for delete is in the Publish Specification. See the Code Overview<sup>1</sup> page for the code download and library details. The ClientTest project containing these code examples is also available on that page.

The following code example is structurally identical to the publish example, save for the actual HTTP operation on line 47, which is a DELETE, instead of a PUT with XML content.

```
1 package test;
2
3 import gov.mda.trustfabric.TrustFabric;
4 import gov.mda.util.RestServiceClient;
5
6 import java.io.FileInputStream;
7 import java.security.cert.CertificateFactory;
8 import java.security.cert.X509Certificate;
9
10 import org.apache.commons.io.FileUtils;
11 import org.apache.http.HttpResponse;
12
13 public class TestDeleteClient {
```

---

<sup>1</sup> <https://mise.mda.gov/drupal/node/25>

```

14
15     /**
16      * @param args
17      */
18     public static void main(String[] args) {
19         RestClient m_client;
20
21
22         /* Strongly recommend that these be loaded from a configuration file
23         dynamically in production code */
24
25         String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public
26         certificate for the MISE
27         String trustFabricUrl = "<a
28         href="https://mise.mda.gov/miseresources/TrustFabric.xml">https://mise.mda.gov/mise
29         resources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
30         String trustFabricBackupPath =
31         "C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location
32         for a cached version of the trust fabric
33         String serverScheme = "https";
34         String serverHost = "mise.mda.gov";
35         String serverPort = "9443";
36         String serverBasePath = "/services";
37         String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore
38         which contains the certificate and private key for this trusted system
39         String keystorePass = "password";
40
41
42
43         try {
44             FileInputStream isCert = new
45             FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
46             CertificateFactory certFactory =
47             CertificateFactory.getInstance("X.509");
48             X509Certificate cert = (X509Certificate)
49             certFactory.generateCertificate(isCert);
50
51             TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath,
52             cert);
53
54             m_client = new RestClient(serverScheme,
55             serverHost, Integer.valueOf(serverPort), serverBasePath);
56
57             m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
58             keystorePass);
59
60             HttpResponse response = m_client.delete("/publish/pos/id", "", null,
61             null);
62         } catch (Exception e) {
63             e.printStackTrace();
64         }
65     }
66 }

```

For the DELETE operation, the /id on the publish URL must actually be the ID under which the information product was originally published.

```
1 HttpResponse response = m_client.delete("/publish/pos/id", "", null, null);
```

In production, the delete, response-handling, and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the delete operation.

Please note that the current base URL for the MSIE is `/services/MDAService/`, followed by `publish`, `search`, or `retrieve`, as described in this guide.

## 9. Interfacing with the Search Service

The search service provides the interface to search for information products on the MISE. The complete interface description for search is in the Search/Retrieve Specification. See the [Code Overview](#) page for the code download and library details. The ClientTest project containing these code examples is also available on that page.

This example walks through a query for Position reports based on the first Position [User Story](#), a search bounded by geospatial area and time. This example assumes the searching system can interface with the MISE [security services](#), including registering the system as a trusted system in the Trust Fabric.

Unlike the previous two examples which only require the SSL handshake with the MISE, the search and retrieve operations require that the consumer system pass the entitlement attributes of the user. The meanings of the attributes are discussed in detail in the [Attribute Specification](#). These entitlement attributes are provided to the MISE via a SAML assertion. When the SAML assertion is validated, the MISE returns a session cookie to the client system, which must be provided for all subsequent search and retrieve operations.

```
1 package test;
2
3 import gov.mda.Constants;
4 import gov.mda.saml.AssertionBuilder;
5 import gov.mda.saml.SAMLUtills;
6 import gov.mda.trustfabric.TrustFabric;
7 import gov.mda.util.RestServiceClient;
8
9 import java.io.FileInputStream;
10 import java.security.cert.CertificateFactory;
11 import java.security.cert.X509Certificate;
12
13 import org.apache.commons.io.FilenameUtils;
14 import org.apache.http.HttpResponse;
15 import org.apache.http.entity.ContentType;
16 import org.apache.http.util.EntityUtils;
17 import org.opensaml.saml2.core.Assertion;
18 import org.opensaml.saml2.core.Attribute;
19
20 public class TestSearchClient {
21     /**
22      * @param args
23      */
24     public static void main(String[] args) {
25         RestServiceClient m_client;
26
27         /* Strongly recommend that these be loaded from a configuration file
```

```

29 dynamically in production code */
30
31     String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public
32 certificate for the MISE
33     String trustFabricUrl = "<a
34 href=https://mise.mda.gov/miseresources/TrustFabric.xml>https://mise.mda.gov/mise
35 resources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
36     String trustFabricBackupPath =
37 "C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location
for a cached version of the trust fabric
38     String serverScheme = "https";
39     String serverHost = "mise.mda.gov";
40     String serverPort = "9443";
41     String serverBasePath = "/services";
42     String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore
which contains the certificate and private key for this trusted system
43     String keystorePass = "password";
44
45     try {
46         FileInputStream isCert = new
47 FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
48         CertificateFactory certFactory =
49 CertificateFactory.getInstance("X.509");
50         X509Certificate cert = (X509Certificate)
51 certFactory.generateCertificate(isCert);
52
53         TrustFabric.initializeFromURL(trustFabricUrl,
54 trustFabricBackupPath, cert);
55
56         m_client = new RestServiceClient(serverScheme,
57 serverHost, Integer.valueOf(serverPort), serverBasePath);
58         m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
59 keystorePass);
60
61         //Form the user assertion
62         String assertingPartyID = "test.client";
63         AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
64         builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION,
65 10*60); // valid for 10 minutes
66         builder.addAttribute("ElectronicIdentityId",
67 "gfipm:2.0:user:ElectronicIdentityId", "<a
68 href=mailto:testuser@testsystem.gov>testuser@testsystem.gov</a>");
69         builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T.
70 User");
71
72         Attribute attr = builder.addAttribute("CitizenshipCode",
73 "mise:1.4:user:CitizenshipCode", "USA");
74         builder.addAttribute("LawEnforcementIndicator",
75 "mise:1.4:user:LawEnforcementIndicator", "true");
76         builder.addAttribute("PrivacyProtectedIndicator",
77 "mise:1.4:user:PrivacyProtectedIndicator", "true");
78
79         builder.signUsingPkcs12(assertingPartyID,
80 FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
81         Assertion assertion = builder.getAssertion();
82
83         //Important to not use SAMLUtils.asPrettyXMLString(object) as it will
84 cause the signature validation to fail
85         HttpResponse response = m_client.post("/MDAUserService/login",
86 null, SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);
87         EntityUtils.consumeQuietly(response.getEntity());
88         response = m_client.get("/MDAService/search/pos?ulat=3.75&ulng=-
89 2.0&llat=-2.75&llng=3.0&start=2012-06-10T12:10:00&end=2013-01-25T12:30:00", null);

```

```
70         //do something with the response
71     } catch(Exception e) {
72         e.printStackTrace();
73     }
74 }
75 }
```

Note again that all the configuration parameters should not be hardcoded as strings in production code, but should be loaded dynamically from a configuration file or configuration database.

Examining the code in detail, the following 4 lines load the SSL certificate for the MISE from the file system, and initialize the Trust Fabric. The Trust Fabric code attempts to load the MISE-hosted Trust Fabric from the MISE endpoint first, and can fall back to a local copy if the MISE endpoint cannot be accessed.

```
1 FileInputStream isCert = new
  FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
2 CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
3 X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);
4
5 TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);
```

The next section of the code creates the MISE Rest Client and initializes it with the client key store. The client key store must contain the private key corresponding to the certificate registered for the publishing system in the Trust Fabric.

```
1 m_client = new RestServiceClient(serverScheme, serverHost,
  Integer.valueOf(serverPort), serverBasePath);
2
3 m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
  keystorePass);
```

The next section of the code deals with the creation of the SAML assertion with the user's attributes. The AssertionBuilder is a utility provided by the MDA toolkit to aid in creating the SAML. The required attributes are defined in the attribute specification. Every assertion must provide the ElectronicIdentityID, FullName, CitizenshipCode, and entitlement attributes. Note that the example below shows how to explicitly set the expiration time for the assertion. If not included, the sessions formed by each assertion are timed-out automatically.

```
1 String assertingPartyID = "test.client";
2 AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
3 builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION, 10*60); // valid
  for 10 minutes
4 builder.addAttribute("ElectronicIdentityId", "gfipm:2.0:user:ElectronicIdentityId",
  "<a href='mailto:testuser@testsystem.gov'>testuser@testsystem.gov</a>");
5 builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T. User");
```

The following code shows how to set the citizenship and entitlement information. These attributes are mapped from the consumer system's internal user database.

```
1 Attribute attr = builder.addAttribute("CitizenshipCode",
  "mise:1.4:user:CitizenshipCode", "USA");
```

```
2 builder.addAttribute("LawEnforcementIndicator",  
    "mise:1.4:user:LawEnforcementIndicator", "true");  
3 builder.addAttribute("PrivacyProtectedIndicator",  
    "mise:1.4:user:PrivacyProtectedIndicator", "true");
```

As the final step in the process to create the SAML assertion, the assertion must be signed using the private key of the consumer system. This is the same private key/key store used to establish the SSL connection. Note that this signing operation explicitly includes the `assertingPartyID`, which should match the entity ID of the consumer system registered with the Trust Fabric.

```
1 builder.signUsingPkcs12(assertingPartyID,  
    FilenameUtils.separatorsToSystem(keystorePath), keystorePass);  
2 Assertion assertion = builder.getAssertion();
```

Once the assertion has been created, the HTTP request for the session and the search can be performed. Prior to the actual request, the consuming system must establish a session with the MISE with the entitlements for the requesting user. This code makes that request using the assertion that was just created. The `RestClient` internally stores the session cookie provided back by the MISE to use in future requests.

```
1 //Important to not use SAMLUtils.asPrettyXMLString(object) as it will cause the  
    signature validation to fail  
2 HttpResponse response = m_client.post("/MDAUserService/login", null,  
    SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);  
3 EntityUtils.consumeQuietly(response.getEntity());
```

Finally, once the assertion has been created and the session established, the search request can be made. The search request shown requests all `Position` instances in the specified bounding box, published in the specified time period. This will return an Atom feed containing summaries of all matching records in the MISE. The retrieve operation for each of the individual records in the Atom feed is discussed in the next section.

```
1 response = m_client.get("/MDAService/search/pos?ulat=3.75&ulng=-2.0&llat=-  
    2.75&llng=3.0&start=2012-06-10T12:10:00&end=2013-012-25T12:30:00", null);
```

In production, the searching response handling, and error handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the search operation.

Please note that the current base URL for the MISE is `/services/MDAService/`, followed by `publish`, `search`, or `retrieve`, as described in this guide.

## 10. Interfacing with the Retrieve Service

Once a search operation has been performed, the Retrieve interface on the MISE allows a consuming system to retrieve the complete NIEM-M XML instance of any record. Alternately, the retrieve URL for any record can be accessed directly if known, bypassing the search operation entirely.

The complete interface description for search is in the Search/Retrieve Specification. See the [Code Overview](#) page for the code download and library details. The ClientTest project containing these code examples is also available on that page.

Each record published to the MISE creates a unique retrieve URL for that record. As long as that record exists in the MISE cache, it can be accessed via that URL. The following example demonstrates the retrieve operation. As with the search, retrieve requires that the consumer system pass the entitlement attributes of the user. The meanings of the attributes are discussed in detail in the [Attribute Specification](#). These entitlement attributes are provided to the MISE via a SAML assertion. When the SAML assertion is validated, the MISE returns a session cookie to the client system, which must be provided for all subsequent retrieve operations.

```
1  package test;
2
3  import gov.mda.Constants;
4  import gov.mda.saml.AssertionBuilder;
5  import gov.mda.saml.SAMLUtils;
6  import gov.mda.trustfabric.TrustFabric;
7  import gov.mda.util.RestServiceClient;
8
9  import java.io.FileInputStream;
10 import java.security.cert.CertificateFactory;
11 import java.security.cert.X509Certificate;
12
13 import org.apache.commons.io.FilenameUtils;
14 import org.apache.http.HttpResponse;
15 import org.apache.http.entity.ContentType;
16 import org.opensaml.saml2.core.Assertion;
17 import org.opensaml.saml2.core.Attribute;
18
19 public class TestRetrieveClient {
20
21     /**
22      * @param args
23      */
24     public static void main(String[] args) {
25         RestServiceClient m_client;
26
27         /* Strongly recommend that these be loaded from a configuration file
28            dynamically in production code */
29
30         String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public
31         String trustFabricUrl = "<a
32 href=https://mise.mda.gov/miseresources/TrustFabric.xml>https://mise.mda.gov/mise
33 resources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
34         String trustFabricBackupPath =
35 "C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location
36         for a cached version of the trust fabric
37         String serverScheme = "https";
38         String serverHost = "mise.mda.gov";
39         String serverPort = "9443";
40         String serverBasePath = "/services";
41         String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore
42         which contains the certificate and private key for this trusted system
43         String keystorePass = "password";
44
45         try {
46             FileInputStream isCert = new
47             FileInputStream(FilenameUtils.separatorToSystem(miseCert));
```

```

41         CertificateFactory certFactory =
CertificateFactory.getInstance("X.509");
42         X509Certificate cert = (X509Certificate)
certFactory.generateCertificate(isCert);
43
44         TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath,
cert);
45
46         m_client = new RestServiceClient(serverScheme,
serverHost, Integer.valueOf(serverPort), serverBasePath);
47         m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
keystorePass);
48
49         //Form the user assertion
50         String assertingPartyID = "test.client";
51         AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
52         builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION,
10*60); // valid for 10 minutes
53         builder.addAttribute("ElectronicIdentityId",
"gfipm:2.0:user:ElectronicIdentityId", "<a
href="mailto:testuser@testsystem.gov">testuser@testsystem.gov</a>");
54         builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T.
User");
55
56         Attribute attr = builder.addAttribute("CitizenshipCode",
"mise:1.4:user:CitizenshipCode", "USA");
57         builder.addAttribute("LawEnforcementIndicator",
"mise:1.4:user:LawEnforcementIndicator", "true");
58         builder.addAttribute("PrivacyProtectedIndicator",
"mise:1.4:user:PrivacyProtectedIndicator", "true");
59
60         builder.signUsingPkcs12(assertingPartyID,
FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
61         Assertion assertion = builder.getAssertion();
62
63         //Important to not use SAMLUtils.asPrettyXMLString(object) as it will
cause the signature validation to fail
64         HttpResponse response = m_client.post("/MDAUserService/login",
null, SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);
65
66         String entityID = "https%3A%2F%2Fmise.agencyone.gov%2F";
67         String uuid = "79869883848892520";
68         response = m_client.get("/MDAService/retrieve/ian?entityid=" + entityID
+ "&recordid=" + uuid, null);
69
70         //do something with the response
71
72     } catch(Exception e) {
73         e.printStackTrace();
74     }
75 }
76 }

```

Note that all the configuration parameters should not be hardcoded as strings in production code, but should be loaded dynamically from a configuration file or configuration database.

Examining the code in detail, the following 4 lines load the SSL certificate for the MISE from the file system, and initialize the Trust Fabric. The Trust Fabric code attempts to load the MISE-hosted Trust Fabric from the MISE endpoint first, and can fall back to a local copy if the MISE endpoint cannot be accessed.

```

1 FileInputStream isCert = new
  FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
2 CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
3 X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);
4
5 TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);

```

The next section of the code creates the MISE Rest Client and initializes it with the client key store. The client key store must contain the private key corresponding to the certificate registered for the publishing system in the Trust Fabric.

```

1 m_client = new RestServiceClient(serverScheme, serverHost,
  Integer.valueOf(serverPort), serverBasePath);
2
3 m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
  keystorePass);

```

The next section of the code deals with the creation of the SAML assertion with the user's attributes. The AssertionBuilder is a utility provided by the MDA toolkit to aid in creating the SAML. The required attributes are defined in the attribute specification. Every assertion must provide the ElectronicIdentityID, FullName, CitizenshipCode, and entitlement attributes. Note that the example below shows how to explicitly set the expiration time for the assertion. If not included, the sessions formed by each assertion are timed-out automatically.

```

1 String assertingPartyID = "test.client";
2 AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
3 builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION, 10*60); // valid
  for 10 minutes
  builder.addAttribute("ElectronicIdentityId", "gfipm:2.0:user:ElectronicIdentityId",
4 "<a href='mailto:testuser@testsystem.gov'>testuser@testsystem.gov</a>");
5 builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T. User");

```

The following code shows how to set the citizenship and entitlement information. These attributes are mapped from the consumer system's internal user database.

```

1 Attribute attr = builder.addAttribute("CitizenshipCode",
  "mise:1.4:user:CitizenshipCode", "USA");
2 builder.addAttribute("LawEnforcementIndicator",
  "mise:1.4:user:LawEnforcementIndicator", "true");
3 builder.addAttribute("PrivacyProtectedIndicator",
  "mise:1.4:user:PrivacyProtectedIndicator", "true");

```

As the final step in the process to create the SAML assertion, the assertion must be signed using the private key of the consumer system. This is the same private key/key store used to establish the SSL connection. Note that this signing operation explicitly includes the assertingPartyID, which should match the entity ID of the consumer system registered with the Trust Fabric.

```

1 builder.signUsingPkcs12(assertingPartyID,
  FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
2 Assertion assertion = builder.getAssertion();

```

Once the assertion has been created, the HTTP request for the session and the retrieve can be performed. Prior to the actual request, the consuming system must establish a session with the MISE with the entitlements for the requesting user. This code makes that request using the

assertion that was just created. The RestClient internally stores the session cookie provided back by the MISE to use in future requests.

```
1 //Important to not use SAMLUtils.asPrettyXMLString(object) as it will cause the
  signature validation to fail
2 HttpResponseMessage response = m_client.post("/MDAUserService/login", null,
  SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);
```

Finally, once the assertion has been created and the session established, the retrieve request can be made. Each record has an ID that is unique to the publishing system, so the entity ID/record ID pair provides a unique key for the record. The entity ID in each case is the same as the entity ID in the trust fabric.

```
1 String entityID = "https%3A%2F%2Fmise.agencyone.gov%2F";
2 String uuid = "79869883848892520";
3 response = m_client.get("/MDAService/retrieve/ian?entityid=" + entityID +
  "&recordid=" + uuid, null);
```

In production, the response-handling and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the retrieve operation.

Please note that the current base URL for the MISE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

## 11. Testing on the Test Service Platform

Once the initial development work has been completed, the operation of the code can be tested against the MISE Test Platform. The following steps detail the process to test.

All information exchanged on the Test Platform must be TEST ONLY. No operational data may be exchanged on the Test Platform

1. Contact the MISE Engineering Team to obtain a test key store. The test key store provides the private key and certificate for one of the identities in the test Trust Fabric. This will allow connections with the security services on the MISE Test Platform.
2. Test the SSL handshake process with the Test Platform. Using the test key store, it should be possible to make the initial SSL connection, enabling further testing. Test the
3. Once the SSL connection has been established, test information can be published. Test, publish, and update for success and error conditions.
4. With the SSL connection, the delete operation can also be tested. Test delete for both success and error conditions.
5. If the search and retrieve services are required, the SAML assertions for user entitlement exchange should be tested. Ensure that the trusted system code can create and sign the necessary assertions to pass user entitlement information. Once a SAML assertion is correctly passed to the Test Platform, it will return a session cookie representing that user entitlement session, enabling access to the search and retrieve services. This test

process should also test that the trusted system code correctly separates sessions, handles errors, logs out, and handles SAML re-assertion when a session expires.

6. Once user entitlement sessions can be established, the search service can be tested. Test that issuing queries for previously published information is successful. Test for error conditions and handling empty responses. Finally, if a query will return too much information, the MISE will issue a 413 error code. Make sure this case is handled, typically by refining and re-issuing the query.
7. The final operation to test is retrieve. Using the results of a search, check that the full NIEM-M instance document can be retrieved. Again, make sure that error conditions are correctly handled
8. For a system that does both publish and search/retrieve, a final end-to-end test can be performed by publishing and then searching for and retrieving an information product or a set of products.

Testing a publishing system requires steps 1-4, since user entitlements and session handling is not required for publish and delete. Testing a search/retrieve consumer system requires steps 1,2, and 5-8. The MISE Engineering Team is available to help with error logs and debugging code on the MISE side. The [Specifications](#) detail all of the error code and header information that might be returned by the MISE for success and failure states in the interaction.

## 12. Going Live on the Integration Platform

Once the Testing is complete, the trusted system development team should perform the following actions in conjunction with the MISE Engineering Team to go live on the MISE Integration Platform.

1. Provide the public certificate for the private key that will be used for all interactions with the MISE Integration Platform.
2. The MISE engineering team will place this certificate and the trusted system information in the Trust Fabric and sign the new Trust Fabric with the MISE private key as discussed in the [Security Specification](#).
3. At an agreed-upon time, the new Trust Fabric will be loaded on the MISE Integration Platform and made available to all trusted systems. This is a hot-reload operation, which does not require that the MISE Integration Platform be taken offline.
4. The trusted system can now begin publishing to and consuming information from the MISE. The first publish/search/retrieve operations should be monitored by both the trusted system development team and the MISE Engineering Team to ensure successful operation and if any troubleshooting is required.

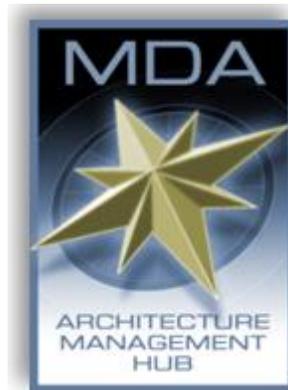
---

---

# APPENDIX B - ATTRIBUTE SPECIFICATION

---

---



## Maritime Information Sharing Environment (MISE)

Attribute Specification

Version 1.0

25 March 2013

# 1. Introduction

Maritime security is a national priority that depends on the ability to efficiently, effectively, and appropriately share and safeguard information among trusted maritime partners within the Global Maritime Community of Interest (GMCOI). The Maritime Information Sharing Environment (MISE) as defined in the National Maritime Architecture Plan enables secure, standardized sharing of unclassified maritime information among a wide variety of federal, state and local agencies as well as international participants. MISE employs attribute-based access control and a standardized set of security attributes for information access policy enforcement to facilitate information sharing with non-provisioned users in a dynamic environment. This security model embraces the philosophy that user accounts are most effectively managed by a user's parent organization and exploiting existing user account verification, management, and revocation processes already in place. Allowing access using an individual's existing local user identity and password improves security by eliminating the requirement to create and maintain yet another user identity. The federated model also eliminates the need for changes to user account privileges across multiple systems; an individual need only advise their parent organization of changes in employment status or organizational role, and changes to account privileges at the local level are sufficient to ensure security across the MISE domain. By federating the identity management to the trusted systems, the cost and burden of managing user accounts is greatly reduced for the MISE thus reducing overall sustainment costs. A common set of security attributes for entities, users, and data is necessary to consistently share and protect information across the federation of systems in the MISE.

## 1.1. PURPOSE

This specification defines a common set of attributes used within the Maritime Information Sharing Environment (MISE) to communicate information about users and the trusted systems that connect to the Information Sharing Infrastructure (ISI) on the user's behalf.

There are three categories for attributes defined for the National MDA Architecture:

1. Entity Attributes: Attributes that pertain to a trusted system within the MISE.
2. User Attributes: Attributes that pertain to a human user.
3. Data Attributes: Attributes that pertain to data.

Figure 1 summarizes the entity, user, and data attributes defined for the MISE.

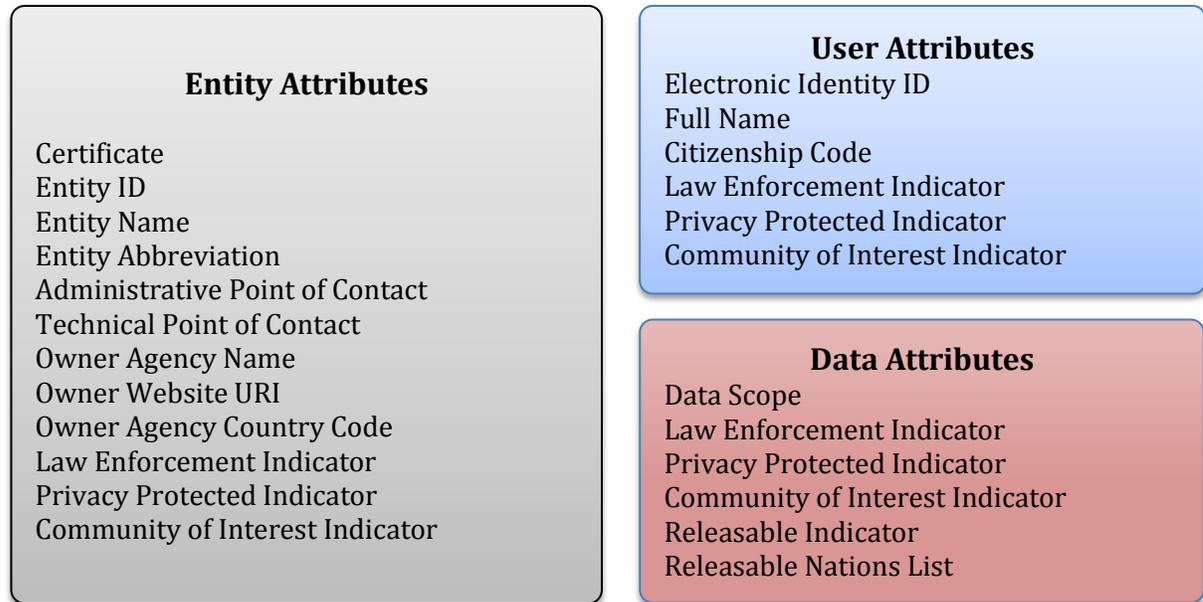


Figure 1 - MISE Attributes Summary

Entity attributes are used to capture relevant information about the trusted system i.e. SSL certificate, administrative information, and website Uniform Resource Identifier (URI). The Owner Agency Country Code is used in protecting data based on nation. An entity is restricted from accessing information unless it is explicitly coded for release to their country code. The "Indicator" attributes are key to establishing what categories of information the trusted system can access.

User attributes provide detailed information on the individual users of a trusted system, and serve to ensure that they are only able to access that subset of the available information to which they individually require and qualify for access. The user attributes are assigned and managed by the trusted system to which the user belongs.

Data attributes are fundamental to entitlement management and data management processes within the MISE. The three data attributes defined in this specification as “indicators” support entitlement management within the MISE. The “scope” attribute provides the flexibility to associate data with a specific incident or operation to provide context for data management.

## 1.2. INDICATORS

Entitlement management within the MISE involves run-time authorization decisions about whether a trusted system and requesting user are authorized to access a requested information resource. Three primary security indicators are used by data providers to declare information access policy to set access restrictions on information they share with MISE. These information access policies are the basis for entitlement decisions within the MISE. A fourth indicator, the “Releasable Indicator”, is a Boolean used in conjunction with the three primary Security Indicators to mark data as releasable to the public domain under the restrictions of the associated indicator. The three primary security indicators are listed in the following table in order of decreasing degree of restriction.

Indicator	Abbreviation
Law Enforcement	LEI
Privacy Protected	PPI
Protected Critical Infrastructure Information	PCII
Sensitive but Unclassified	SBU
Federal State Local and Tribal	FSLT
Private Sector Only	PSO
Community of Interest	COI

Table 1 – Security Indicators

The following describes the three primary attributes and explains their intent as well as how the attribute might be used:

**Law Enforcement Indicator:** This Indicator is the most restrictive, and is used to code data for release to entities such as federal, state and local law enforcement agencies. Only Entities assigned the Law Enforcement Indicator and the U.S. owner agency code can access this data. Within an Entity with the Law Enforcement Indicator, only Users who are U.S. Citizens and assigned the Law Enforcement Indicator by that Entity will be able to access the information.

**Privacy Protected Indicator:** Personally Identifiable Information (PII) is information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify a single individual. State and Federal legislation, as well as the policy of many agencies such as DOD and DHS, impose strict limitations on the release of PII. The Privacy Protected Indicator is designed to restrict access to and distribution of PII across the MISE Domain. Application of the Privacy Protected Indicator means the information will only be released to Entities with the requisite country code, and then only to Individuals with that attribute and the requisite country code assigned to their account.

**Protected Critical Infrastructure Information:** Protected Critical Infrastructure Information Indicator (PCII) Communities are responsible for protecting national/state identified critical infrastructure facilities. Critical infrastructure information means information not customarily in the public domain and related to the security of criminal infrastructure or protected systems.

**Sensitive but Unclassified:** Non PPI, non LEA information that is not publically releasable. Sensitive but Unclassified is an indicator category of information that requires protections against discloser. This security indicator limits the discloser of information obtained or developed in carrying out certain security or research and development activities to the extent that it has been determined that disclosure of the information would be an unwarranted invasion of personal privacy; reveal a trade secret or privileged or confidential commercial or financial information; or be detrimental to the safety of passengers in transportation.

**Federal, State, Local, & Tribal:** Defines the level of government authorized to receive information. Information limited to jurisdictional boundaries, or that is partitioned to specific geographic area, depending on resources and limitations.

**Private Sector Only:** Private Sector Only is an indicator that protects commercial and proprietary interest. Information cannot be redistributed to other entities due to the enclosed information and is only intended for authorized disclosures.

**Community of Interest Indicator:** The community of interest indicator is the minimum access level assigned to trusted systems participating in the environment. By default, all trusted systems are granted the Community of Interest Indicator.

The following two modifiers can be used in conjunction with any of the three primary security indicators:

**Releasable Indicator (default False):** This attribute is a Boolean value to mark data as releasable to the public domain under the restrictions of the associated indicator. This attribute is used to indicate data can be released in accordance with the consuming systems policies, business processes and as required to support their mission.

**Releasable Nations List (default USA):** This provides a comma-separated list of nations who can access the associated data. This is a space-separated list of three-letter country codes, for example (CAN USA FRA). This attribute is used to indicate data can be released to only those nations identified by country codes.

Scope provides tags to associate data with a specific incident or operation. These tags are used by data providers to relax or override their IAP for normal operations so trusted systems and users are able to access data they would normally be restricted from, but only within the scope of their involvement in the specified event or operation. Scope may be a planned event such as the Olympics or specific response operation in the wake of a specific disaster such as Hurricane Katrina. Each scope is named, and provides three modifiers to the data indicators discussed in the previous section:

**ScopeName:** Unique name of a scope. This will indicate which event or operation within which the scope is in context, e.g. SuperstormSandy.

**ScopeIndicator:** Minimum indicator required for access within the context of this scope. If this data is normally PPI data, the data provider might want to provide it to all COI consumers within the context of this scope.

**ScopeReleasable:** This attribute is a Boolean value to mark data as releasable to the public domain under the restrictions of the associated indicator. This attribute is used to indicate data is can be released in accordance with the consuming systems policies, business processes and as required to support their mission, within the context of the associated scope.

**ScopeReleaseableNations:** This provides a comma-separated list of nations who can access the associated data within the context of the associated scope. This is a space-separated list of three-letter country codes, for example (CAN USA FRA).

As an example, the following table shows in the case of a trusted system providing position reports, the policy during routine operations is PPI required for access because the data contains US Persons information. However, in support of disaster relief operations during Hurricane

Sandy the data provider may decide they have a need to share with any trusted system within the context of Humanitarian Aid and Disaster Relief (HADR) operations so COI can be set as minimum requirement for access. The data provider can modify the indicator, releaseability, and releasable nations in the context of the Hurricane Sandy scope only.

Use Case (Tracks)	Routine Operations	Indicator/Scope
HADR (US Persons)	PPI	COI / SuperstormSandy

Table 2 – Scope Example

The four attributes described for scope are not new attributes, but simply modifiers on the existing data attributes defined in Section 4 of this specification.

## 2. Entity Attributes

The following entity attributes are associated with a trusted system.

2.1 Administrative Point of Contact – Email Address Text	
Name	AdministrativePointofContactEmailAddressText
Description	The electronic mailing address by which to contact the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactEmailAddressText
Example Values	“john.doe@company.com”
2.2 Administrative Point of Contact – Fax Number	
Name	AdministrativePointofContactFaxNumber
Description	The telephone number for a facsimile device by which to contact the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification

Formal Name	gfipm:2.0:entity:AdministrativePointofContactFaxNumber
Example Values	“(555) 555-5555”
<b>2.3 Administrative Point of Contact – Full Name</b>	
Name	AdministrativePointofContactFullName
Description	The complete name of the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactFullName
Example Values	“John Doe”
<b>2.4 Administrative Point of Contact – Telephone Number</b>	
Name	AdministrativePointofContactTelephoneNumber
Description	The telephone number for a telecommunications device by which to contact the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactTelephoneNumber
Example Values	“(555) 555-5555”
<b>2.5 Certificate</b>	
Name	Certificate
Description	An electronic certificate used by the entity as a cryptographic trust anchor within a federation for the purposes of digital signatures and/or encryption. The certificate is represented in X.509 v3, base-64 encoded format.
Data Type	Base-64 Binary
Typical Usage	Authentication, Registration, Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:Certificate
Example Values	"MIICJzCCAZCgAwIBAgIBGDANBgkqhkiG9w0BAQUFADBAMQswCQYDQQGEwJVUzEMMAoGAIUEChMDSVNDM

	SMwiQYDVQQDExpJU0MgQ0RLIFNhbXBsZSBDZXJ0aWZpY2F0ZTAeFw0wMzA3MTcwMDAwMDBaFw0wNDA3MTcwMDAwMDBaMEAx CzAJBgNVBAYTAIVTMQwwCgYDVQKQEWnJU0MxIzAhBgNVBAMTGklTQyBDREsgU2FtcGxIIENlcnRpZmljYXRIMIGfMA0GCSqGSIb3DQEBAQUAA4GNA DCBiQKBgQC9GQTkukn+153rATR8dh2Hm8ixF7f7Y7b10VFJnJAJQCKqta4/lhFwQIK5F2Gn8j9tITBiXCF7F6XSvaF8bivN10zR0pvl11NflEm2kwh7Yw0jZJB17Y3FHgl83qYegmm/UwqX5zKUa4xw+cE8XSEqUuwjg0roBMGhAMzFEihHzLwIDAQABozEwLzAMBgNVHRMBAf8EAjAAMA4GAIUdDwEB/wQEAwIAYDAPBgNVHQ4ECHJzYS0xMDI0MA0GCSqGSIb3DQEBBQUAA4GBALWGxxo55ScpLFEcnqEUixFwrzftQGD2ISda7EWp/d7k23fOXgHC7Zal8OpvlBUZ3sC2Fg4finfRHd2J4mXONk50EdjhJILd58GErcCECg4J2uJPz77/zk+giiXldQEPtG+YOaAbZC2SFbdfyYDKiSPhgzyd0/b4cElf4+VzegRM"
<b>2.6 Community of Interest Indicator</b>	
Name	COIIndicator
Description	True if the entity is allowed access to COI data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:entity:COIIndicator
Example Values	"True," "False"
<b>2.7 Entity Abbreviation</b>	
Name	EntityAbbreviation
Description	An abbreviation or acronym for the name of a trusted system.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:EntityAbbreviation
Example Values	"MAGNET," "MSSIS"
<b>2.8 Entity ID</b>	
Name	EntityID
Description	The unique identifier by which the entity is denoted.
Data Type	Text
Typical Usage	Registration, Audit Logging
References	GFIPM 2.0 Metadata Specification

Formal Name	gfipm:2.0:entity:EntityId
Example Values	“MSSIS:123,” “MISE:TIB:MAGNET”
<b>2.9 Entity Name</b>	
Name	EntityName
Description	The name of an entity.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:EntityName
Example Values	“Maritime Analysis Global Network,” “Maritime Safety and Security Information System”
<b>2.10 Law Enforcement Indicator</b>	
Name	LawEnforcementIndicator
Description	True if the entity is allowed access to law enforcement protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:entity:LawEnforcementIndicator
Example Values	“True,” “False”
<b>2.11 Owner Agency Country Code</b>	
Name	OwnerAgencyCountryCode
Description	The country of the organization or agency by which the entity is owned and operated.
Data Type	Country Code ISO 3166-1
Typical Usage	Registration
References	
Formal Name	mise:1.4:entity:OwnerAgencyCountryCode
Example Values	“USA,” “GBR,” “FRA”
<b>2.12 Owner Agency Name</b>	
Name	OwnerAgencyName
Description	The name of the organization or agency by which the trusted

	system is owned.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:OwnerAgencyName
Example Values	”NORAD-USNORTHCOM”
<b>2.13 Owner Agency Website URI</b>	
Name	OwnerAgencyWebSiteURI
Description	The Internet address or website of the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:OwnerAgencyWebSiteURI
Example Values	”http://website.company.com”
<b>2.14 Privacy Protected Indicator</b>	
Name	PrivacyProtectedIndicator
Description	True if the entity is allowed access to privacy-protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:entity:PrivacyProtectedIndicator
Example Values	”True,” ”False”
<b>2.15 Technical Point of Contact – Email Address Text</b>	
Name	TechnicalPointofContactEmailAddressText
Description	The electronic mailing address by which to contact the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactEmailAddressText

Example Values	“john.doe@company.com”
<b>2.16 Technical Point of Contact – Fax Number</b>	
Name	TechnicalPointofContactFaxNumber
Description	The telephone number for a facsimile device by which to contact the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactFaxNumber
Example Values	“(555) 555-5555”
<b>2.17 Technical Point of Contact – Full Name</b>	
Name	TechnicalPointofContactFullName
Description	The complete name of the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactFullName
Example Values	“John Doe”
<b>2.17 Technical Point of Contact – Telephone Number</b>	
Name	TechnicalPointofContactTelephoneNumber
Description	The telephone number for a telecommunication device by which to contact the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactTelephoneNumber
Example Values	“(555) 555-5555”

Table 3 – List of Entity Attributes

### 3. User Attributes

The following user attributes are associated with an end user.

3.1 Citizenship Code	
Name	CitizenshipCode
Description	The country that has assigned rights, duties, and privileges to the user because of the birth or naturalization of the user in that country.
Data Type	ISO 3166-1 Alpha-3 Country Code
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:user:CitizenshipCode
Example Values	"USA," "GBR," "FRA"
3.2 Community of Interest Indicator	
Name	COIIndicator
Description	True if the user is allowed access to COI data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:user:COIIndicator
Example Values	"True," "False"
3.3 Electronic Identity Id	
Name	ElectronicIdentityId
Description	The unique identifier that is associated with the electronic identity for the user within the user's identity provider's user base.
Data Type	Text
Typical Usage	Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:user:ElectronicIdentityId
Example Values	"DOE.JOHN.A.2370295257"
3.4 Full Name	

Name	FullName
Description	The complete name of the user.
Data Type	Text
Typical Usage	Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:user:FullName
Example Values	“John Doe,” “Jim Q. Public”
<b>3.5 Law Enforcement Indicator</b>	
Name	LawEnforcementIndicator
Description	True if the user is allowed access to law enforcement protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:l.4:user:LawEnforcementIndicator
Example Values	“True,” “False”
<b>3.6 Privacy Protected Indicator</b>	
Name	PrivacyProtectedIndicator
Description	True if the user is allowed access to privacy-protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:l.4:user:PrivacyProtectedIndicator
Example Values	“True,” “False”

Table  
4 –  
List of  
User  
Attributes

## 4. Data Attributes

The following data attributes are associated with information exchanged within the MISE.

4.1 Community of Interest Indicator	
Name	COIIndicator
Description	True if the data is COI protected, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:CommunityOfInterestIndicator
Example Values	“True,” “False”
4.2 Data Scope	
Name	Scope
Description	An indicator which denotes a scope, situation, or event of interest for which data is deemed relevant and/or accessible.
Data Type	Text
Typical Usage	Authorization
References	
Formal Name	mise:1.4:data:Scope
Example Values	“HurricaneKatrina,” “Baltimore1812Exercise,” “OperationTomadachi”
4.3 Law Enforcement Indicator	
Name	LawEnforcementIndicator
Description	True if the data is law enforcement-protected, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:LawEnforcementIndicator
Example Values	“True,” “False”
4.4 Privacy Protected Indicator	

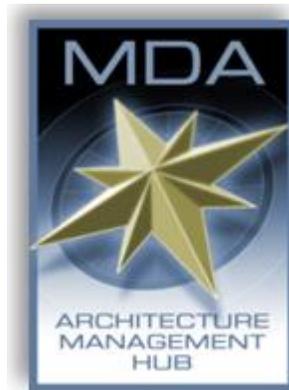
Name	PrivacyProtectedIndicator
Description	True if the data is privacy-protected, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:PrivacyProtectedIndicator
Example Values	“True,” “False”
<b>4.5 Releasable Indicator</b>	
Name	ReleasableIndicator
Description	True if the data is publicly releasable
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:ReleasableIndicator
Example Values	“True,” “False”
<b>4.6 Releasable Nations</b>	
Name	ReleasableNationsCodeList
Description	A space separated list of countries with assigned rights, duties, and privileges to access the data.
Data Type	ISO 3166-1 Alpha-3 Country Code
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:ReleasableNationsCodeList
Example Values	”USA GBR FRA”

Table 5 – List of Data Attributes

# APPENDIX C - INTERFACE SECURITY SPECIFICATION

---

---



## National MDA Architecture

Interface Security Specification

Version 1.0

25 March 2013

# 1. Introduction

The Maritime Information Share Environment (MISE) specifications for services such as Publication, Search, and Retrieval define Representational State Transfer (RESTful) interfaces specific to each service, but do not discuss specifics regarding how those interfaces are secured, how identity of the trusted system invoking the service is guaranteed, or how authenticated user attributes are delivered when applicable. The purpose of this specification is to discuss these aspects, which apply to all MISE services, and augment each RESTful interface defined in separate specifications. In addition, sample code demonstrating many aspects of MISE interface security is available at <https://mise.mda.gov>.

The *Security Architecture View* section of the *National Maritime Architecture Framework* should be reviewed prior to reading this specification, as it provides important overview and context to the material presented here. This specification does not repeat that overview and context information, but assumes the reader is familiar with it.

The design of MISE interface security has followed a number of patterns used in the U.S. Department of Justice’s *Global Federated Identity and Privilege Management* (GFIPM). There are some fundamental architectural differences between GFIPM and the MISE, which impact the design. Key among these are:

- MISE uses a hub and spoke network topology whereas GFIPM uses a point-to-point topology.
- MISE uses RESTful service interfaces whereas GFIPM uses Simple Object Access Protocol (SOAP)-based interfaces.

Despite these differences, there is a strong relationship between MISE and GFIPM. Portions of GFIPM documents are therefore incorporated into this document and modified to fit the MISE. In particular, [GFIPM Trust] and [GFIPM Services] were referenced in creation of this specification.

## 1.1. REFERENCES

RFC 2119	“RFC 2119—Key Words for Use in RFCs to Indicate Requirement Levels” Internet RFC/STD/FYI/BCP Archives <a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a>
GFIPM Trust	Federated Identity and Privilege Management (GFIPM): Cryptographic Trust Model <a href="http://www.it.ojp.gov/gist/Document/73">http://www.it.ojp.gov/gist/Document/73</a>
GFIPM Services	Federated Identity and Privilege Management (GFIPM): Web Services System-to-System Profile <a href="http://www.it.ojp.gov/gist/Document/122">http://www.it.ojp.gov/gist/Document/122</a>
SAML2 Metadata	“Metadata for the OASIS Security Markup Language (SAML) V2.0” OASIS Standard, 15 March 2005 Document Identifier: saml-metadata-2.0-os <a href="http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf">http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf</a>

SAML2 Core	“Assertions and Protocol for the OASIS Security Markup Language (SAML) V2.0” OASIS Standard, 15 March 2005 Document Identifier: saml-core-2.0-os <a href="http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf">http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf</a>
NIST SP 800-52	Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations National Institute of Science and Technology (NIST) Special Publication 800-52 <a href="http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf">http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf</a>

## 1.2. DOCUMENT STRUCTURE

Section 2 below describes details regarding key aspects of MISE interface security introduced in the *Security Architecture View* section of the *National Maritime Architecture Framework*.

Section 3 presents details of the Trust Fabric document.

Section 4 presents details of the Security Assertions Markup Language (SAML) assertions.

# 2. Process Flow and Processing Rules

This section discusses details regarding key aspects of MISE interface security, including the purpose of each as well as important implementation and operational requirements. Figure below illustrates these key aspects, and will be referenced throughout this section.

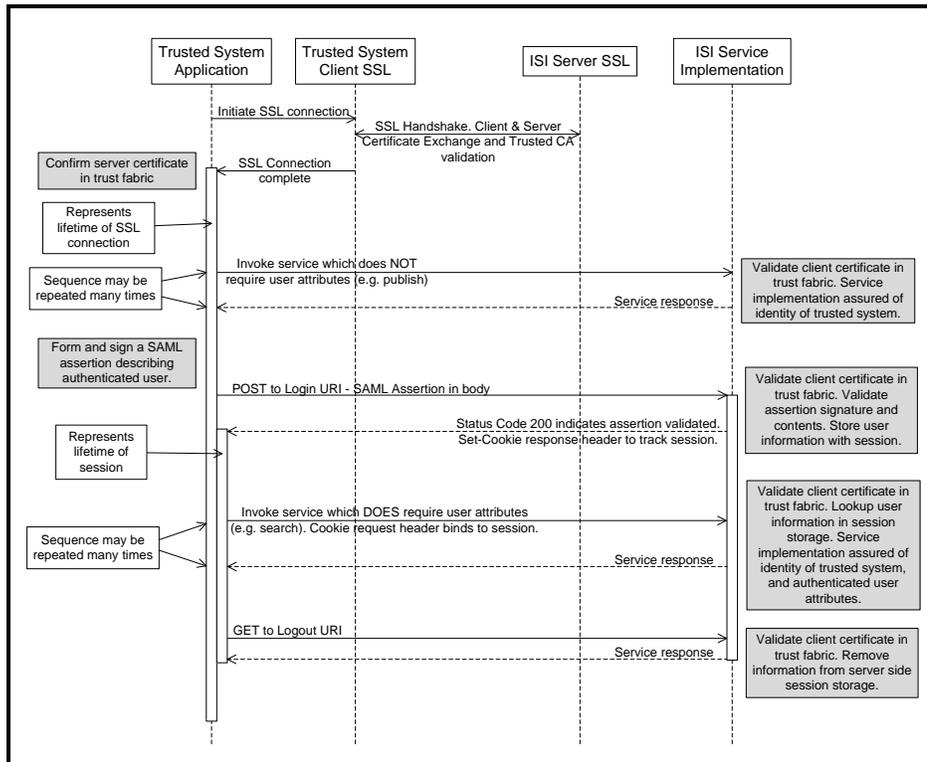


Figure 1 - Service Invocation Process Flow

## 2.1. X.509 CERTIFICATES AND PRIVATE KEYS

X.509 certificates and associated private keys are used for several purposes in MISE interface security. Specifically:

- Signing the trust fabric document by the MISE Certificate Authority (CA).
- Signing Security Assertions Markup Language (SAML) assertions (which contain user attributes) by information consumer systems.
- Client and server certificates for securing Secure Sockets Layer (SSL) connections between trusted systems and the Information Sharing Infrastructure (ISI).

### 2.1.1. PUBLIC / PRIVATE KEY PAIR

The process of creating an X.509 certificate begins with generating a pair of keys that are mathematically related - a *public key* and a *private key*. One key locks or encrypts data, and the other unlocks or decrypts the data. Neither key can perform both functions by itself.<sup>2</sup> The private key, as its name implies, must be kept secure. The public key is included inside the X.509 certificate, and must be available to any entity needing to engage in secure interaction with the possessor of the private key.

To be used in the MISE, all generated key pairs must be 2048-bit RSA<sup>3</sup> keys.

More detail regarding each usage of X.509 certificates in MISE interface security is provided in subsequent sections of this document. For each usage, Table 1 shows which system has access to the private key, and how the corresponding X.509 certificate is distributed. The rightmost column (certificate signing) is discussed in the next section.

Use	Private Key	X.509 Certificate Distribution	X.509 Certificate Signing
Signing trust fabric document	MISE Management	Provided to each trusted system during on-boarding process.	MISE CA
Signing SAML assertions	Information consumer system which asserts user attributes	Included in trust fabric	Any well-known root CA
ISI SSL server certificate	ISI	Included in trust fabric	Any well-known root CA
Trusted system SSL client certificate	Trusted system	Included in trust fabric	Any well-known root CA

Table 1 - MISE X.509 Certificate Uses

### 2.1.2. MISE CERTIFICATE AUTHORITY (CA)

<sup>2</sup> See [http://en.wikipedia.org/wiki/Public-key\\_cryptography](http://en.wikipedia.org/wiki/Public-key_cryptography) for more background information.

<sup>3</sup> RSA stands for Rivest, Shamir, Adelman; the surnames of the computer scientist and two cryptographers who developed the algorithm.

The MISE Management operates a certificate authority (CA) to provide trust and security to the environment. The sole purpose of this CA is to sign the Trust Fabric. The CA does not issue certificates to trusted systems. Trust in the MISE is anchored by the Trust Fabric document and the MISE CA's signature of the document.

### 2.1.3. X.509 CERTIFICATES

Before a keypair can be used in the MISE for secure interaction, an X.509 certificate must be created containing the public key. An X.509 certificate binds a name to a public key value. The role of the certificate is to associate a public key with the identity contained in the X.509 certificate. All certificates must be digitally signed by a CA. A CA is a trusted entity that confirms the integrity of the public key value in a certificate. To be used in the MISE, an X.509 certificate must be signed by a well-known *root certificate authority*<sup>4</sup>. Any root CA trusted by all major browsers is acceptable.

### 2.1.4. CREATING X.509 CERTIFICATES

Numerous tools and processes are available for creating key pairs and X.509 certificates. The exact process chosen by a trusted system will vary depending on the platform the trusted system implementation is based upon, agency procedures, and the chosen root CA.

In some cases a trusted system will choose to generate a keypair and a certificate signing request (CSR) internally using a tool such as OpenSSL<sup>5</sup> or Java's keytool, and submit the CSR to the root CA for signing. In other cases, a trusted system may choose to use tools provided by the root CA for generation of the keypair in addition to signing the certificate. For step-by-step instructions, see the Implementation Guide at <https://mise.mda.gov>.

## 2.2. SSL CONNECTIONS

All MISE service invocations must take place over SSL network connections. Both server and client SSL certificates are required. The top portion of Figure illustrates establishing an SSL connection.

The following requirements are standard with SSL connections, and are part of any SSL implementation:

- The client side must validate the signature of the certificate presented by the server, confirm that it was signed by a root CA trusted by the client, and confirm that the server proves possession of the private key associated with the certificate.
- The client side must validate that the Subject common name (CN) within the server certificate matches the domain name portion of the service Uniform Resource Locator (URL) endpoint.
- The server side must validate the signature of the certificate presented by the client, confirm that it was signed by a root CA trusted by the server, and confirm the client proves possession of the private key associated with the certificate.

---

<sup>4</sup> See [http://en.wikipedia.org/wiki/Root\\_certificate](http://en.wikipedia.org/wiki/Root_certificate) for more background information.

<sup>5</sup> <http://www.openssl.org/>

---

Beyond these standard SSL requirements, the MISE requires:

- Each side of the SSL connection must confirm that the certificate presented by the opposite side exists within a **RoleDescriptor** element of the Trust Fabric, and must have **used="signing"** on the **KeyDescriptor**. (Details of Trust Fabric document format are presented in section 3.)
- SSL v3 or Transport Layer Security (TLS) 1.1 (and higher) must be used. TLS 1.2 is recommended. In addition, it is recommended that the TLS implementation conform to [NIST SP 800-52].

Taken together, these requirements guarantee a secure point-to-point communication channel between the client (trusted system) and server (ISI). In addition each side knows the identity of the other side, knows the other side is a current approved member of the MISE, and has all metadata in the trust fabric about the other side available to it.

SSL connections are typically reused for a series of service invocations (Figure illustrates this reuse). There is no requirement that a related series of service invocations take place over the same SSL connection; however, there is a requirement that the verification steps enumerated above be confirmed each time a new connection is established.

### 2.3. SAML ASSERTION PROCESSING

As illustrated in the lower portion of Figure , some MISE service invocations (e.g. Search, Retrieval) are done on behalf of an individual user, or a set of users with identical user attributes. In these cases, authenticated user attributes must be available to the service.

When user attributes are required, they are delivered using SAML assertions. In SAML terms, information consumer systems act as *identity providers*. Prior to invoking services on behalf of users, an information consumer system creates a SAML assertion in accordance with the guidelines specified in section 4.1 below. This SAML assertion contains authenticated *user attributes* pertaining to the user or group of users who will be granted access to the information obtained from the subsequent series of service invocations. The assertion is then digitally signed by the information consumer system using the private key associated with the signing certificate, which is a part of its **MISEConsumerDescriptor** role information within the trust fabric document. This provides a cryptographic guarantee to the ISI, and to information provider systems, of the identity of the information consumer system asserting the user attributes and promising to handle the information in accordance with MISE guidelines.

Rather than sending the SAML assertion with each service invocation, assertions are sent once at the beginning of a series of service invocations on behalf of a user or group of users. There is significant overhead associated with delivering the SAML assertion, cryptographically validating the digital signature, and validating the contents of the assertion document. Sending the assertion once to apply to a series of service invocations reduces the overall impact of this overhead. In addition, this pattern allows service interfaces to be fully RESTful. Request and response message bodies simply contain information associated with the service, without each interface needing to accommodate inclusion of a SAML assertion document.

Figure above illustrates the pattern used to allow a single SAML assertion to be bound to multiple service invocations. Key points related to this pattern include:

- 
- Some MISE services require user attributes, which are delivered in signed SAML assertions. Other services do not require user attributes. Whether or not a specific service requires user attributes is specified in the individual service documentation. If a service that requires user attributes is invoked without associating a SAML assertion, the service will return an HTTP status code of 403 (Forbidden) and a MISE error code of 104 (SAML assertion required but missing)(see section 2.4.1).
  - All service invocations, including the Login and Logout services, must occur over SSL connections validated against the trust fabric, with the identity of the invoking trusted system guaranteed as described in section 2.2 above.
  - Before invoking a service that requires user attributes, an information consumer system must create a SAML assertion and send it to the ISI using the Login service specified in section 2.3.1 below. The login service performs full validation of the assertion signature and contents, and creates a session context for subsequent service invocations to reference. The session key is returned to the information consumer system in a **Set-Cookie** HTTP response header.
  - The information consumer system may then proceed with any number of MISE service invocations on behalf of the user or group of users as asserted with the SAML assertion. Each of these service invocations **MUST** include a **Cookie** HTTP request header whose value is the session key set by the Login request. The information consumer system has the responsibility to ensure that information retrieved from the ISI will only be made available to users described in the SAML assertion.
  - Sessions and SAML assertions both have a defined lifetime. If a client sends a session key referencing a session which has expired, the MISE interface security implementation will not deliver user attributes to the service implementation; which means the error response the trusted system will see is the same as if a session key had not been sent. Specifically this will be an HTTP status code of 403 (Forbidden) and a MISE error code of 104 (SAML assertion required but missing)(see section 2.4.1). The client trusted system should respond to this error condition by invoking the login service again with a valid SAML assertion, then re-invoking the failed service request with the new session cookie returned by the login service.
  - Sessions expire after 20 minutes of inactivity. Any service invocation referencing the session resets this 20-minute timer. In addition, it is recommended that trusted systems invoke the Logout service (see section 0) if the trusted system is able to determine that the session will no longer be used. This allows the ISI to free resources used to store the session information prior to the 20-minute automatic expiration period.
  - SAML assertions expire when outside the time window expressed in the **NotBefore** and **NotOnOrAfter** attributes of the **<Conditions>** element. MISE sessions automatically expire when outside this time window, separately from the 20-minute inactivity timer.
  - An information consumer system will commonly serve many simultaneous users, and thus will commonly have many simultaneous open sessions with the ISI. It is the responsibility of the information consumer system to ensure that the correct session key is used when invoking services on behalf of a given user or group of users.
-

- Figure shows a series of service invocations operating in the context of a single SSL connection for simplicity of illustration; however this is not a requirement. Information consumer systems should keep SSL connection(s) to the ISI open during active communication to enhance performance by avoiding the overhead of establishing a fresh SSL connection for each service invocation. However, there is no requirement that all service invocations for a given session occur over the same SSL connection. In addition, service invocations for multiple sessions may take place over a single SSL connection. It IS REQUIRED that full SSL certificate validation back to the trust fabric take place every time a fresh SSL connection is established, so if a connection pooling system is used, trust fabric validation must be integrated with the connection pool.

### 2.3.1. LOGIN SERVICE INTERFACE

URI	From MISELoginService element in MISEInfrastructureDescriptor role in trust fabric.	Example: <a href="https://mda.gov/services/login">https://mda.gov/services/login</a>
Method	POST	
Request Headers		None specified
Request Content Type	application/xml	
Request Content	Signed SAML Assertion	
Status Codes	200 (OK)	Successful. Response content as described below.
	Other values	If error conditions are encountered, the response will be in accordance with section 0 below.
Response Headers	Set-Cookie: MISESession=xxxxxxx; Path=/ Path=	“xxxxxxx” will be replaced by a randomly-generated value unique to the session.
Response Content		Empty

Table 2 – Login Service Interface

### 2.3.2. LOGOUT SERVICE INTERFACE

URI	From MISELogoutService element in MISEInfrastructureDescriptor role in trust fabric.	Example: <a href="https://mda.gov/services/logout">https://mda.gov/services/logout</a>
Method	GET	
Request Headers	Cookie: MISESession=xxxxxxx	“xxxxxxx” is the key of the session to terminate, previously returned from Login service.
Request Content		Empty
Status Codes	200 (OK)	Successful. Response content as described below.
	Other values	If error conditions are encountered, the response

		will be in accordance with section 0 below.
Response Headers		None specified
Response Content		Empty

Table 3 – Logout Service Interface

## 2.4. MISE ERROR RESPONSE CONTENT

MISE service invocations which result in HTTP status codes in the 4xx (Client Error) and 5xx (Server Error) ranges may return an XML document as the response content providing additional details about the error encountered. The HTTP status is the important code to determine the server response, and the response content is provided only for debugging purposes. When such an error response document is returned, the response **Content-Type** is **application/xml**, and the response document is in the following format:

```
<MISEError>
  <Code>100</Code>
  <Description>Client certificate not presented during SSL handshake</Description>
</MISEError>
```

### 2.4.1. MISE ERROR CODES FOR INTERFACE SECURITY

Table 4 lists MISE error codes relevant to interface security, which pertain to all MISE services. Additional service-specific error codes may be defined in individual service specification documents.

MISE Error Code	HTTP Status Code Returned	Error Description
100	403	Client certificate not presented during SSL handshake
101	500	Internal server error accessing trust fabric
102	403	Client certificate not found in trust fabric
103	403	Session cookie not associated with trusted system
104	403	SAML assertion required but missing
201	400	SAML assertion signature validation failed
202	403	SAML signing certificate not in trust fabric
203	403	SAML signing certificate not associated with trusted system
204	400	SAML assertion issued by different entity than sender
205	400	MISE SAML assertions MUST NOT include a Subject
206	400	MISE SAML assertions MUST NOT include AuthnStatement
207	400	MISE SAML assertions MUST include Conditions element
208	400	NotBefore condition of assertion failed
209	400	NotOnOrAfter condition of assertion failed

210	400	MISE SAML assertions MUST include single AudienceRestriction element
211	400	MISE SAML assertions MUST include AudienceRestriction of 'urn:mise:all'
212	403	User attribute '<formalName>' disallowed by trust fabric
213	403	Asserting trusted system is not an information consumer system
299	500	Internal server error processing SAML assertion

Table 4 - MISE Interface Security Error Codes

## 2.5. TRUST FABRIC LIFECYCLE MANAGEMENT PROCEDURES

This section describes policies and procedures used to manage the MISE Cryptographic Trust Fabric. It includes details about how the Trust Fabric is created and distributed, as well as the conditions under which the Trust Fabric is updated.

### 2.5.1. TRUST FABRIC CREATION PROCEDURE

Upon occurrence of a triggering condition for a Trust Fabric update (see section 2.5.3), the Trust Fabric must be regenerated. The process of generating a new Trust Fabric document consists of two basic operations: editing the document to reflect the desired policy change (e.g., new trusted system added to the environment) and digitally signing the new document with the MISE CA private key. The following steps describe the process in more detail.

1. Starting with the most recent Trust Fabric document, edit the document as needed to incorporate the necessary changes.
2. Copy the edited Trust Fabric document to removable media.
3. Connect the removable media containing the unsigned Trust Fabric document to the physical machine on which the signing operation will be performed. Also connect the removable media containing the CA private key to the machine.
4. Perform the cryptographic signing operation on the Trust Fabric document using the CA private key. At no point during this operation shall the CA private key be copied from the removable media onto any other storage device. Also, at no point during this operation shall the physical machine be connected to a network.
5. Copy the signed Trust Fabric document onto the removable media that contains the unsigned Trust Fabric document.

### 2.5.2. TRUST FABRIC DISTRIBUTION PROCEDURE

Upon the occurrence of a triggering condition for a Trust Fabric update, and after the generation and signing of a new Trust Fabric document, the new Trust Fabric document must be distributed to all trusted systems. The following steps describe the process in more detail.

1. Publish the new Trust Fabric document at a well-known URL.
2. Notify all trusted systems of the new Trust Fabric document via the technical contact points they have provided.

Note that while the integrity of the Trust Fabric document is paramount to the security of the federation, the Trust Fabric need not necessarily be kept confidential. The security of the MISE does not rely on the contents of the trust fabric document being kept secret, but upon its

accuracy being guaranteed by the MISE CA signature. Therefore, it is permissible for the Trust Fabric URL to be publicly accessible, and encryption of the Trust Fabric document is not necessary.

### 2.5.3. TRIGGERING CONDITIONS FOR TRUST FABRIC UPDATES

The following events shall constitute cause for a Trust Fabric regeneration and redistribution.

1. A new trusted system joins the MISE.
9. An existing trusted system leaves the MISE.
10. An existing trusted system undergoes a configuration change that affects its entry in the trust fabric (e.g., certificate expiration, migration to a new server, key compromise on a server, etc.).
11. The MISE CA public key certificate expires.
12. It is suspected that the MISE CA private key has been compromised.

### 2.5.4. TRUSTED SYSTEM RETRIEVAL AND USAGE OF TRUST FABRIC

A trusted system implementation **MUST** retrieve the trust fabric document from the well-known URL when initially connecting to the MISE, and promptly when notified of a change by the MISE Management. It is **RECOMMENDED** that trusted systems be implemented in a manner that allows the trust fabric to be hot-reloaded while the trusted system is operational. In addition, it is **RECOMMENDED** that trusted system implementations automatically periodically retrieve the current trust fabric document from the well-known URL, and activate the new version in the running system if it has changed.

The following verification steps **MUST** be performed by the trusted system each time the trust fabric document is parsed and loaded into the trusted system for use, to ensure the trust fabric document put into use is indeed the official version created and signed by the MISE Management:

1. The digital signature contained within the trust fabric document **MUST** be validated.
2. The certificate used to sign the trust fabric **MUST** be compared against the MISE CA certificate, which is delivered to the trusted system upon joining the MISE by a separate out-of-band process.
3. HTTPS is **REQUIRED** to retrieve the trust fabric from the well-known URL. A client certificate is not required. This allows the trust fabric document to be retrieved for examination and supports a wide variety of trusted system administrative procedures.
4. The trusted system SSL configuration **MUST** validate the common name of the server certificate presented when connecting to the well-known URL against the domain name of the URL, and confirm the server certificate presented is signed by a CA trusted in the MISE (see section 2.2).

Sample code (written in Java) is available at <https://mise.mda.gov> to demonstrate loading, validating, and automatic periodic hot reloading of the trust fabric from a well-known URL. Since the trust fabric document is a SAML metadata file with a few simple extensions, this sample code is able to leverage the open source OpenSAML project to simplify implementation. Trusted system implementations not written in Java, or which already include other SAML

implementations, may also be able to simplify implementation by relying on existing SAML metadata implementations.

## 3. MISE Trust Fabric Document Format

### 3.1. TRUST FABRIC DOCUMENT SPECIFICATION

At a technical level, trust between all communications endpoints in the MISE is implemented using a combination of client and server TLS certificates, and the SAML 2.0 standard for federated system entity metadata. Information necessary to enforce trust is delivered to participants via the Trust Fabric document, which defines the most current cryptographic security context of the MISE. The document contains an **<md:EntityDescriptor>** entry for each communications endpoint in the environment, including the ISI, Information Provider Systems, and Information Consumer Systems. The MISE Management maintains the document and makes a new version of it available to trusted systems whenever the membership changes because of the addition or removal of a trusted system. To ensure compliance with the current Trust Fabric, each communications endpoint MUST incorporate the most current version of the Trust Fabric document into its security policy decisions in a timely fashion. The MISE Management will advise trusted systems of the urgency with which a new Trust Fabric document must be incorporated when the new document is made available. When the new Trust Fabric document is being published because of a security or trust violation, or because of the removal of a trusted system for disciplinary reasons, it is imperative that members incorporate the new Trust Fabric document as soon as is reasonably possible, and as a best practice not more than 24 hours after its release.

The MISE Trust Fabric document conforms to the specification defined in [SAML2 Metadata]. It also uses an extension schema for the **<md:RoleDescriptor>** element. This extension schema defines the three extensions to **RoleDescriptorType** listed below. These extensions are defined rather than using roles defined in SAML 2.0 specifications (such as **SPSSODescriptor** or **IDPSSODescriptor**) so that MISE roles and associated information can be stated explicitly in the trust fabric without implying characteristics of SAML service providers and identity providers which are not used in MISE.

1. **MISEInfrastructureDescriptor** - this role is only present within the **<md:EntityDescriptor>** entry defining the ISI.
2. **MISEConsumerDescriptor** - this role is present within the entry for any trusted system that acts as an Information Consumer System.
3. **MISEProviderDescriptor** - this role is present within the entry for any trusted system that acts as an Information Provider System.

Each **<md:EntityDescriptor>** must include at least one of these roles. A trusted system entry may include both the **MISEConsumerDescriptor** role and the **MISEProviderDescriptor** role.

Section 4.2 contains this extension schema, and Section 3.2 contains a sample Trust Fabric document conformant with these requirements.

### 3.1.1. SAML <ENTITIESDESCRIPTOR> ELEMENT REQUIREMENTS

The following additional requirements apply to the <EntitiesDescriptor> element, which is the top-level XML element within the MISE Trust Fabric document. These requirements supplement the requirements described in [SAML2 Metadata].

1. The Name attribute within <EntitiesDescriptor> MUST be present.
2. The validUntil attribute within <EntitiesDescriptor> MUST be present.
3. The <ds:Signature> element within <EntitiesDescriptor> MUST be present.
4. The <Extensions> element within <EntitiesDescriptor> MUST NOT be present.
5. Nested <EntitiesDescriptor> elements within the top-level <EntitiesDescriptor> MUST NOT be present.
6. One or more <EntityDescriptor> elements within <EntitiesDescriptor> MUST be present.

### 3.1.2. SAML <ENTITYDESCRIPTOR> ELEMENT REQUIREMENTS

The following requirements apply to <EntityDescriptor> elements that appear in the Trust Fabric document. Each <EntityDescriptor> element provides entity metadata for a specific communications endpoint (ISI or trusted system). These requirements supplement the requirements described in [SAML2 Metadata].

1. The entityID attribute within <EntityDescriptor> MUST be present, and MUST be set to the value that was agreed upon for this entity between the entity and the MISE Management. (The entity (trusted system) chooses its entityID value, but the choice MUST be approved by the MISE Management.)
2. The <ds:Signature> element within <EntityDescriptor> MUST NOT be present.
3. The <EntityDescriptor> element for the ISI MUST contain exactly one <RoleDescriptor> element of type MISEInfrastructureDescriptorType. The <EntityDescriptor> element for each trusted system must contain either a <RoleDescriptor> element of type MISEConsumerDescriptorType or a <RoleDescriptor> element of type MISEProviderDescriptorType, and may contain one of each.
4. Each <EntityDescriptor> element MUST contain at least one <ContactPerson> element with each technical contactType. An <EntityDescriptor> element MAY contain additional <ContactPerson> elements.
5. The following requirements apply to each <ContactPerson> element within an <EntityDescriptor> element.
  - a. The <Extensions> element MUST NOT be present.
  - b. The <Company> element MUST be present.
  - c. The <GivenName> element MUST be present.
  - d. The <SurName> element MUST be present.

- e. At least one **<EmailAddress>** element MUST be present.
- f. At least one **<TelephoneNumber>** element MUST be present.
6. The **<AdditionalMetadataLocation>** element within **<EntityDescriptor>** MUST NOT be present.
7. Each **<EntityDescriptor>** element MAY contain one **<Extensions>** element, and the **<Extensions>** element MAY contain one or more **<gfipm:EntityAttribute>** elements as defined by the GFIPM Entity Attribute Extension Schema.

### 3.1.3. SAML **<ROLEDESCRIPTOR>** ELEMENT REQUIREMENTS

**RoleDescriptor** types defined by the MISE trust fabric extension schema are instantiated in the trust fabric document by specifying the **xsi:type** attribute on a **<RoleDescriptor>** element. For example:

```
<md:RoleDescriptor xsi:type="mise:MISEConsumerDescriptorType"
  protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  ...
</md:RoleDescriptor>
```

Requirements for each MISE extension role are detailed in the following subsections.

#### MISEINFRASTRUCTUREDESCRIPTOR ROLE REQUIREMENTS

1. The **xsi:type** attribute within **<RoleDescriptor>** MUST be present, and MUST have “**mise:MISEInfrastructureDescriptorType**” as its value.
2. The **protocolSupportEnumeration** attribute within **<RoleDescriptor>** MUST be present, and MUST have “**urn:oasis:names:tc:SAML:2.0:protocol**” as its value.
3. The **<ds:Signature>** element within **<RoleDescriptor>** MUST NOT be present.
4. One or more **<KeyDescriptor>** elements containing a use attribute with a value of “**signing**” MUST be present within **<RoleDescriptor>**.
5. Exactly one **<ds:KeyInfo>** element MUST be present within each **<KeyDescriptor>** element. Exactly one **<ds:X509Data>** element MUST be present within the **<ds:KeyInfo>** element. Exactly one **<ds:X509Certificate>** element MUST be present within the **<ds:X509Data>** element.
6. The **<MISELoginService>** element within **<RoleDescriptor>** MUST be present, and must contain a Binding attribute with a value of “**urn:mise:bindings:REST**”.
7. The **<MISELogoutService>** element within **<RoleDescriptor>** MUST be present, and must contain a Binding attribute with a value of “**urn:mise:bindings:REST**”.
8. The **<MISESearchService>** element within **<RoleDescriptor>** MUST be present, and must contain a Binding attribute with a value of “**urn:mise:bindings:REST**”.

#### MISECONSUMERDESCRIPTOR ROLE REQUIREMENTS

1. The **xsi:type** attribute within **<RoleDescriptor>** MUST be present, and MUST have “**mise: MISEConsumerDescriptorType**” as its value.

2. The **protocolSupportEnumeration** attribute within **<RoleDescriptor>** MUST be present, and MUST have “urn:oasis:names:tc:SAML:2.0:protocol” as its value.
3. The **<ds:Signature>** element within **<RoleDescriptor>** MUST NOT be present.
4. One or more **<KeyDescriptor>** elements containing a use attribute with a value of “signing” MUST be present within **<RoleDescriptor>**.
5. Exactly one **<ds:KeyInfo>** element MUST be present within each **<KeyDescriptor>** element. Exactly one **<ds:X509Data>** element MUST be present within the **<ds:KeyInfo>** element. Exactly one **<ds:X509Certificate>** element MUST be present within the **<ds:X509Data>** element.

#### MISEPROVIDERDESCRIPTOR ROLE REQUIREMENTS

1. The **xsi:type** attribute within **<RoleDescriptor>** MUST be present, and MUST have “mise: MISEProviderDescriptorType” as its value.
2. The **protocolSupportEnumeration** attribute within **<RoleDescriptor>** MUST be present, and MUST have “urn:oasis:names:tc:SAML:2.0:protocol” as its value.
3. The **<ds:Signature>** element within **<RoleDescriptor>** MUST NOT be present.
4. One or more **<KeyDescriptor>** elements containing a use attribute with a value of “signing” MUST be present within **<RoleDescriptor>**.
5. Exactly one **<ds:KeyInfo>** element MUST be present within each **<KeyDescriptor>** element. Exactly one **<ds:X509Data>** element MUST be present within the **<ds:KeyInfo>** element. Exactly one **<ds:X509Certificate>** element MUST be present within the **<ds:X509Data>** element.

## 3.2. SAMPLE TRUST FABRIC DOCUMENT

The diagram below contains a sample of the Trust Fabric Document provided by the MISE Management to each trusted system during the onboarding process.

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntitiesDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
  xmlns:gfirm="http://gfirm.net/standards/metadata/2.0/entity"
  xmlns:mise="http://mda.gov/standards/trustfabric/1.0"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  Name="Maritime Information Sharing Environment Trust Fabric"
  validUntil="2015-11-15T00:00:00.000Z"
  xsi:schemaLocation="urn:oasis:names:tc:SAML:2.0:metadata saml20/saml-schema-
metadata-2.0.xsd http://mda.gov/standards/trustfabric/1.0 mise-trust-fabric-
extension.xsd http://gfirm.net/standards/metadata/2.0/entity gfirm-entity-attribute-
2.0.xsd">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="">
        <ds:Transforms>
```

```

signature" />
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
      <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-
exc-c14n#" PrefixList="mise xs" />
    </ds:Transform>
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>q64PXRBBjnoTNL3Bg4ShLDSPrBw=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue><!-- Base 64 encoded signature--></ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data>
    <ds:X509Certificate>
      <!-- Base 64 encoded certificate embedded here
      This is the MISE CA certificate
      used to sign the trust fabric document.
      -->
    </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<md:EntityDescriptor entityID="https://isi.mda.gov/">
  <md:RoleDescriptor xsi:type="mise:MISEInfrastructureDescriptorType"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <md:KeyDescriptor use="signing">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:X509Data>
        <ds:X509Certificate>
          <!-- Base 64 encoded certificate embedded here
          This is the server certificate which the ISI will present
          during SSL connection handshake.-->
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </md:KeyDescriptor>
  <md:KeyDescriptor use="signing">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:X509Data>
        <ds:X509Certificate>
          <!-- Base 64 encoded certificate embedded here
          This is the MISE CA
          certificate used to sign the trust fabric document. -->
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </md:KeyDescriptor>
  <mise:MISELoginService Binding="urn:mise:bindings:REST"
  Location="https://isi.mda.gov/service/login" />
  <mise:MISELogoutService Binding="urn:mise:bindings:REST"
  Location="https://isi.mda.gov/service/logout" />
  <mise:MISESearchService Binding="urn:mise:bindings:REST"
  Location="https://isi.mda.gov/service/search" />
</md:RoleDescriptor>
<md:Organization>
  <md:OrganizationName xml:lang="en">MISE
</md:OrganizationName>
  <md:OrganizationDisplayName xml:lang="en">Maritime
  Information Sharing Environment</md:OrganizationDisplayName>
  <md:OrganizationURL xml:lang="en">http://www.mda.gov
</md:OrganizationURL>

```

```

</md:Organization>
<md:ContactPerson contactType="technical">
  <md:Company>SPAWAR Systems Center Pacific</md:Company>
  <md:GivenName>Olithia</md:GivenName>
  <md:SurName>Strom</md:SurName>
  <md:EmailAddress>olithia.strom@navy.mil</md:EmailAddress>
  <md:TelephoneNumber>619-553-0728</md:TelephoneNumber>
</md:ContactPerson>
</md:EntityDescriptor>
<md:EntityDescriptor entityID="https://mise.agencyone.gov/">
  <md:Extensions>
    <gfipm:EntityAttribute FriendlyName="COIIndicator"
      Name="mise:1.4:entity:COIIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
      <gfipm:EntityAttributeValue xsi:type="xs:string">True
    </gfipm:EntityAttributeValue>
    </gfipm:EntityAttribute>
    <gfipm:EntityAttribute FriendlyName="LawEnforcementIndicator"
      Name="mise:1.4:entity:LawEnforcementIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
      <gfipm:EntityAttributeValue xsi:type="xs:string">True
    </gfipm:EntityAttributeValue>
    </gfipm:EntityAttribute>
    <gfipm:EntityAttribute FriendlyName="PrivacyProtectedIndicator"
      Name="mise:1.4:entity:PrivacyProtectedIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
      <gfipm:EntityAttributeValue xsi:type="xs:string">True
    </gfipm:EntityAttributeValue>
    </gfipm:EntityAttribute>
    <gfipm:EntityAttribute FriendlyName="OwnerAgencyCountryCode"
      Name="mise:1.4:entity:OwnerAgencyCountryCode"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
      <gfipm:EntityAttributeValue xsi:type="xs:string">USA
    </gfipm:EntityAttributeValue>
    </gfipm:EntityAttribute>
  </md:Extensions>
  <md:RoleDescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
  xsi:type="mise:MISEConsumerDescriptorType">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            <!-- Base 64 encoded certificate embedded here
              This is the client certificate which the trusted
              system will present during SSL connection handshake.
              The private key matching this certificate will also
              be used by this trusted system for signing SAML
              assertions.
            -->
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
  </md:RoleDescriptor>
</md:Organization>
  <md:OrganizationName xml:lang="en">Agency One
</md:OrganizationName>
  <md:OrganizationDisplayName xml:lang="en">Agency
  One</md:OrganizationDisplayName>
  <md:OrganizationURL xml:lang="en">http://www.agencyone.gov/
</md:OrganizationURL>
</md:Organization>

```

```

    <md:ContactPerson contactType="technical">
      <md:Company>SPAWAR Systems Center Pacific</md:Company>
      <md:GivenName>Olithia</md:GivenName>
      <md:SurName>Strom</md:SurName>
      <md:EmailAddress>olithia.strom@navy.mil</md:EmailAddress>
      <md:TelephoneNumber>619-553-0728</md:TelephoneNumber>
    </md:ContactPerson>
  </md:EntityDescriptor>
  <md:EntityDescriptor entityID="https://mise.agencythree.gov/">
    <md:RoleDescriptor xsi:type="mise:MISEProviderDescriptorType"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <md:KeyDescriptor use="signing">
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <ds:X509Data>
            <ds:X509Certificate>
              <!-- Base 64 encoded certificate embedded here
                This is the client certificate which the trusted
                system will present during SSL connection handshake.
              -->
            </ds:X509Certificate>
          </ds:X509Data>
        </ds:KeyInfo>
      </md:KeyDescriptor>
    </md:RoleDescriptor>
    <md:ContactPerson contactType="technical">
      <md:Company>SPAWAR Systems Center Pacific</md:Company>
      <md:GivenName>Olithia</md:GivenName>
      <md:SurName>Strom</md:SurName>
      <md:EmailAddress>olithia.strom@navy.mil</md:EmailAddress>
      <md:TelephoneNumber>619-553-0728</md:TelephoneNumber>
    </md:ContactPerson>
  </md:EntityDescriptor>
</md:EntitiesDescriptor>

```

## 4. MISE SAML Assertion Format

### 4.1. MISE SAML ASSERTION SPECIFICATION

SAML Assertions are used to convey user attribute information from information consumer systems to the ISI. This section contains normative language that describes MISE-specific requirements that apply to any SAML assertion generated by an information consumer system for use in MISE services. These requirements augment the SAML assertion format requirements that appear in the SAML 2.0 specification ([SAML2 Core]).

1. The **<Assertion>** element MUST be signed, MUST NOT be encrypted, and MUST be the root element. The **<EncryptedAssertion>** element is not used in MDA. Assertions are always signed and transmitted over SSL, but XML encryption is not used.
2. The **Version** attribute within **<Assertion>** MUST have “2.0” as its value.
3. The **<Issuer>** element within **<Assertion>** MUST be present, and its value MUST match the **entityID** in the trust fabric of the information consumer system initiating the service request sequence on behalf of a user.

4. The **<ds:Signature>** element MUST be present, and the **<X509Certificate>** within the **<KeyInfo>** element MUST be one of the signing certificates associated with the issuer in the trust fabric.
5. The **<Subject>** element MUST NOT be present.
6. The **<Conditions>** element MUST be present, and MUST contain the NotBefore and NotOnOrAfter attributes.
7. The **<AudienceRestriction>** element within **<Conditions>** MUST be present, and MUST contain an **<Audience>** element with the value urn:mise:all.
8. An **<Assertion>** element MUST NOT contain an **<AuthnStatement>** element.
9. An **<Assertion>** element MUST NOT contain an **<AuthzDecisionStatement>** element.
10. An **<Assertion>** element MUST contain exactly one **<AttributeStatement>** element.
11. The **<AttributeStatement>** element in an **<Assertion>** MAY contain one or more **<Attribute>** elements and MUST NOT contain any **<EncryptedAttribute>** elements.
12. Each **<Attribute>** element MAY contain application-level user attribute data corresponding to a MISE user attribute defined in [MISE Attributes].
13. If the **<Attribute>** element corresponds to a MISE user attribute defined in [MISE attributes], then the Name attribute within the **<Attribute>** element MUST contain the fully qualified formal name of the attribute as defined in [MISE Attributes].
14. Each **<Attribute>** element MUST contain one or more **<AttributeValue>** elements.
15. Each **<AttributeValue>** element MUST contain the following attribute name/value pairs:
  - a. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  - b. xsi:type="xs:string"
16. Each **<AttributeValue>** element MUST contain data corresponding to the value of the MISE user attribute represented by its enclosing **<Attribute>** element.

## 4.2. EXTENSION SCHEMA FOR <MD:ROLEDESCRIPTOR>

The diagram below contains the SAML Metadata extension schema for the <md:RoleDescriptor> element, which allows MISE roles and associated information to be stated explicitly in the trust fabric document.

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema xmlns=http://www.w3.org/2001/XMLSchema
  xmlns:mise=http://mda.gov/standards/trustfabric/1.0
  xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:ds=http://www.w3.org/2000/09/xmldsig#
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  targetNamespace=http://mda.gov/standards/trustfabric/1.0
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  version="1.0">
  <import namespace="urn:oasis:names:tc:SAML:2.0:metadata"
    schemaLocation="saml20/saml-schema-metadata-2.0.xsd"/>
  <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
    schemaLocation="saml20/saml-schema-assertion-2.0.xsd"/>
```

```

<import namespace=http://www.w3.org/2000/09/xmldsig#
  schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-
core-schema.xsd"/>

  <element name="MISEInfrastructureDescriptor"
type="mise:MISEInfrastructureDescriptorType"/>
  <element name="MISELoginService" type="md:EndpointType"/>
  <element name="MISELogoutService" type="md:EndpointType"/>
  <element name="MISESearchService" type="md:EndpointType"/>
  <complexType name="MISEInfrastructureDescriptorType">
    <complexContent>
      <extension base="md:RoleDescriptorType">
        <sequence>
          <element ref="mise:MISELoginService"/>
          <element ref="mise:MISELogoutService"/>
          <element ref="mise:MISESearchService"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <element name="MISEConsumerDescriptor" type="mise:MISEConsumerDescriptorType"/>
  <complexType name="MISEConsumerDescriptorType">
    <complexContent>
      <extension base="md:RoleDescriptorType"/>
    </complexContent>
  </complexType>

  <element name="MISEProviderDescriptor" type="mise:MISEProviderDescriptorType"/>
  <complexType name="MISEProviderDescriptorType">
    <complexContent>
      <extension base="md:RoleDescriptorType"/>
    </complexContent>
  </complexType>
</schema>

```

### 4.3. SAMPLE SAML ASSERTION

The diagram below contains a sample SAML assertion, which provides a cryptographic guarantee of the identity of the trusted system asserting the user attributes and promising to handle the information in accordance with MISE guidelines.

```

<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  ID="_1025e5dabb24f891e338c4d38171982e" IssueInstant="2012-12-05T14:50:21.085Z"
  Version="2.0">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://mise.agencyone.gov/</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="#_1025e5dabb24f891e338c4d38171982e">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-
exc-c14n#" PrefixList="xs" />
          </ds:Transform>
        </ds:Transforms>
      </ds:Reference>
    </ds:SignedInfo>
  </ds:Signature>

```

```

        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>qepBuvjTTzrg+I7YTHes8nxPFY=</ds:DigestValue>
    </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue><!-- Base 64 encoded signature --></ds:SignatureValue>
<ds:KeyInfo>
    <ds:X509Data>
        <ds:X509Certificate>
            <!-- Base 64 encoded certificate embedded here
                This is the certificate of the information consumer system
                which signed the assertion.
            -->
        </ds:X509Certificate>
    </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<saml2:Conditions NotBefore="2012-12-05T14:50:16.085Z" NotOnOrAfter="2012-12-
05T15:00:21.085Z">
    <saml2:AudienceRestriction>
        <saml2:Audience>urn:mise:all</saml2:Audience>
    </saml2:AudienceRestriction>
</saml2:Conditions>
<saml2:AttributeStatement>

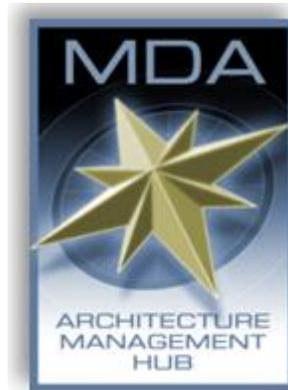
    <saml2:Attribute FriendlyName="ElectronicIdentityId"
Name="gfipm:2.0:user:ElectronicIdentityId"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">eric.jakstadt@trustedfederal.com</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute FriendlyName="FullName" Name="gfipm:2.0:user:FullName"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">Eric G. Jakstadt</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute FriendlyName="CitizenshipCode"
Name="mise:1.4:user:CitizenshipCode" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
        <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">USA</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute FriendlyName="LawEnforcementIndicator"
Name="mise:1.4:user:LawEnforcementIndicator"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">true</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute FriendlyName="PrivacyProtectedIndicator"
Name="mise:1.4:user:PrivacyProtectedIndicator"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">true</saml2:AttributeValue>
    </saml2:Attribute>
</saml2:AttributeStatement>
</saml2:Assertion>

```

# APPENDIX D - PUBLICATION INTERFACE SPECIFICATION

---

---



## Maritime Information Sharing Environment (MISE)

Publication Interface Specification

Version 1.0

25 March 2013

# 1. Publication Overview

As discussed in the *National MDA Architecture*, information providers can choose between two integration approaches. In Figure 2, below, Information Provider System B chose to publish information to the Information Sharing Infrastructure (ISI) cache and delegate to the ISI the work of responding to access requests on behalf of users from information-consumer systems. This specification presents the details of the *Publication* interface used by a provider such as Figure 2's Provider B to keep the ISI cache up to date as a data set changes over time.

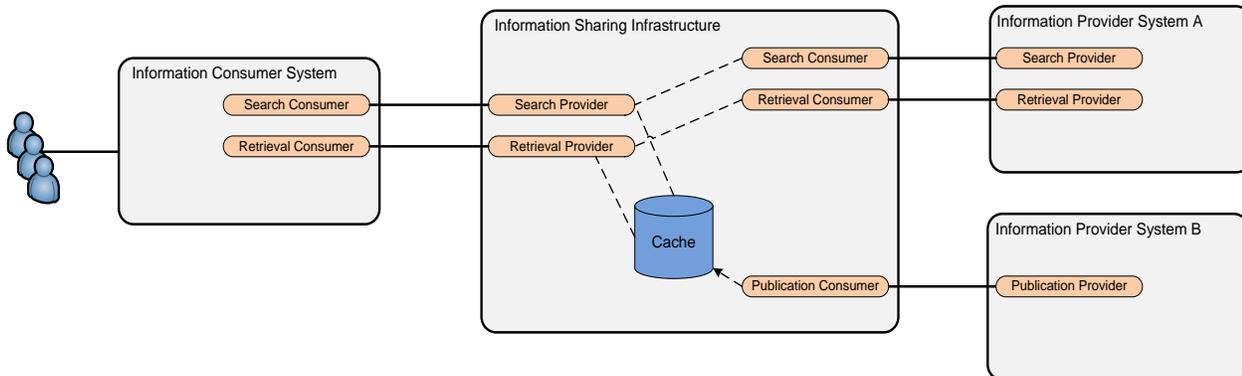


Figure 2 - Service Providers and Consumers

The publication interface follows the Representational State Transfer (REST) style. The ISI defines a URI endpoint for publication, and information-provider systems send HTTP requests and receive responses to URI paths beneath this URI endpoint. Details of these HTTP request and response messages are covered in Section 3.

All messages are authenticated and secured in the manner described in the *MISE Interface Security Specification*. For publication, the *Trusted System Authentication* portion of the interface security specification applies, but the *User Attribute Conveyance* portion does not apply since publication is not done on behalf of any individual user.

Each published data set must have an associated *Information Access Policy (IAP)*. This IAP information is carried in each record published to the ISI. For more information on the IAP, see the *MISE Attribute Specification* and the *National MDA Information Exchange Package Documents (IEPD)*. The IAP is carried via attributes in the XML documents published to the ISI.

## 2. Message Flow Patterns

This section describes the sequence of HTTP request/response messages that is expected to occur during normal operation. These processes help keep the ISI cache up to date.

### 2.1. INITIAL FULL PUBLICATION

Before publishing a data set, an information provider will work with MISE management to configure the trusted system and data set at the ISI. This includes creating an IAP for the data

set. Once these steps are complete, the information provider should do an initial full publication of the data set to the ISI, meaning all records that the information provider wishes to share should be sent to the ISI cache using the publication interface described herein.

The first step is to HTTP GET the version resource (see Section 3.1.1). If the version resource cannot be read, or if the version number is not one that the information provider is implemented to support, no further interaction with the interface should be taken.

After that, the information provider should HTTP PUT each individual shareable record (see Section 3.2.1). During this process, if the information provider fails to connect to the ISI, or if the ISI returns a status code indicating a server-side error (5xx), the information provider should periodically retry the PUT until it succeeds and contact the MISE helpdesk if errors persist.

If the information provider's back-end data store changes during the initial full publication, the provider system must track that and ensure that all shareable records have been successfully PUT to the ISI before the initial full publication process is considered complete.

Note that since the publication interface follows HTTP RESTful semantics, no harm is done if the information provider PUTs the same record more than once. The HTTP terminology for this characteristic is *idempotence*. Therefore, during any kind of error-recovery scenario, the provider is free to PUT the record if there is any question whether a previous PUT succeeded.

## 2.2. ONGOING UPDATING

After the initial full publication is complete, the information provider should update the ISI cache whenever any changes occur to the set of shareable records.

As with initial full publication, the information provider should periodically HTTP GET the version resource to confirm that the version of the publication interface is compatible. This does not necessarily need to be done before each record update, but it should be checked frequently—at least daily. The version resource is defined to support If-Modified-Since to make the check highly efficient.

Whenever any shared record is added or changed in the provider's back-end data store, the full record should be sent to the ISI using HTTP PUT (see Section 3.2.1). Whenever any shared record is deleted from the provider's back-end data store, the provider should use HTTP DELETE (see Section 3.2.2) to delete the record from the ISI cache. If the information provider fails to connect to the ISI, or if the ISI returns a status code indicating a server-side error (5xx), the information provider should periodically retry the PUT or DELETE until it succeeds and contact the MISE helpdesk if errors persist. Note that the HTTP DELETE method is also idempotent, so the provider is free to DELETE the record if there is any question whether a previous DELETE succeeded.

## 3. Resource Reference

This section presents details of resources defined as part of the publication interface and aspects of HTTP protocol usage that are important to interoperability.

All URIs are defined relative to *<BaseURI>*, which represents the HTTP endpoint of the publication interface at the ISI. The physical URI will be provided when the trusted system is

integrated with the ISI. For example only, this specification uses <https://isi.gov/publish> as the base URI.

### 3.1. METADATA

#### 3.1.1. INTERFACE VERSION

The version of the MISE publication interface described in this document is 1.0. Before interacting further, an information provider should GET the version resource to confirm that the ISI interface version matches what is expected by the information provider implementation.

Table 1 presents details of the HTTP GET request an information provider uses to retrieve the version resource and the possible responses, which may be received back from the ISI.

URI	<BaseURI>/version	Example: <a href="https://isi.gov/publish/version">https://isi.gov/publish/version</a>
Method	GET	
Request Headers	Authorization	As described in the <i>MISE Interface Security Specification</i> .
	If-Modified-Since	Information provider system may send this header if it has previously read and cached this resource.
Request Content Type		Empty
Status Codes	200 (OK)	Successful. Response content as described below.
	304 (Not Modified)	The version resource has not been modified since the time specified in the If-Modified-Since request header.
	401 (Unauthorized)	No Authorization header or userid/password does not match any trusted system.
Response Headers	Last-Modified	
Response Content Type	application/xml; charset=UTF-8	Response content only returned if status code is 200.
Response Content	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;MISEInterface&gt;   &lt;Name&gt;Publication&lt;/Name&gt;   &lt;MajorVersion&gt;1&lt;/MajorVersion&gt;   &lt;MinorVersion&gt;0&lt;/MinorVersion&gt; &lt;/MISEInterface&gt;</pre>	

Table 1 – Version Resources

### 3.2. RECORD

The record resource represents an individual record as stored in the ISI cache.

The URI for an individual record is <BaseURI>/<IEPDName>/<RecordID>, where:

- *<IEPDName>* matches one of the three currently defined national information products (noa, ian, pos).
- *<RecordID>* is assigned by the information provider. It is unique to an individual record within the scope of the provider’s data set.

For example, a full record URI may be similar to <https://isi.gov/publish/noa/12345>.

Before a record URI may be accessed, the ISI must be configured to accept a particular IEPD type from the provider. As discussed in the *MISE Interface Security Specification*, all HTTP requests to the ISI are authenticated, so the ISI knows which provider data set the record URI refers to—even though that information is not encoded within the URI. Records associated with a particular information provider cannot be accessed in any way through the publication interface by any other information provider.

### 3.2.1. ADD OR UPDATE A RECORD

Table 2 presents details of the HTTP PUT request an information provider uses to add or update a record and the possible responses, which may be received back from the ISI.

URI	<i>&lt;BaseURI&gt;/&lt;IEPDName&gt;/&lt;RecordID&gt;</i>	Example: <a href="https://isi.gov/publish/noa/12345">https://isi.gov/publish/noa/12345</a>
Method	PUT	
Request Headers	Authorization	As described in the <i>MISE Interface Security Specification</i> .
Request Content Type	application/xml; charset=UTF-8	
Request Content	NIEM-M representation of record.	Must validate against the schema for <i>IEPDName</i> but need not include a <i>schemaLocation</i> attribute. The XML declaration should specify UTF-8 encoding.
Status Codes	201 (Created)	Record did not previously exist at ISI and was successfully added.
	204 (No Content)	Record previously existed at ISI and was successfully updated.
	400 (Bad Request)	Request content did not validate against the schema for <i>IEPDName</i> .
	401 (Unauthorized)	No Authorization header, or <i>userid/password</i> does not match any trusted system.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for publication of this IEPD.
Response Headers	Location	Only sent with status code 201. Matches record URI of request.

Response Content Type	Empty
-----------------------	-------

Table 2 – Adding or Updating Resources

### 3.2.2. DELETE A RECORD

Table 3 presents details of the HTTP DELETE request an information provider uses to delete a record and the possible responses, which may be received back from the ISI.

URI	<BaseURI>/<IEPDName>/<RecordID>	Example: https://isi.gov/publish/noa/12345
Method	DELETE	
Request Headers	Authorization	As described in the <i>MISE Interface Security Specification</i> .
Request Content Type	Empty	
Status Codes	204 (No Content)	Record was successfully deleted if it existed.
	401 (Unauthorized)	No Authorization header, or userid/password does not match any trusted system.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for publication of this IEPD.
Response Content Type	Empty	

Table 3 – Deleting Records

### 3.2.3. PUBLISHING WITH A DATA SCOPE

For specific events and situations, the ISI provides the means for a data provider to specify a different level of data access. For instance, a data provider might allow temporary access to vessel position data for a wider range of data consumers during a hurricane. To publish data in a specific scope, the data provider must provide the **Scope** and **DataAttribute** parameters. A new record can be published with scope, or an existing record can be updated with a new scope.

URI	<BaseURI>/<IEPDName>/<RecordID>?<Scope=XXX>&<DataAttribute=YYY>&<Releasable=B>&<Nation=NNN,MM>M	Example: <a href="https://isi.gov/publish/noa/12345?Scope=HurricaneKatrina&amp;DataAttribute=COI&amp;Releasable=F&amp;Nation=USA">https://isi.gov/publish/noa/12345?Scope=HurricaneKatrina&amp;DataAttribute=COI&amp;Releasable=F&amp;Nation=USA</a>
Scope	String defining the scope in which this record has modified data access. These are defined by the MISE Board for specific events	HurricaneKatrina
DataAttribute	This is the modified data access attribute for that scope, as defined in the <i>MISE Attribute Specification</i> .	COI
Releasable	Optional. Boolean. Indicates whether the data is	T or F

	releasable within the scope.	
Nation	Optional. Comma-separated list of ISO 3-letter country codes. Indicates which nations to which the data can be provide in this scope.	USA,CAN
Method	PUT	
Request Headers	Authorization	As described in the <i>MISE Interface Security Specification</i> .
Request Content Type	application/xml; charset=UTF-8	
Request Content	NIEM-M representation of record.	Must validate against the schema for <i>IEPDName</i> but need not include a <i>schemaLocation</i> attribute. The XML declaration should specify UTF-8 encoding.
Status Codes	201 (Created)	Record did not previously exist at ISI and was successfully added.
	204 (No Content)	Record previously existed at ISI and was successfully updated.
	400 (Bad Request)	Request content did not validate against the schema for <i>IEPDName</i> .
	401 (Unauthorized)	No Authorization header, or userid/password does not match any trusted system.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for publication of this IEPD.
Response Headers	Location	Only sent with status code 201. Matches record URI of request.
Response Content Type	Empty	

Table 4 – Adding or Updating a Record with Data Scope

### 3.2.4. RECORD EXPIRATION

Records in the ISI cache will always be considered expired and be deleted after 30 days in the cache. Data providers can exercise more precise control over their records via the following means:

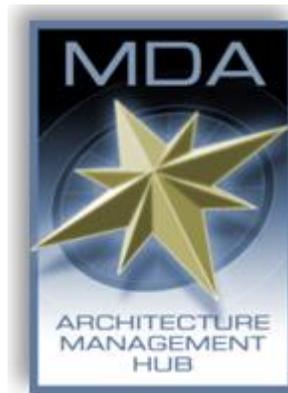
1. The DELETE interface described in 3.2.2, above.

13. Each IEPD includes the DocumentExpirationDate element. When set to a value less than 30 days, the record will be expired and deleted from the cache when that date is reached. If set to more than 30 days, it will be ignored and the record is deleted at 30 days.

# APPENDIX E - SEARCH/RETRIEVE INTERFACE SPECIFICATION

---

---



## Maritime Information Sharing Environment (MISE)

Search/Retrieve Interface Specification  
Version 1.0

25 March 2013

# 1. Introduction

The Maritime Information Sharing Environment (MISE) is being established to allow secure sharing of unclassified information among partners within the MDA community of interest. The Information Sharing Infrastructure (ISI) is the information and request broker for the MISE.

This document describes the data-consumer facing representational state transfer (REST) architecture providing search and retrieve (SR) functionality for the MISE. By conforming to this interface, the ISI provides data consumers with the ability to find and retrieve the right information at the right time, based on the needs, rights, and authorities of the user and the organizations requesting the information. This document outlines the base level of functionality for the SR interface. The REST interface is divided logically into two parts. The search interface provides query endpoints and an associated set of query arguments, returning a list of results and summaries of each result. The retrieval interface provides the ability to access full records. The two interfaces work together to provide access to all information in the ISI.

All interactions with the SR interface are secured as described by the *MISE Interface Security Specification*. For search and retrieve operations, both the *Trusted System Authentication* and the *User Attribute Conveyance* portions of the interface security specification apply. All records available on the ISI are formatted using NIEM-Maritime, as described in the *National MDA Architecture Information Exchange Package Documents (IEPD)*. For further details on interfacing with the SR interface, see the *MISE Implementation Guide*.

## 2. General Consumer Search Interface

### 2.1. URL STRUCTURE AND QUERY RESULTS

For the purposes of this document, the ISI is assumed to be accessible at the global uniform resource identifier (URI) <https://mise.mda.gov/services/MDAService>, provided as an example base URL. SR provides a series of URL endpoints that provide global query functionality and focus-area-specific (or IEPD-specific) queries. The query URLs have the following form:

- [https://mise.mda.gov/services/MDAService/search/<iepdname>?=-](https://mise.mda.gov/services/MDAService/search/<iepdname>?=)

The <iepdname> takes the form of one of the message types provided by the ISI. Currently, these are:

1. noa (Notice of Arrival)
2. ian (Indicator and Notification)
3. pos (Position)
4. loa (Levels of Awareness)

Each URI in association with an IEPD name queries a single record type. For example, [https://mise.mda.gov/services/MDAService/search/noa?=-](https://mise.mda.gov/services/MDAService/search/noa?=) provides the notice-of-arrival specific

---

query endpoint. Using this scheme, further endpoints can be added for specific queries as new record types are defined for new focus areas.

All queries to the search interface return an Atom feed that contain summaries of the record or records matching that query. The following listing shows an example feed returned from a search for Position interface. The search that returns this result takes the following form:

```
https://mise.mda.gov/services/MDAService/search/track?start=2012-08-19T11:40:00&end=2013-12-30T11:40:00&ulat=3.75&ulng=-2.0&llat=-2.75&llon=3.0
```

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <title>Query Response for Entity: https://mise.agencyone.gov/</title>
  <link rel="alternate"
href="https://mise.mda.gov/services/MDAService/search/track?start=2012-08-19T11:40:00&end=2013-12-30T11:40:00&ulat=3.75&ulng=-2.0&llat=-2.75&llon=3.0" />
  <subtitle>Query Response from the Maritime Information Sharing Environment</subtitle>
  <id>TRACK-9991956437ffa1c06327e0d98977989f4f7c6f46b1f3c52e3ac11d04d45273c3</id>
  <entry>
    <title>TRACK</title>
    <![CDATA[<link rel="alternate"
href=https://mise.mda.gov/services/MDAService/retrieve/ian?entityid=https%3A%2F%2Fmise.agencyone.gov%2F&recordid=79869882775656568/>]]>
    <author>
      <name>DCMP</name>
    </author>
    <id>8397109112108101736578</id>
    <updated>2012-12-31T19:28:08Z</updated>
    <summary type="text/xml">
      <posex:Message
xmlns:posex="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2"
      mda:securityIndicatorText="LEI" mda:releasableNationsCode="USA"
mda:releasableIndicator="true"
      xmlns:m="http://niem.gov/niem/domains/maritime/2.1"
      xmlns:mda="http://niem.gov/niem/domains/maritime/2.1/mda/3.2"
      xmlns:nc="http://niem.gov/niem/niem-core/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
      xmlns:ism="urn:us:gov:ic:ism"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <mda:Vessel>
          <m:VesselAugmentation>
            <m:VesselIMONumberText>IMO0000001</m:VesselIMONumberText>
            <m:VesselName>MV Example</m:VesselName>
          </m:VesselAugmentation>
          <m:VesselNationalFlagISO3166Alpha3Code>USA</m:VesselNationalFlagISO3166Alpha3Code>
        </mda:Vessel>
        <mda:Position>
          <m:LocationPoint>
            <gml:Point gml:id="tp1">
              <gml:pos>-1.0 -1.0</gml:pos>
            </gml:Point>
          </m:LocationPoint>
          <mda:PositionSpeedMeasure>
            <nc:MeasureText>12</nc:MeasureText>
            <nc:SpeedUnitCode>kt</nc:SpeedUnitCode>
          </mda:PositionSpeedMeasure>
          <mda:PositionCourseMeasure>
            <nc:MeasureText>180</nc:MeasureText>
          </mda:PositionCourseMeasure>
        </mda:Position>
      </posex:Message>
    </summary>
  </entry>
</feed>
```

```

        <m:AngleUnitText>deg</m:AngleUnitText>
      </mda:PositionCourseMeasure>
      <mda:PositionHeadingMeasure>
        <nc:MeasureText>180</nc:MeasureText>
        <m:AngleUnitText>deg</m:AngleUnitText>
      </mda:PositionHeadingMeasure>
      <mda:PositionNavigationStatus>
        <nc:StatusText>Under way using engines</nc:StatusText>
      </mda:PositionNavigationStatus>
      <mda:PositionDateTime>
        <nc:DateTime>2011-11-30T00:00:00Z</nc:DateTime>
      </mda:PositionDateTime>
    </mda:Position>
  </posex:Message>
</summary>
<dc:creator>DCMP</dc:creator>
</entry>
</feed>

```

In the Atom feed resulting from the query, four elements should be noted. First, the <link> element at the top level of the feed echoes back the query URL that returned this feed. Each individual record is content in an <entry> element. The <id> element supplies the unique ID of the record on the ISI. The <link> element provides the retrieval URL for the record on the ISI. As long as that record is available on the ISI, the link will provide access to it. The <summary> element contains the summary elements of the record. **Note that the <link> elements have replaced non-XML characters with their entity representations, so that the XML content is valid.**

All the records that match the query are returned, with one <entry> element for each record, up to the result size limit. See Sections 2.6 and 2.7 for more details on the headers and large result-set transfers.

The ISI search interface also provides utility endpoints:

- <https://mise.mda.gov/services/MDAService/search/<iepdname>/documentation>: Provides the IEPD for that focus area, allowing new consumers to download the schema and other documentation for a record type representation.
- <https://mise.mda.gov/services/MDAService/search/<iepdname>/rendering>: Provides rendering stylesheets to display the NIEM-M XML as HTML or other formats. Internally, the ISI will use this endpoint when a consumer provides an HTTP-Accept header other than the standard NIEM-M XML.
- <https://mise.mda.gov/services/MDAService/specification>: Provides a link to the current version of this document, describing the ISI REST search/retrieve interface.

## 2.2. PROTOCOL, SESSIONS, AND SECURITY

All interactions with the ISI are done over Secure Socket Layer/Transport Layer Security (SSL/TLS) connections. SSL/TLS are enforced to protect information in transit and session cookies. Each client querying against the SR interface is authenticated by the ISI and is provided with a limited-duration session cookie, which must be supplied in the header of following requests. The method for authentication is discussed the *MISE Interface Security Specification*.

### 2.3. SEARCH OPERATION

URI	https://mise.mda.gov/services/MDAService/search/iepdname>?=-	Example: https://mise.mda.gov/services/MDAService/search/noa?VesselName-Enterprise
Method	GET	
Request Headers	Authorization	As described in the MISE Interface Security Specification.
Request Content Type	Empty	
Status Codes	200 (Success)	Returns the Atom feed with the results of the search.
	400 (Bad Request)	Request was not formatted correctly.
	401 (Unauthorized)	No Authorization header.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for search of this IEPD.
Response Headers	Empty	
Response Content Type	Atom feed with search results.	

Table 1 – Search Operation Detail

### 2.4. QUERY PARAMETERS

All IEPD search endpoints support a common set of query parameters, shown in Table 2.

Parameter Name	Value Type	Comment
VesselName	String	
VesselSCONUMText	String	
VesselMMSIText	String	
VesselIMONumberText	String	
VesselNationalFlag	String	
VesselClassText	String	
VesselCallSignText	String	
VesselHullNumberText	String	
VesselOwnerName	String	
VesselCategoryText	String	

InterestCategoryCode	String	
InterestLevel	String	
PortNameText	String	
InterestType	String	
InterestCategory	String	
InterestLevel	String	
InterestStart	ISO8601 DateTime	GMT
InterestEnd	ISO8601 DateTime	GMT
ArrivalPort	String	
ExpectedArrivalTime	ISO8601 DateTime	GMT
start	ISO8601 DateTime	GMT
end	ISO8601 DateTime	GMT
ulat	WGS84 upper left latitude of a bounding box	
ulng	WGS84 upper left longitude of a bounding box	
llat	WGS84 lower right latitude of a bounding box	
llng	WGS84 lower right longitude of a bounding box	
eid	String	<a href="#">The eid parameter allows a consumer to specify a specific data provider. This field must contain the entity ID of a provider system from the Trust Fabric document. This is only used in addition to other parameters, to narrow the returned results.</a>

Table 2 – Common Query Arguments

These parameters can be combined to query for any of the record types.

- <https://mise.mda.gov/services/MDAService/search/noa?VesselName=Atlantic%20Light&PortNameText=Miami>

For **start**, **end**, and other dates, the representation is a GMT ISO8601 datetime indicating a time period within which to return new or updated records. When either **start** or **end** date is specified, the time period is unbounded, but still a valid query. For instance, if only **start** is specified, there is no **end** time on the query:

- <https://mise.mda.gov/services/MDAService/search/pos?VesselName=Atlantic%20Light&start=2012-02-29T00:00:00Z>

Refer to the *MISE Implementation Guide* for examples, queries, and use cases.

## 2.5. SCOPE

For specific events and situations, the ISI provides an additional set of query parameters that can be used to query for messages within a specific scope. When applied by an information publisher, the scope parameters modify the entitlements for a record for the duration of the scope.

Parameter Name	Value Type	Comment
Scope	String	

For example, consider a Search query for all notice of arrival messages inbound to the Port of Miami during Hurricane Katrina:

- <https://mise.mda.gov/services/MDAService/search/noa?PortNameText=Miami&scope=HurricaneKatrina>

Valid scope values can be found on the National MDA Architecture website.

## 2.6. HEADER INFORMATION

The ISI SR interface uses the following HTTP headers for specific purposes. All other HTTP headers should be interpreted according to the HTTP 1.1 standard.

- **Accept:** By default, if no Accept is specified, the summary feed will be returned in the Atom format as specified above. However, the following Accept headers may be supplied, resulting in translated feeds:
- `application/kml` – returns the Atom feed translated into a KML overlay.
- **Transfer-Encoding:** For large requests, the server may respond using HTTP 1.1 chunked transfer. When this happens, the Transfer-Encoding HTTP response header is set in place of the Content-Length header, which the protocol would otherwise require. Because the Content-Length header is not used, the ISI does not need to know the length of the content before it starts transmitting a response to the client. The ISI can begin transmitting responses with dynamically generated content before knowing the total size of that content. The size of each chunk is sent right before the chunk itself so that a

client can tell when it has finished receiving data for that chunk. The data transfer is terminated by a final chunk of length zero.

## 2.7. ERROR CODES

The ISI SR interface uses the following HTTP response codes for specific purposes. All other response codes should be interpreted according to the HTTP 1.1 standard.

- 413: If the resulting record set matching a query is too large for efficient transfer, even via HTTP chunked transfer, the ISI returns the 413 error code, indicating that the response set is too large. In this situation, the client should retry the query with a more restrictive set of parameters.

# 3. Focus-Area Specific REST Parameters

This section describes the more specific parameters to query information from each focus area. The URIs below follow the same format as the general query interface, with additional parameters described below to provide specific information.

## 3.1. NOTICE OF ARRIVAL

Parameter Name	Value Type	Comment
NoticeType	String	
NoticeTransactionType	String	
CDCCargoDeclared	Boolean	T or F

## 3.2. VESSEL POSITION

Parameter Name	Value Type	Comment
num	Integer	The num parameter is applied to queries for Position information. It specifies how many previous positions should be returned in reverse date order. The default number is 5.

## 3.3. INDICATORS AND NOTIFICATIONS

Parameter Name	Value Type	Comment
ActivityName	String	

ActivityStart	ISO8601 DateTime	GMT
ActivityEnd	ISO8601 DateTime	GMT

### 3.4. LEVELS OF AWARENESS

Parameter Name	Value Type	Comment
LevelOfAwarenessCode	Integer	This parameter is applied to queries specifically for the Levels of Awareness IEPD type. This specifies the LOA level, 1-4 that should be returned. If not specified, all matching LOA messages will be returned.

## 4. Retrieve Interface

In each Atom feed returned, the <link> element for each entry points to the canonical URI that represents that record. As long as the ISI can reference that record, the ID-specific URI will point to the same record. If the record is no longer accessible, the ISI returns an HTTP 404 status code, indicating that that record is no longer available. Each record URI takes the form <https://mise.mda.gov/services/MDAService/retrieve/<type>?entityid=<eid>&recordid=<id>>. The ID is the same as the <id> element in the <entry>. The <eid> references the data provider system. This value is the Entity ID from the Trust Fabric document. The <type> variable is the focus area, such as *pos* for position reports. Note that the <eid> parameter value URL-encoded. For example, the entity ID <http://agencyone.gov> would be represented in the URL as `https%3A%2F%2Fmise.agencyone.gov`.

### 4.1. RETRIEVE OPERATION

URI	<a href="https://mise.mda.gov/services/MDAService/retrieve/&lt;type&gt;?entityid=&lt;eid&gt;&amp;recordid=&lt;id&gt;">https://mise.mda.gov/services/MDAService/retrieve/&lt;type&gt;?entityid=&lt;eid&gt;&amp;recordid=&lt;id&gt;</a>	Example: <a href="https://mise.mda.gov/services/MDAService/retrieve/track?entityid=agencyone.gov&amp;recordid=8397109112108101736578">https://mise.mda.gov/services/MDAService/retrieve/track?entityid=agencyone.gov&amp;recordid=8397109112108101736578</a>
Method	GET	
Request Headers	Authorization	As described in the <i>MISE Interface Security Specification</i> .
Request Content Type	Empty	
Status Codes	200 (Success)	Returns the NIEM-M formatted record.

Response Content Type	400 (Bad Request)	Request was not formatted correctly.
	401 (Unauthorized)	No Authorization header.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for search of this IEPD.
	NIEM-M Record	

Table 3 – Retrieve Operations Detail

## 4.2. SCOPE

For specific events and situations, the ISI provides an additional set of query parameters that can be used to query for messages within a specific scope. When applied by an information publisher, the scope parameters modify the entitlements for a record for the duration of the scope.

Parameter Name	Value Type	Comment
Scope	String	

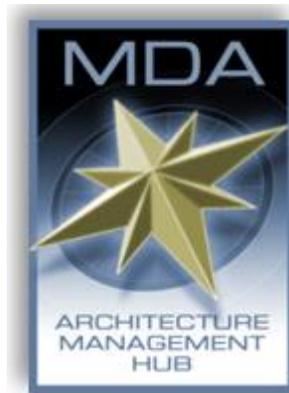
Scope must be applied to the retrieve query, similar to the search query:

- <https://mise.mda.gov/services/MDAService/retrieve/track?entityid=agencyone.gov&recordid=8397109112108&scope=HurricaneKatrina>

# APPENDIX F - GOVERNANCE MANUAL

---

---



## Maritime Information Sharing Environment (MISE)

Governance Manual  
Version 1.0

# 1. Introduction

## 1.1. BACKGROUND

The National Maritime Domain Awareness (MDA) Concept of Operations describes an end-state in which any authorized user is able to access the information they require through seamless validation of the needs, rights and authorities of the user balanced against the permissions or restrictions established by organizations publishing the information. The Maritime Information Sharing Environment (MISE) provides that environment as a national capability to share and search maritime information across organizational boundaries relevant to the maritime domain.

### Value to the Maritime Community:

- **User Convenience** – Users can access multiple services using a common set of standardized security credentials making it easier to sign on, access applications, and manage account information.
- **Interoperability** – MISE specifies common security standards and framework so applications can adopt interoperable security specifications for authentication and authorization.
- **Cost-Effectiveness** – MISE facilitates information sharing by using standardized Extensible Markup Language (XML)-based credentials that include information about each user's identity and privileges. This reduces the cost and complexity of identity administration required to access applications and vet users.
- **Privacy** – MISE will reduce the propagation of personally identifiable information (PII), reduce redundant capture and storage of PII, and depersonalize data exchanges across domains using privacy metadata.
- **Security** – the MISE model will improve the security of PII and data in participant organizations' applications by providing a standardized approach to online identities between agencies or applications.

### Contents

The MISE Governance Manual defines the governance structure for MISE, including the parties that play a role in the governance structure (e.g. Board of Directors, MISE Management, information provider representatives, and information consumer representatives) and the decisions to be made by each party.

### Target Audiences

The target audience for this document includes managers and technical representatives of prospective MISE participant organizations who are planning to implement a service within the MISE. It also includes vendors, contractors, and consultants who are required to establish technical interoperability with National Information Exchange Model - Maritime (NIEM-M) standards as part of a contracted project or product implementation.

## 2. MISE Governance Structure

The MISE governance structure will consist of the MISE Board of Directors (BOD), MISE Management, and representatives (Information Providers and Information Consumers) from various organizations participating in the MISE Configuration Control Board (CCB). The governance structure is captured in the figure below.

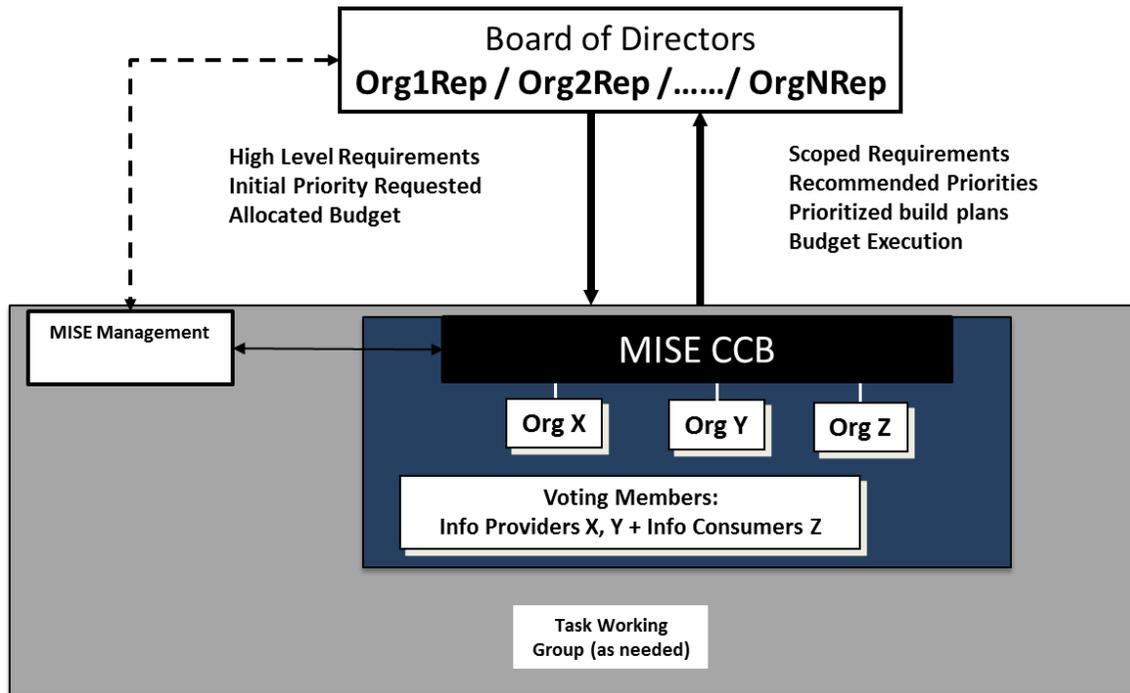


Figure 3 MISE Governance Structure

The specific organizations and individuals that comprise the BOD, MISE Management and CCB will evolve over time, so are not specified in this document.

## 3. Board of Directors

The MISE BOD is the executive-level body with representation from primary stakeholders that guides the MISE and is the final authoritative body to make decisions for the environment.

### 3.1. RESPONSIBILITIES

The BOD provides executive oversight of the MISE and decides on matters that are beyond the authority delegated to the MISE CCB. This includes, but is not limited to:

- Provide high level requirements
- Allocate the MISE budget
- Provide prioritization and approval for execution of CCB recommendations
- Approval authority for any modification to Trusted System Agreements

- Approval authority for any changes to any guidelines, standards, or documents of the MISE
- Approval authority for any changes to the governance structure
- Approval authority for membership policy, membership requests, membership suspension or revocation, and any changes to the MISE systems and security policies
- Identify potential new partners to join the environment

### 3.2. MEETINGS

The BOD shall meet quarterly to discuss general business and other matters that may arise. On an as needed basis, a BOD member may call an emergency meeting to discuss and decide matters that need immediate attention. To call an unscheduled meeting, the board member requesting the meeting must notify the other board members via written or electronic communication and obtain concurrence of no less than 50 percent of the Board to meet.

Regularly scheduled and emergency meetings can occur in person or via teleconference, and require a quorum of the Board of Directors to proceed. If a quorum is not present, the meeting shall be rescheduled to the next date when a quorum can be present. Emergency meetings will have no effect on the next regularly scheduled meeting date.

## 4. MISE Configuration Control Board

The CCB has responsibility for reviewing change requests for MISE exchange standards, services and processes, and forwarding recommendations to the BOD for approval decisions. The CCB is composed of representatives from information provider organizations, information consumer organizations, and MISE management.

### 4.1. RESPONSIBILITIES

The role of the CCB is to provide a common forum where information providers, information consumers and MISE management can jointly recommend MISE changes and enhancements, and forward those recommendations to the BOD. Specific responsibilities include:

- Review Change Requests (CR)
- Identify focus areas for new Information Exchange Package Documentation (IEPD)
- Provide recommendations to improve the execution of MISE
- Analyze recommendations for changes to any guidelines, standards, or documents of the MISE
- Analyze recommendations of any changes to the governance structure
- Analyze recommendations for membership policy, membership suspension and revocation, and any changes to the MISE systems and security policies
- Approve the formation of Task Working Groups (TWG) to target specific tasks that must be completed

- Approve the dissolution of TWG when tasks are completed
- Implement the MISE Work Flow process as shown in the diagram below.

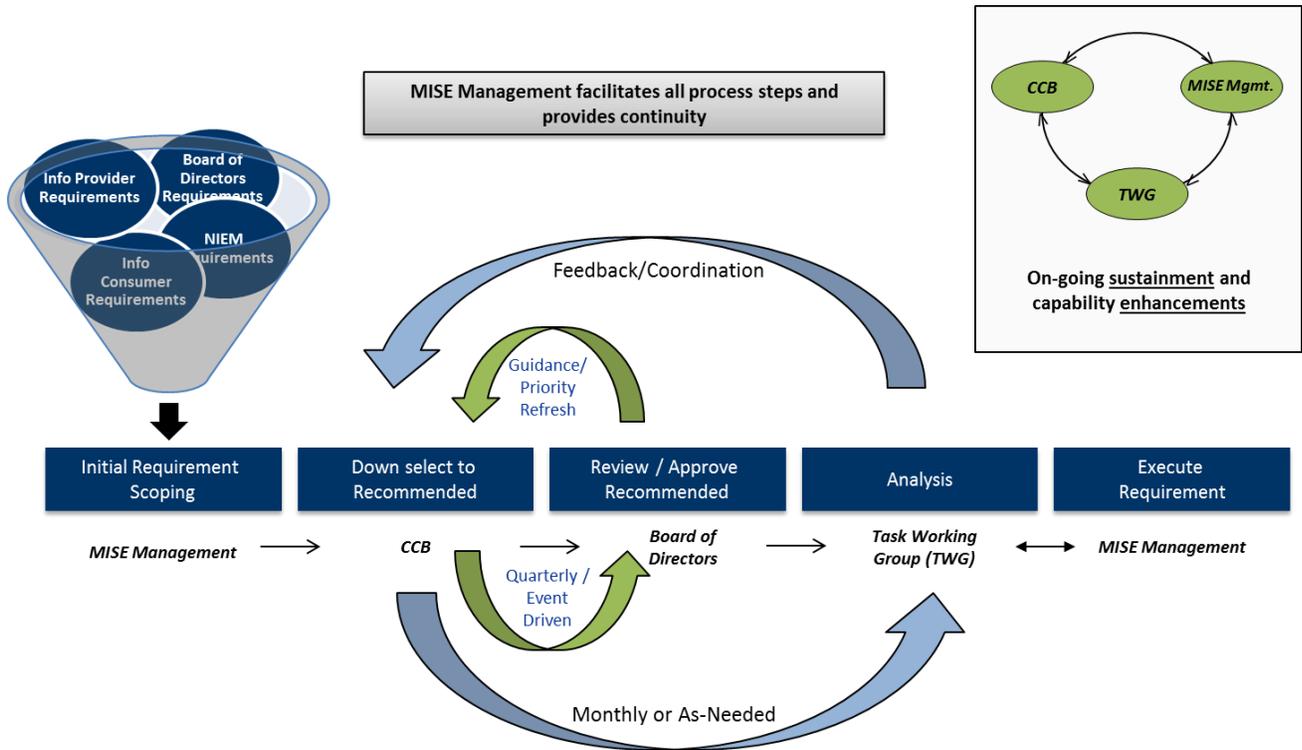


Figure 4 MISE Work Flow

## 4.2. MEETINGS

The CCB shall meet monthly to discuss Change Requests, provide or analyze recommendations, and other matters that may arise. On an as needed basis, the CCB may call an emergency meeting to discuss and decide matters that need immediate attention. Any unscheduled meeting must be approved by MISE management.

Regularly scheduled and emergency meetings can occur in person or via teleconference. If a quorum is not present, the meeting shall be rescheduled to the next date when a quorum can be present. Emergency meetings shall have no effect on the next regularly scheduled meeting date.

## 5. MISE Management

MISE Management will facilitate the identification, collaboration, management, movement, and processing of maritime information by leveraging the MISE. The following are the objectives of MISE Management:

1. Lead the NIEM-M domain to standardize sharing of common maritime information with interagency and international partners

2. Provide standards-based environment to facilitate secure, seamless access to unclassified maritime information among trusted systems
3. Manage day-to-day operations of the MISE

## 5.1. RESPONSIBILITIES

The MISE Management will be responsible for the day-to-day operations of the MISE. This includes, but is not limited to, the following:

- Facilitate CCB Meetings
- Execute approved CCB requirements
- Maintain NIEM-M domain model and exchange standards
- Manage MISE Help Desk
- Assist information providers in developing and maintaining Information Access Policies (IAPs)
- Certify new Trusted Systems within the MISE
- Maintain and disseminate the Trust Fabric

## 6. Information Providers/Consumer Representatives

Information Providers / Information Consumers Representatives participate in the MISE with an approved trusted system adhering to the unclassified maritime information sharing environment rules established by MISE management.

## 7. Responsibilities of the MISE Governing Bodies

The responsibilities of the MISE are listed below in this section.

### 7.1. POLICY

A membership policy for the MISE will be established by the MISE Management and approved by the BOD. Any subsequent changes to the policy will require BOD approval.

### 7.2. APPROVAL

The BOD has the authority to approve new members; MISE Management has the authority to sign any service Level Agreements on behalf of the MISE. Any modifications to the standard agreement or assignments of these agreements will require the BOD approval.

### 7.3. MEMBERSHIP SUSPENSION

If suspicion exists that a member has violated any of the MISE provisions, standards, policies, or procedures that threatens the integrity of the MISE, then MISE Management may suspend membership for up to 15 days. A longer suspension may be imposed by the BOD.

### 7.4. AUDIT / INVESTIGATE

The MISE has the right to audit the MISE-related activities of any member across the MISE. The use of an approved network banner is required in order to inform users that their computer activity on any government network is subject to monitoring. The model computer banner language, quoted below, may be considered for use:

*“You are accessing a U.S. Government Information system, which includes (1) this computer, (2) this computer network, (3) all computers connected to this network, and (4) all devices and storage media attached to this network or a computer on this network. This information system is provided for U.S. Government-authorized use only. Unauthorized or improper use of this system may result in disciplinary action, as well as civil and criminal penalties.*

*By using this information system, you understand and consent to the following:*

- *You have no reasonable expectation of privacy regarding communications or data transiting or stored, or activity conducted on this information system.*
- *At any time, and for any lawful government purpose, the Government may monitor, intercept, and search any communication or data transiting or stored on this information system.*
- *Any communications or data transiting or stored, or activity conducted on this information system may be disclosed or used for any official government purpose.”*

In the case of an alleged breach of MISE policy, the MISE Management will perform a basic audit and investigation. If there is initial validation of the concern based on this initial investigation, the member can be required to provide a third-party audit to show that their procedures adhere to MISE rules.

### 7.5. MEMBERSHIP REVOCATION

The decision to revoke any member’s membership without cause will be in the sole discretion of the BOD and will require 60 days’ notice to such member. If any member decides to cancel their membership, they may do so upon 60 days’ notice to the MISE Management. The MISE Management will be responsible for periodically informing the BOD of the cancellation of any memberships.

## 8. Conflict Resolution

The MISE members agree that any dispute or conflict will be brought to the BOD, via MISE Management, for resolution. All decisions made by the BOD in resolution of a conflict will be deemed final upon notice to the parties involved. Unless otherwise

stipulated in the resolution description, the decision will be implemented in 30 days from initial notice.

### 8.1. DISPUTES AMONG MISE MEMBERS

Disputes among MISE members shall be resolved via the following process. When a dispute occurs among members, one or more members may notify the MISE Management of the dispute in writing. Upon receiving written notification of the dispute, the MISE Management may, at its discretion, take the necessary steps to investigate the dispute. After investigating the dispute, the MISE Management may either render a decision to resolve the dispute or submit the issue to the BOD for a vote. The MISE Management must complete this process no later than 60 days after receiving written notification of the dispute.

### 8.2. DISPUTES BETWEEN MEMBERS AND THE MISE MANAGEMENT

When a dispute occurs between any of the MISE members and the MISE Management, one or more parties of the dispute may notify the BOD of the dispute in writing. Upon receiving written notification of the dispute, the Director in receipt of the written notification must forward the notification to all members of the BOD. The BOD may, at its discretion, take the necessary steps to investigate and render a decision on the dispute. The MISE Board of Directors must complete this process no later than 60 days after receiving written notification of the dispute.

### 8.3. END USER CONFLICT

Any end-user conflict will be addressed solely by the Information Provider / Information Consumer Representative to which the user subscribes, and cannot be appealed to any other MISE member.

## 9. Core MISE Governance Documents

The operation of the MISE is governed by this MISE Governance Manual. This document provides the foundation for governing and operating the MISE.

Further details regarding the required processes to manage the MISE environment are described in the Appendices.

## 10. Glossary

Attributes	Characteristics of a persona that defines a user in a particular role (sent by a trusted system to the ISI)
Board of Directors	The Board of Directors decides on any matters that fall outside of the role of the MISE Management and provide general guidance. Some of the responsibilities include approval of any modification to standard agreements, documents, or governance structure.
MISE Management	The MISE Management is responsible for the day-to-day operations of the MISE.
Governance	Establishment of policies and continuous monitoring of the proper implementation by the members of the governing body of an organization
Information Consumer	A type of trusted system that authenticates users, using an internal or external identity provider, and passes information access requests and user attributes on behalf of the user to the ISI
Information Provider	A type of trusted system that provides both maritime information and corresponding Information Access Policy (IAP) to the ISI
Information Sharing Infrastructure (ISI)	Handles requests from the trusted systems on behalf of the users. Operating on the user attributes, it will make entitlement decisions, processing messages from the information providers based on their IAP and providing responses to the trusted system
Personally Identifiable Information (PII)	Information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify a single individual
Service Provider	An entity that provides services to other entities
Trusted System	Information provider and/or information consumer. It also passes information and IAP or information access requests and user attributes to the ISI. The trusted system relies on an internal or external identity provider to authenticate users.
User	Any person, organization, system, etc. that authenticates to the trusted system via an internal or external identity provider in order to access the ISI.

## 11. Acronyms

BOD	Board of Directors
CCB	Configuration Control Board
CR	Change Request
IAP	Information Access Policy
IEPD	Information Exchange Package Document
ISI	Information Sharing Infrastructure
MISE	Maritime Information Sharing Environment
MDA	Maritime Domain Awareness
NIEM-M	National Information Exchange Model – Maritime
PII	Personally Identifiable Information
TS	Trusted System
TWG	Task Working Group
XML	Extensible Markup Language

## 12. Request to Join Process

The Request to Join Process serves as a preliminary qualifications assessment for potential MISE Trusted Systems to join the environment. It provides an opportunity for potential MISE Stakeholders to learn about the environment and the BOD to determine applicability of the system for the environment. Figure C1 depicts the Request to Join Process. Table C1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Work with maritime community to identify new MISE Stakeholders	BOD	Discussions	Discussions displaying interest in MISE	At any time.
2	Review unsolicited requests by agencies to join MISE	MISE Management	Unsolicited request/interest	Discussions displaying interest in MISE	At any time.
2	Receive informational brief about MISE	MISE Stakeholder	MISE introduction brief	Presented MISE introduction brief	Within 20 days of initial interest
3	Notify Management of interested Stakeholder	BOD	Completed presentation of MISE introduction brief	Notified MISE Management of potential stakeholder	Within 5 days of giving MISE introduction brief
4	Schedule meeting with interested MISE stakeholder	MISE Management	Notification from MISE management of potential stakeholder	Scheduled meeting with potential stakeholder	Within 10 days of notification
5	Technical evaluation of readiness of system	MISE Stakeholder	Information provider survey	Completed information provider survey	Occurs during meeting previously scheduled
6	Compile recommendation report	MISE Management	Completed information provider survey	Completed recommendation report	Within 5 days of meeting with potential stakeholder
7	Review recommendation report	BOD	Completed recommendation	Analysis and Authorization Decision	Meeting of BOD (Quarterly or as needed)
8	Notified of rejection	MISE Stakeholder	BOD Decision	Notification to stakeholder of decision	Within 5 days of BOD Decision
9	Notified of approval	MISE Stakeholder	BOD decision	Notification to stakeholder of decision	Within 5 days of BOD Decision

Table C1. Request to Join Process

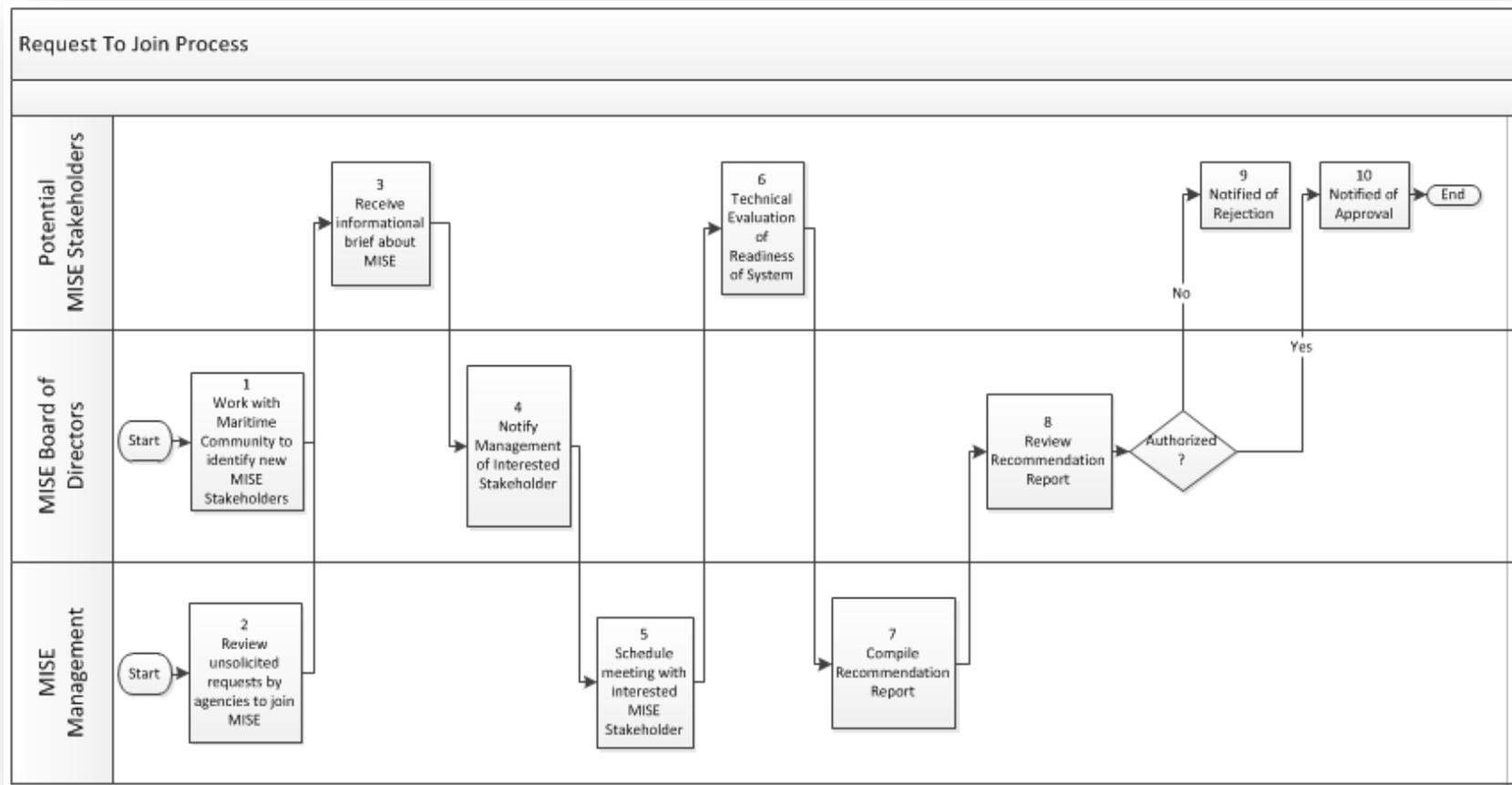


Figure C1: Request to Join Process

### 13. MISE Help Desk Process

The purpose of the Help Desk is to solve operational problems raised by end-users. As the MISE is comprised of many member organizations, each with its own local help desk resources, stakeholders must leverage these local resources to the maximum extent possible. The primary guiding principle in the design of the MISE help desk structure is that all problems SHOULD be solved as close to the user as possible, and with as little centralized effort as possible. Figure D1 depicts the Help Desk Process. Table D1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Experience IT Problem	MISE Trusted System User	Experienced IT problem	Experienced IT problem	At any time
2	Work with Local network Help Desk to Resolve Issue	User’s Network Help Desk	Experienced IT problem	Unresolved IT problem	At any time
3	Work with Trusted System Help Desk to resolve issue	MISE Trusted System	Unresolved It problem	Unresolved IT problem	At Any time
4	Submit issue through mda.gov	MISE Management	Unresolved IT problem	Resolved IT problem	Within 10 business days of notification
5	Work with Trusted System Help Desk to Resolve Issue.	MISE Management	Unresolved IT problem	Resolved IT problem	Within 10 business days of notification

Table D1. MISE Help Desk Process

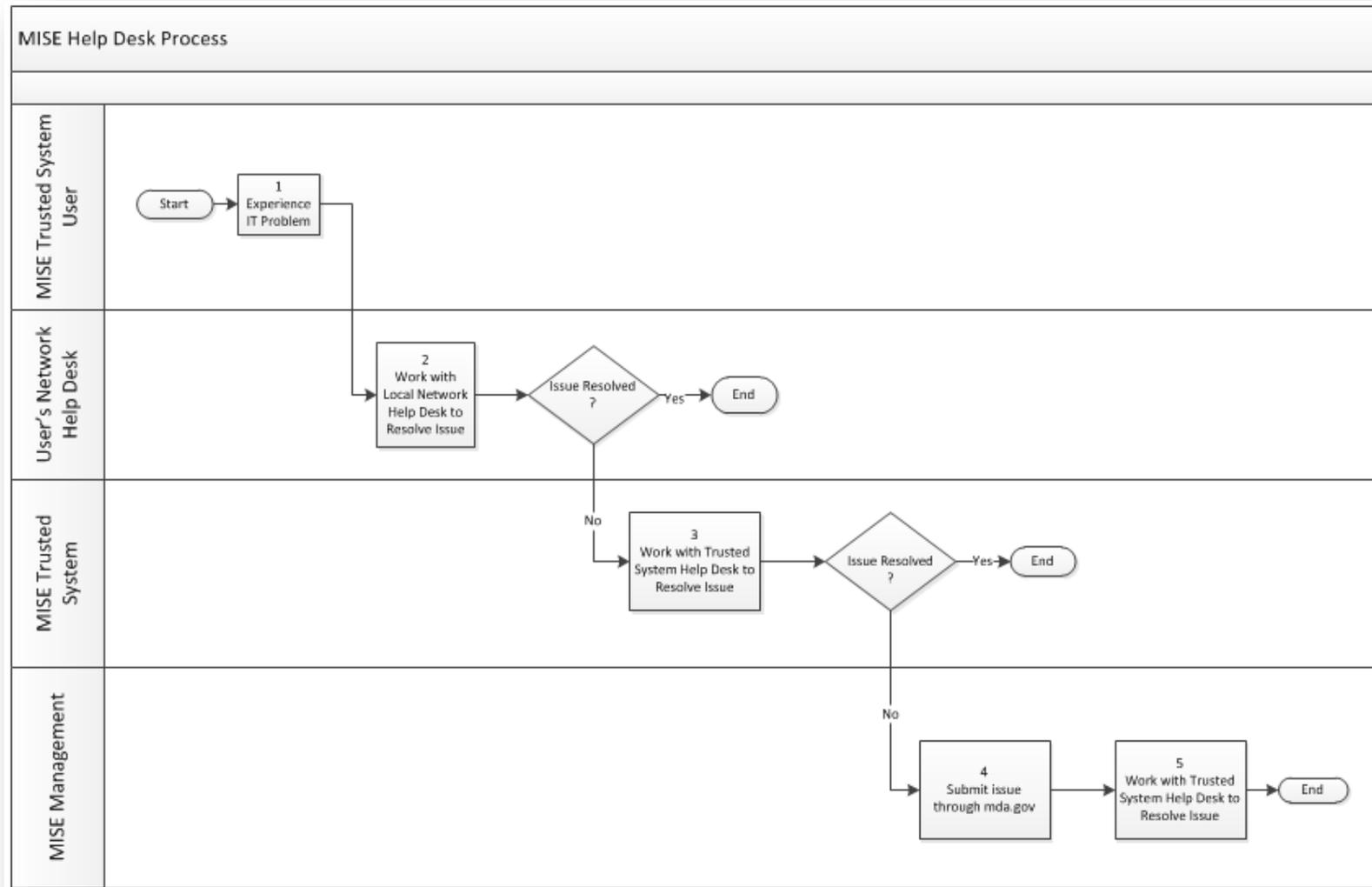


Figure D1. MISE Help Desk Process

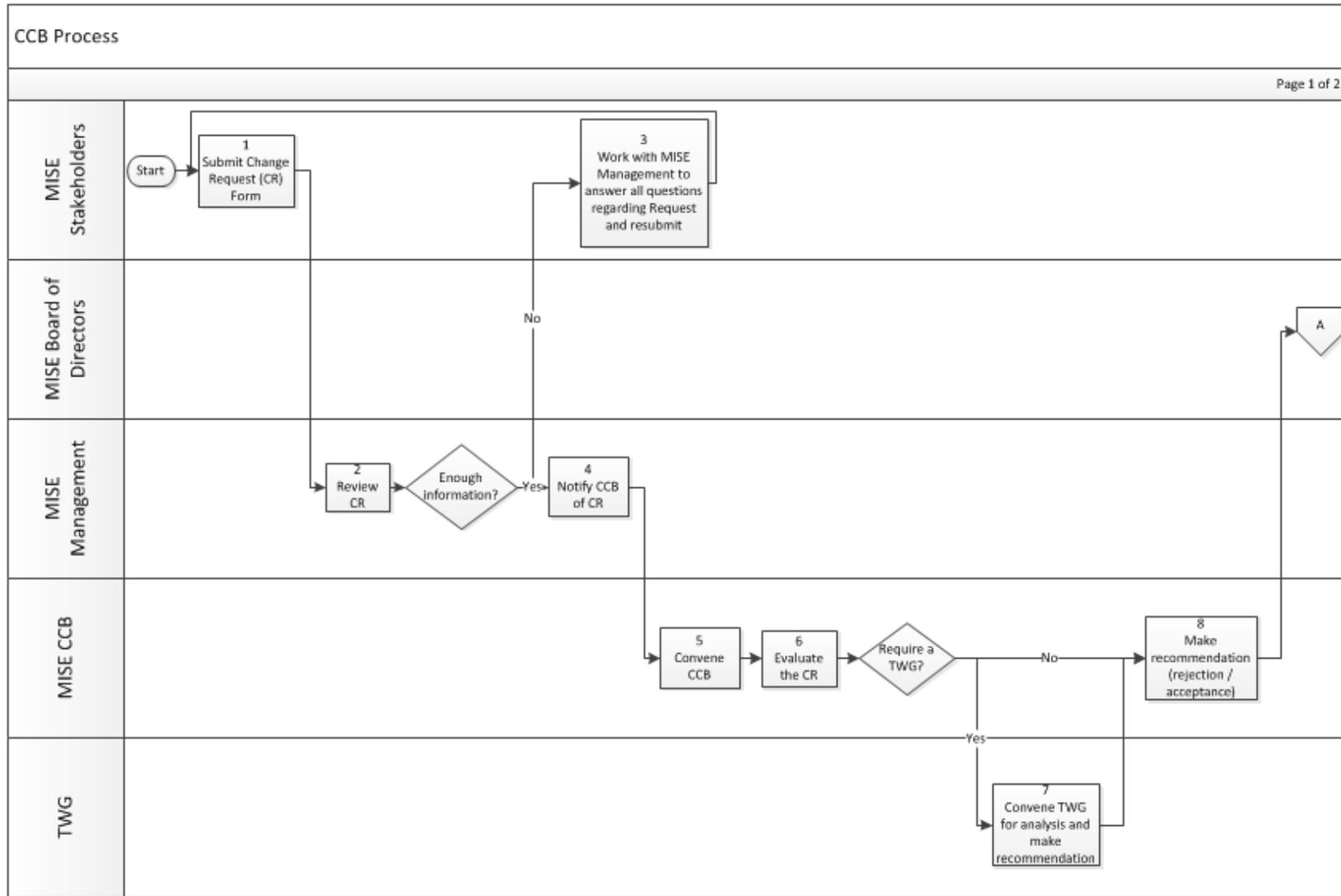
# 14. Configuration Control Board Process

The MISE CCB is responsible for controlling the MISE baseline, and evaluating and approving proposed changes. Figure E1 depicts the CCB Process. Table E1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Submit Change Request (CR) Form	MISE Stakeholder	CR Form Template	CR Form	At any time.
2	Review CR	MISE Management	CR Form	Completed CR Form	Within 10 working days of receipt of CR
3	Work with MISE Management to answer all questions regarding request and resubmit	MISE Management / MISE Stakeholder	Incomplete CR Form	Completed CR Form	Within 5 working days after review of CR
4	Notify CCB of CR	MISE Management	Completed CR Form	Confirmed notification to CCB	Within 15 days of receipt of CR
5	Convene the CCB	CCB	Completed CR Form	Scheduled CCB Meeting	At least quarterly or as needed
6	Evaluate the CR	CCB	Scheduled CCB Meeting Completed CR Form	CCB Meeting Minutes capturing discussion	Within 5 days of CCB Meeting
7	Convene TWG for analysis and make recommendation	TWG	Direction from CCB Completed CR Form	Recommendation / Rejection Report	Within 5 days of CCB Meeting
8	Make recommendation (rejection/approval)	CCB	CCB Meeting Minutes capturing discussion Completed CR Form TWG Recommendation/Rejection Report	CCB recommendation / rejection Cost estimate (if needed)	Quarterly

9	Review CR and recommendation	BOD	Scheduled BOD Meeting Completed CR Form CCB recommendation/rejection Cost estimate (if completed)	BOD meeting minutes capturing discussion BOD cost estimate approval (if needed) BOD authorization / rejection	Quarterly
10	Plan Update	CCB	BOD authorization BOD cost estimate approval (if needed)	Planned schedule Planned changes	Next CCB meeting
11	Close CR	CCB	BOD rejection	Closed CR Explanation of rejection	Within 5 days of decision
12	Notified of closure and explanation of rejection	MISE Stakeholder	Closed CR Explanation of rejection	MISE Stakeholder acknowledgement of rejection	Within 5 days of decision
13	Notified of approval and status update	MISE Stakeholder	Planned changes Planned schedule	MISE Stakeholder acknowledgment of approval	Within 5 days of decision
14	Make changes	MISE Management	Planned Schedule Planned Changes	Implemented changes and final Report	As needed dependent on tasking and resources available
15	Close CR	CCB	Implemented changes and final report	Closed CR	Within 5 Days of changes implemented and finalized
16	Notified of completion	Information Provider / Information Consumer	Closed CR	MISE Stakeholder acknowledgement of completion	Within 5 days of closure

Table E1. CCB Process



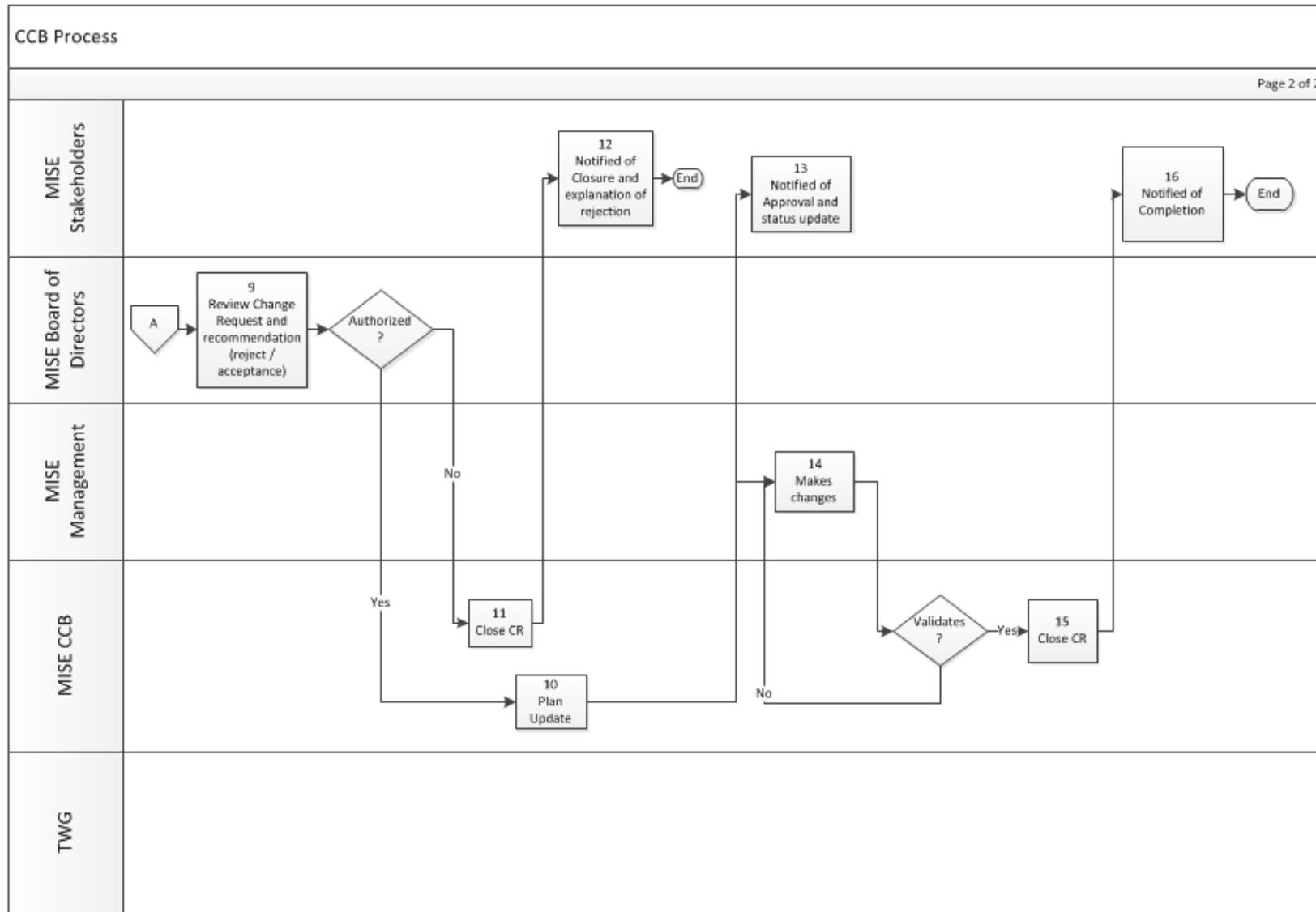


Figure E1. CCB Process

## 15. Onboarding Process

Once a MISE Stakeholder is authorized to join the environment, the Information Provider/Information Consumer will complete the implementation in accordance with the MISE Implementation Guide. The Onboarding Process will allow the Trusted System to be connected to the MISE environment. Figure F1 depicts the Onboarding Process. Table F1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Complete Implementation	MISE Trusted System	Authorization to join MISE Environment	Completed Data Mapping Model Transforms (if needed) Attribute Mapping Developed IAP Implemented Required Services (Publication, Search, Retrieve, Security)	At any time.
3	Add Trusted System to the Test Trust Fabric	MISE Management	Procured Certificate / Key	Trusted System added to the Test Trust Fabric	Within 10 days of TS certificate in key store
4	Submit IAP	MISE Management	Trusted System added to the Trust Fabric	Implemented IAP	Within 10 days of TS certificate in key store
5	Issue HTTP PUT / HTTP GET	MISE Trusted System	Implemented IAP	Issued HTTP PUT/HTTP GET	Within 10 days of TS certificate of key store
6	Test Interoperability	MISE Management / MISE Trusted System	Issued HTTP PUT/HTTP GET	Successful Interoperability Test	Within 10 days of TS certificate of key store
7	Work with MISE Management to address interoperability issues	MISE Trusted System	Unsuccessful interoperability test	Identified and corrected issues	Within 5 days of unsuccessful interoperability test
8	Add Trusted System to	MISE Management	Successful interoperability test	Trusted System add to the Operational Trust Fabric	Within 5 days of successful interoperability test

	Operational Trust Fabric				
9	Troubleshoot	MISE Management	Unsuccessful interoperability test of Operational Trust Fabric	Identified and corrected issues Successful interoperability test of Operational Test Fabric	Within 5 days of unsuccessful test
10	Notified of successful Onboarding	CCB and BOD	Successful interoperability test of Operational Test Fabric	Notification of successful Onboarding	Within 2 days of successful interoperability test

Table F1. Onboarding Process

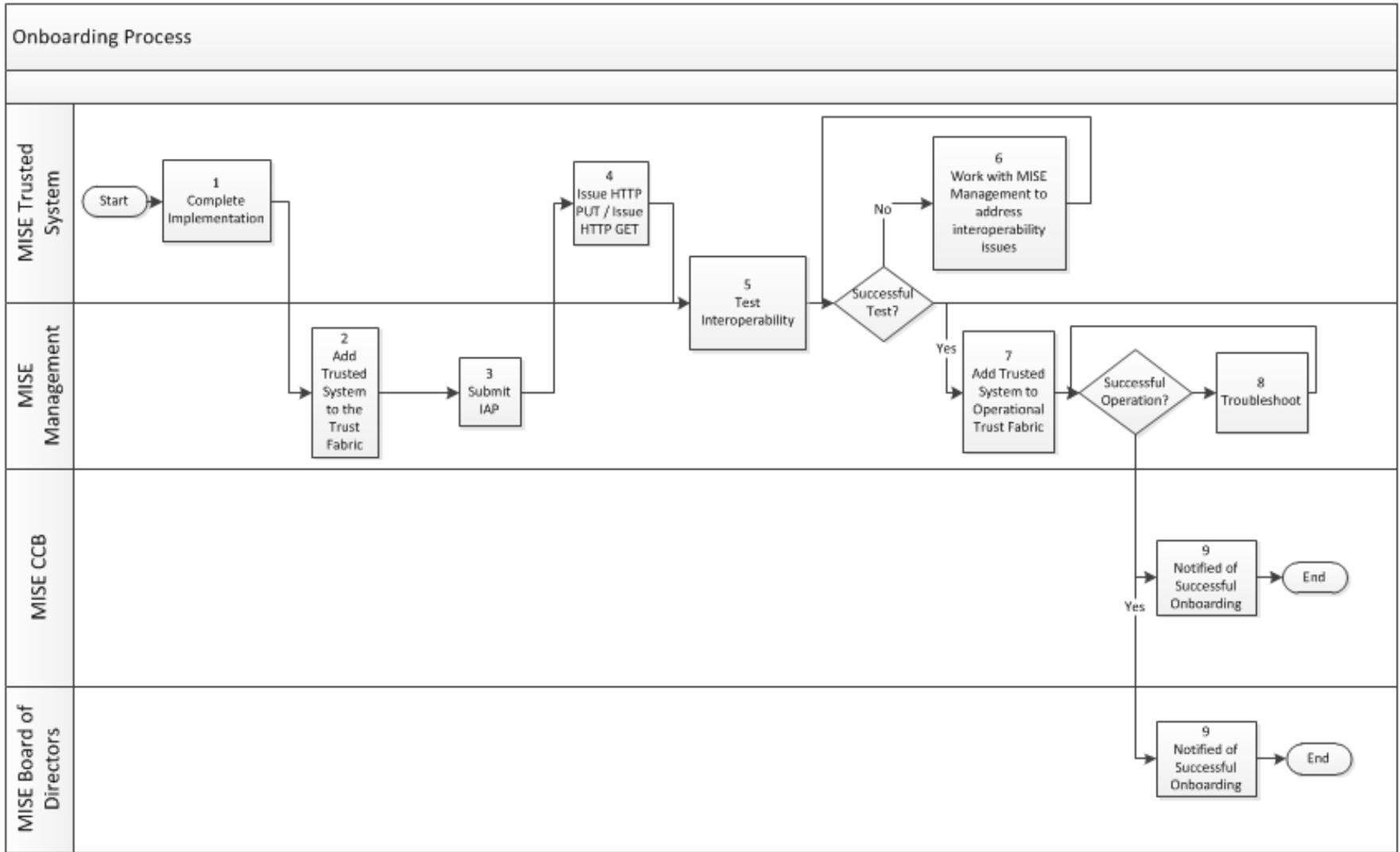


Figure F1. Onboarding Process

# APPENDIX G - NIEM-M LOGICAL MODELS

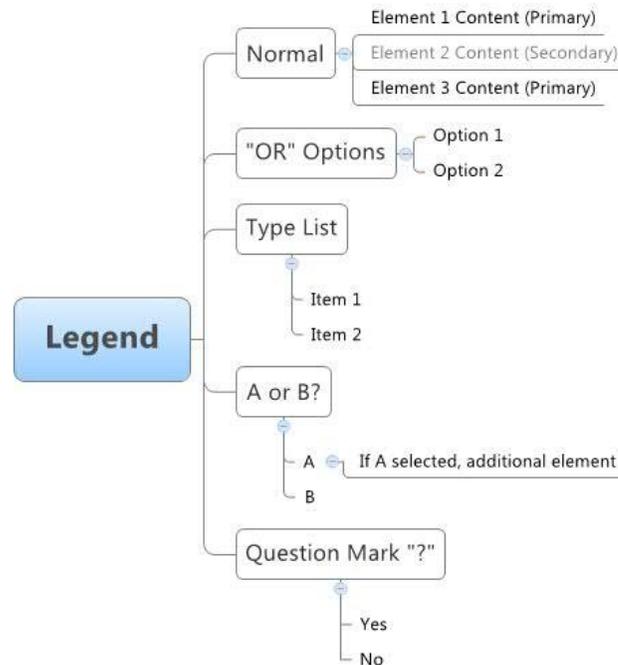
## 1. Introduction

The NIEM-M models, in XML format, can be confusion to non-technical participants. To facilities discussion and understand within the non-technical community, graphical representations of the logical models were developed. The logical models *do not* follow the physical model layout. The physical models are constrained by many factors like when specific elements were added to the model and changing NIEM Naming and Design Rules. However, the logical models do represent every part of the physical model.

Each IEPD and the NIEM-M EIEM are represented. The blocks for the EIEM are contained in one section and are not repeated in each IEPD. IEPD unique blocks are contained in corresponding IEPD section.

### 1.1. UNDERSTANDING THE LOGICAL MODELS

In order to convey the multiple data types, choices, relationships and primary or secondary elements, the logical diagrams use a variety of layout constructs. The below diagram identifies the structure and meaning of those constructs.

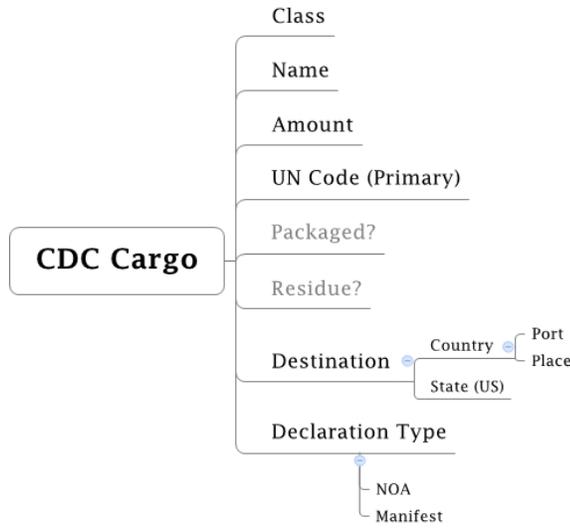


## 2. Enterprise Information Exchange Model

The NIEM-M EIEM defines the reusable objects, or blocks, used in the various IEPDs. Each EIEM block, at the time of this publishing, is represented.

### 2.1. EIEM LOGICAL OBJECTS (BLOCK LEVEL DIAGRAM)

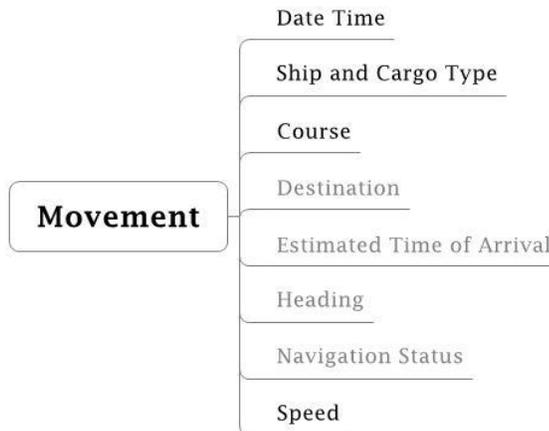
#### 2.1.1. CDC CARGO



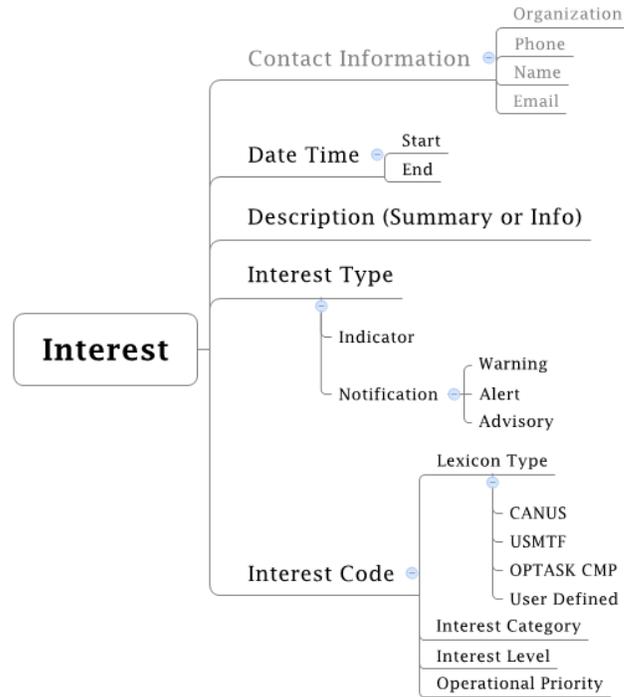
#### 2.1.2. CREW NATIONALITY COUNT



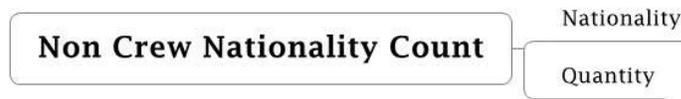
#### 2.1.3. MOVEMENT



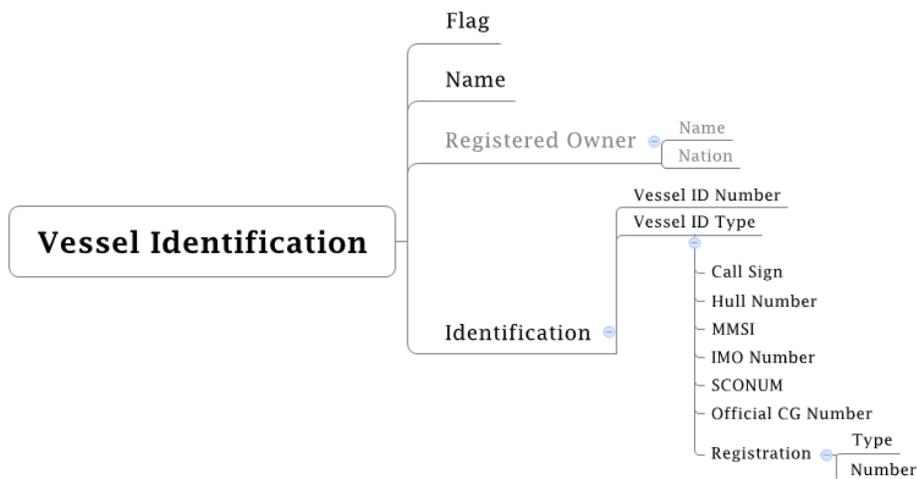
### 2.1.4. INTEREST



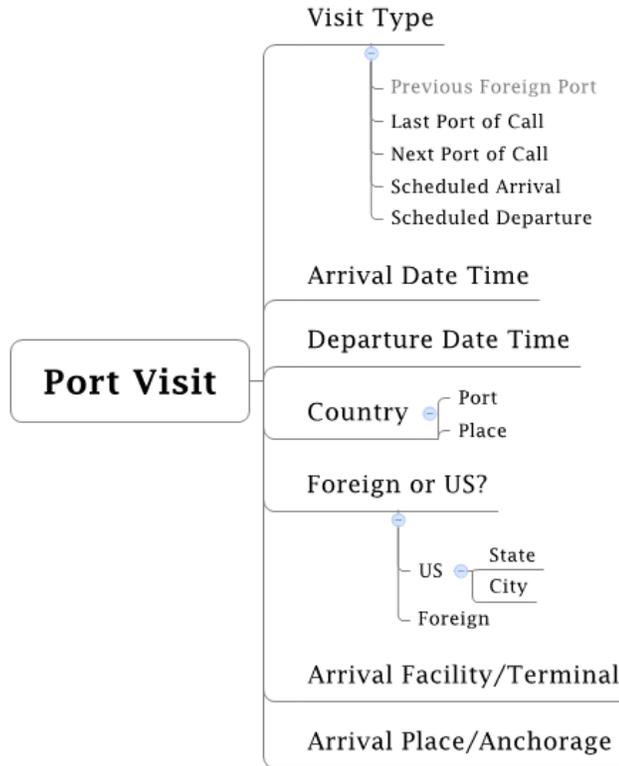
### 2.1.5. NON-CREW NATIONALITY COUNT



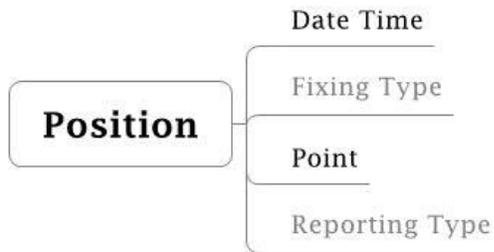
### 2.1.6. VESSEL IDENTIFICATION



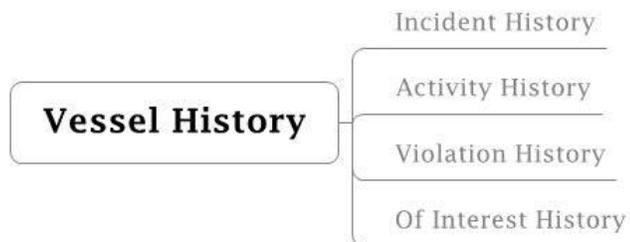
2.1.7. PORT VISIT



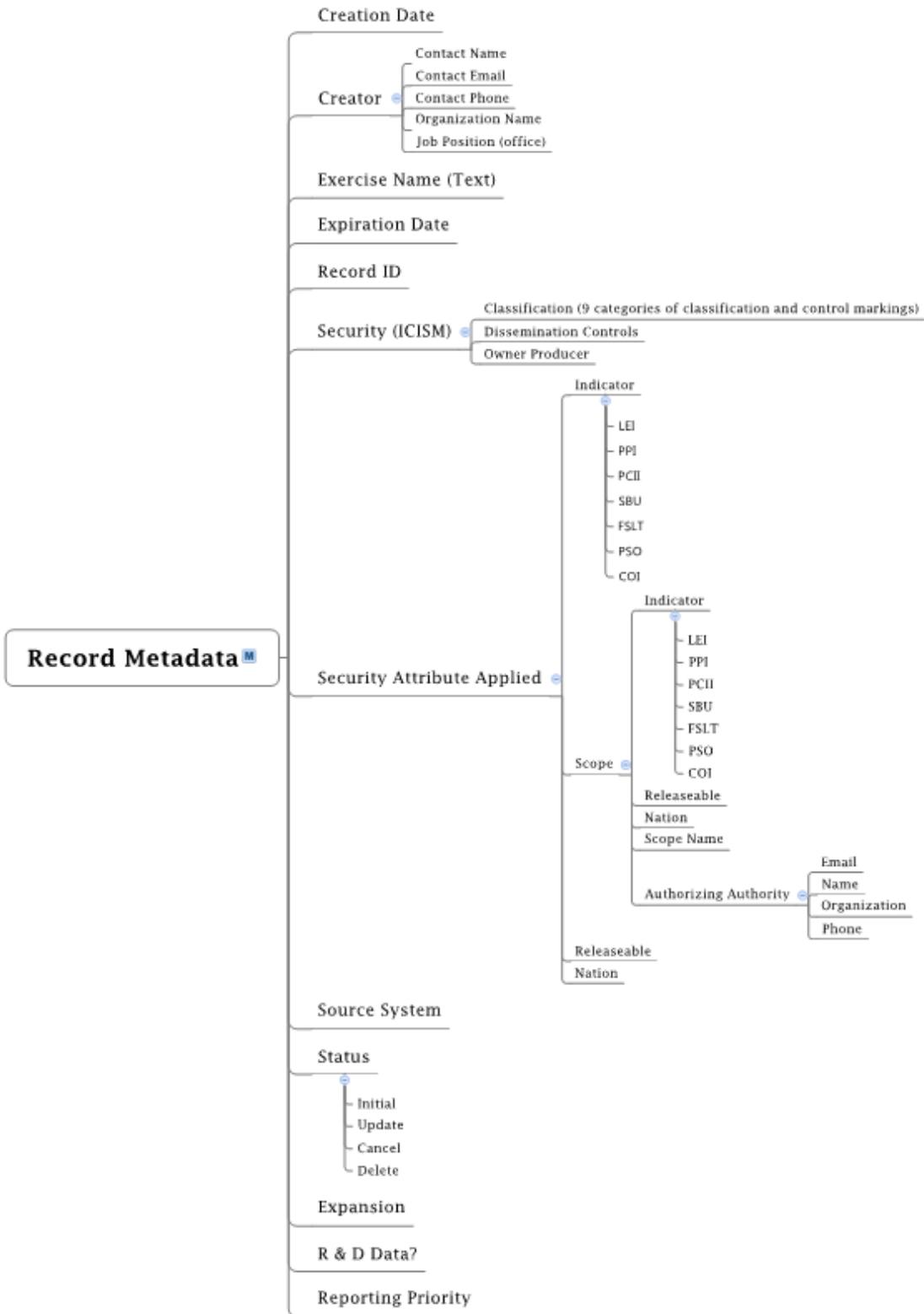
2.1.8. POSITION



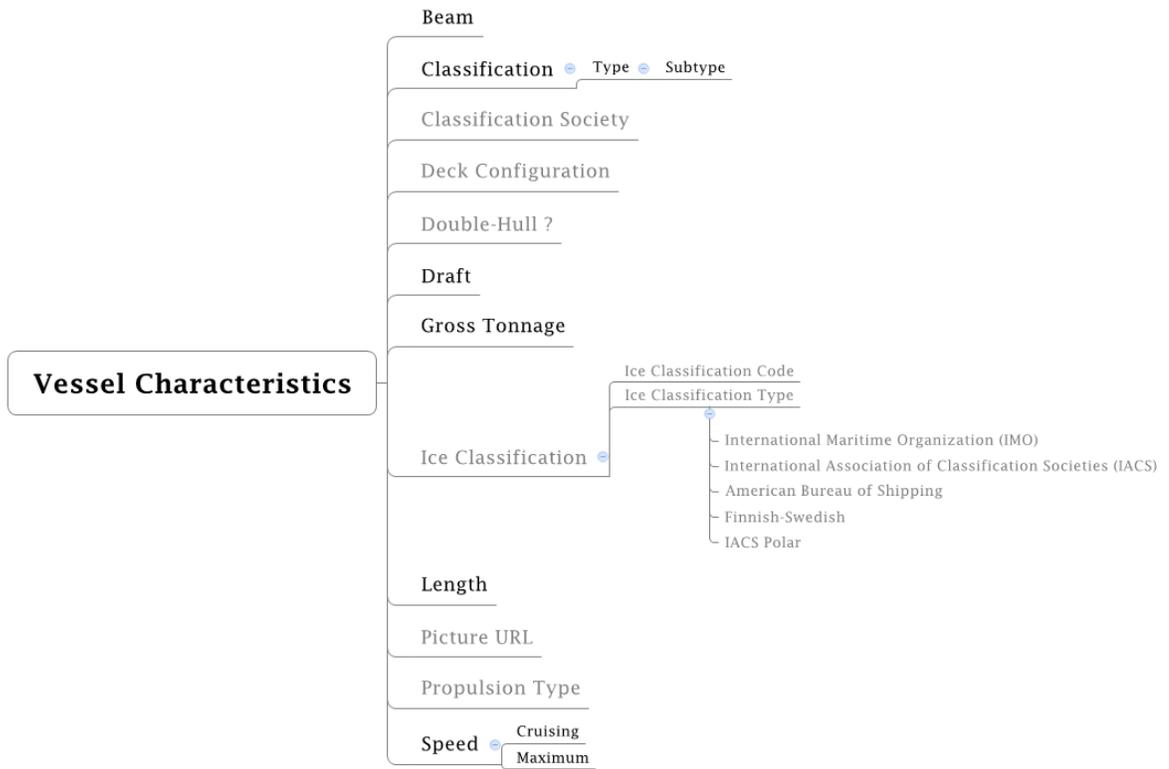
2.1.9. VESSEL HISTORY



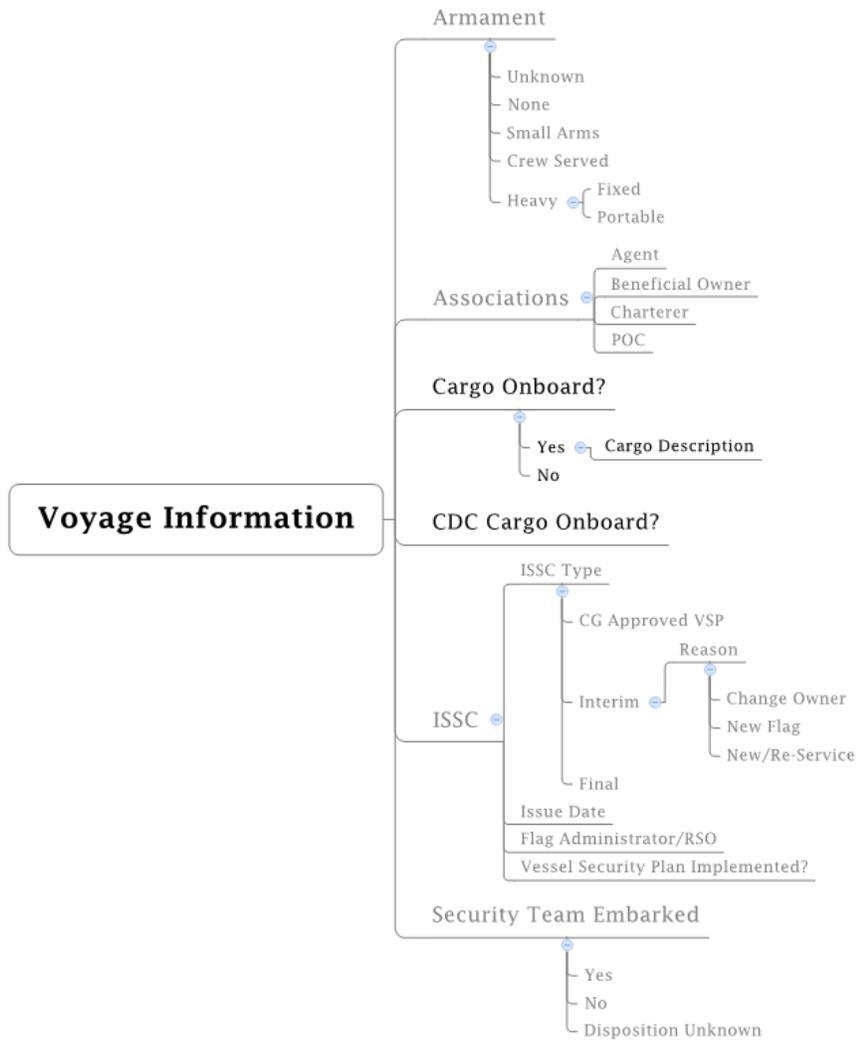
### 2.1.10. RECORD METADATA



### 2.1.II. VESSEL CHARACTERISTICS



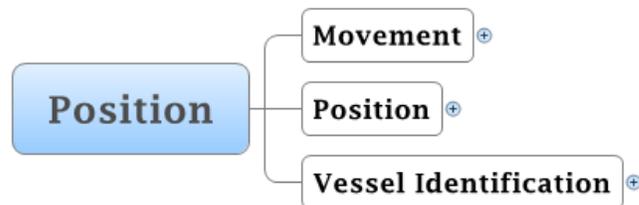
### 2.1.12. VOYAGE INFORMATION



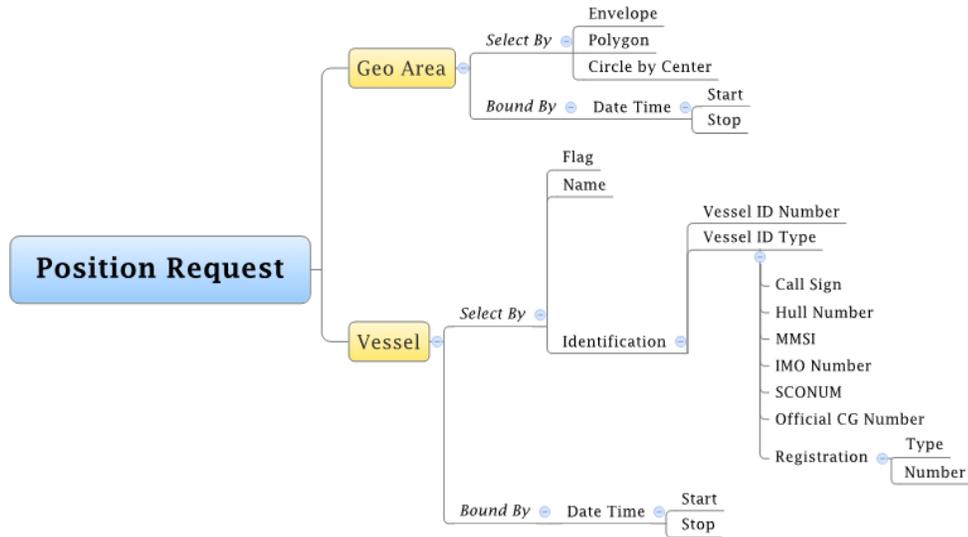
## 3. Position

Below are graphical representations of the exchange and request/query logical models for the NIEM-M Position IEPD.

### 3.1. EXCHANGE MODEL



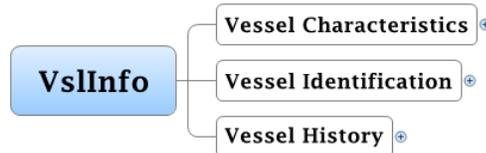
### 3.2. REQUEST/QUERY MODEL



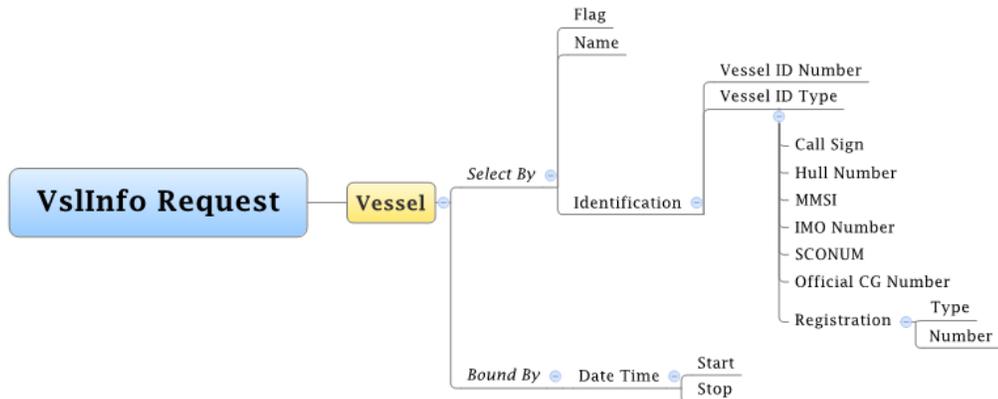
## 4. Vessel Information

Below are graphical representations of the exchange and request/query logical models for the NIEM-M Vessel Information IEPD.

### 4.1. EXCHANGE MODEL



### 4.2. REQUEST/QUERY MODEL

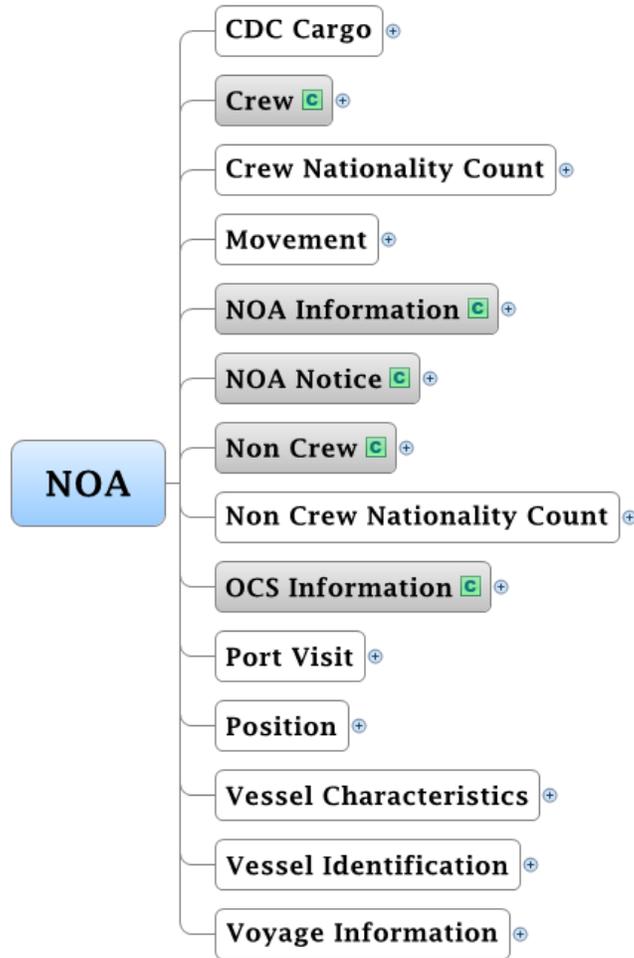


# 5. Notice of Arrival

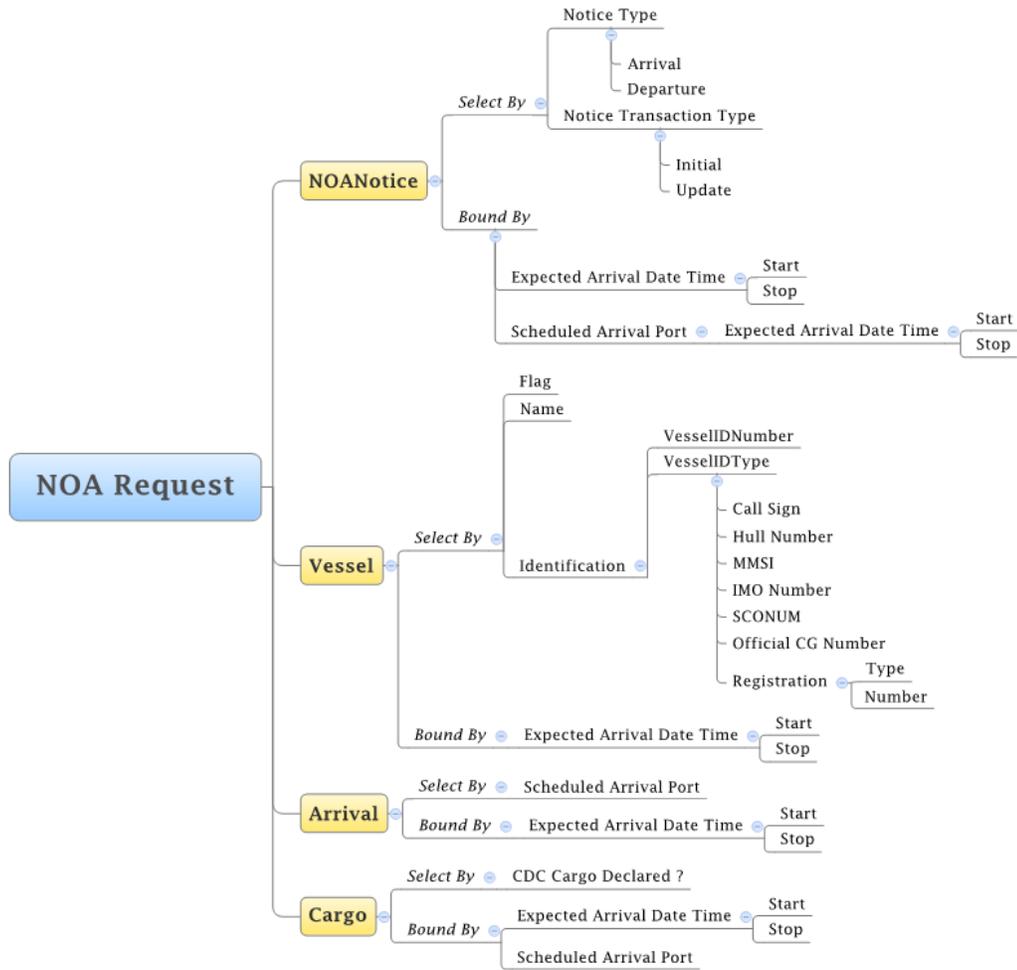
Below are graphical representations of the exchange and request/query logical models for the NIEM-M Notice of Arrival IEPD.

## 5.1. EXCHANGE MODEL

Several NOA blocks are not part of the NIEM-M EIEM and are shown in a later section.

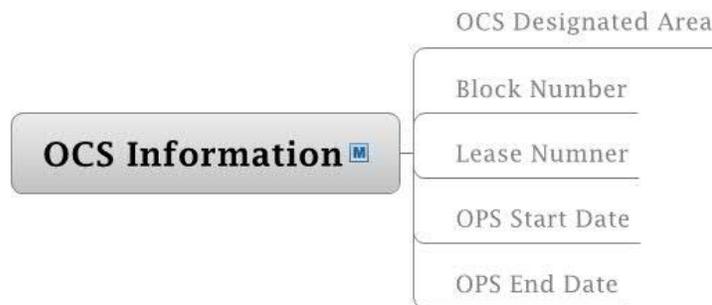


## 5.2. REQUEST/QUERY MODEL

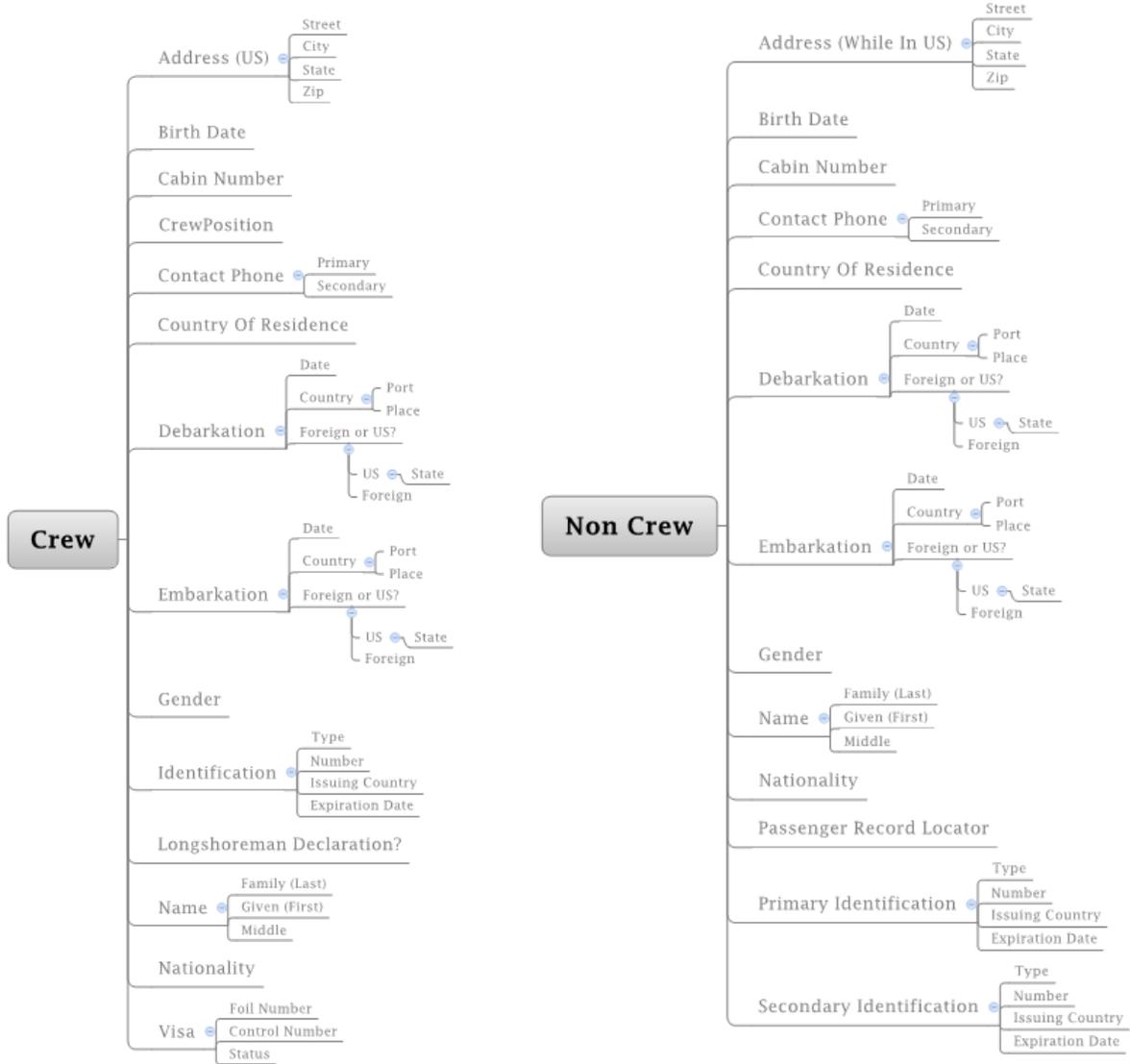


## 5.3. NOA UNIQUE OBJECTS (NON-EIEM BLOCK)

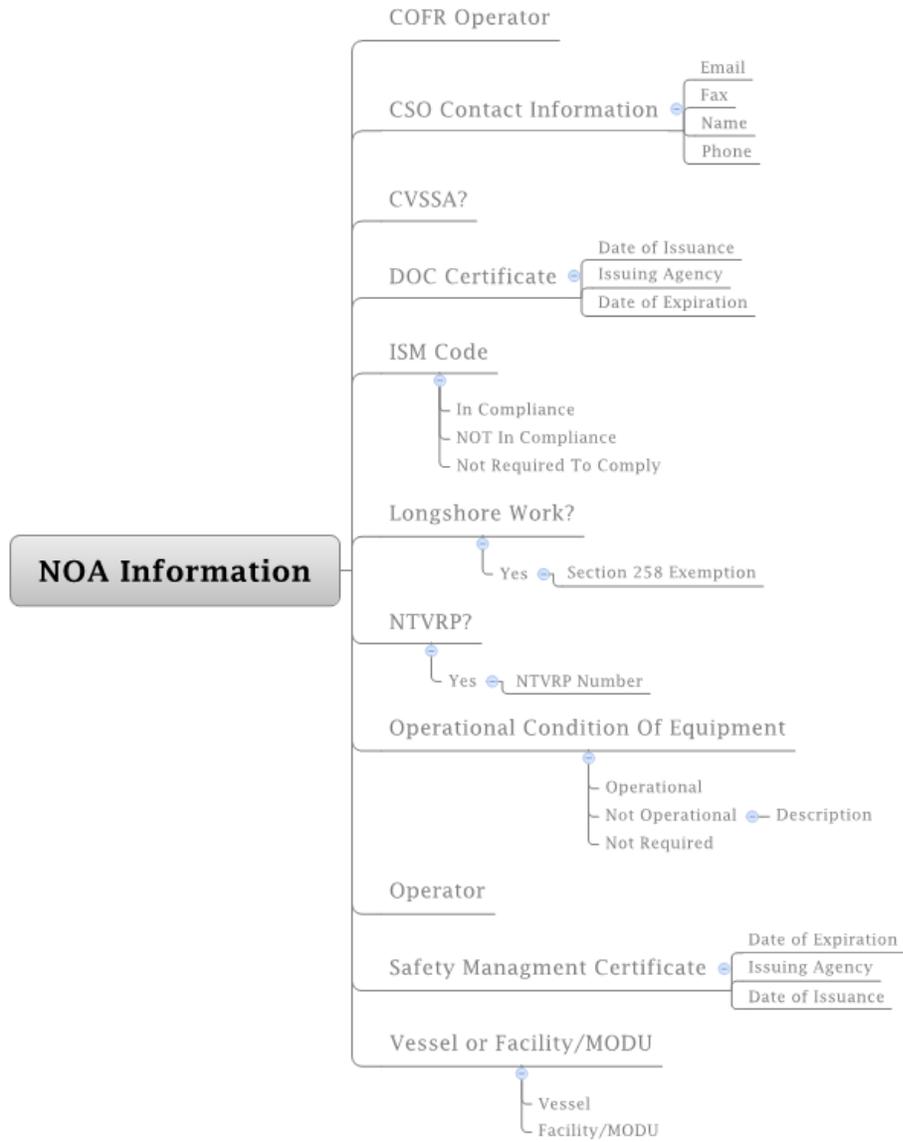
### 5.3.1. OCS INFORMATION



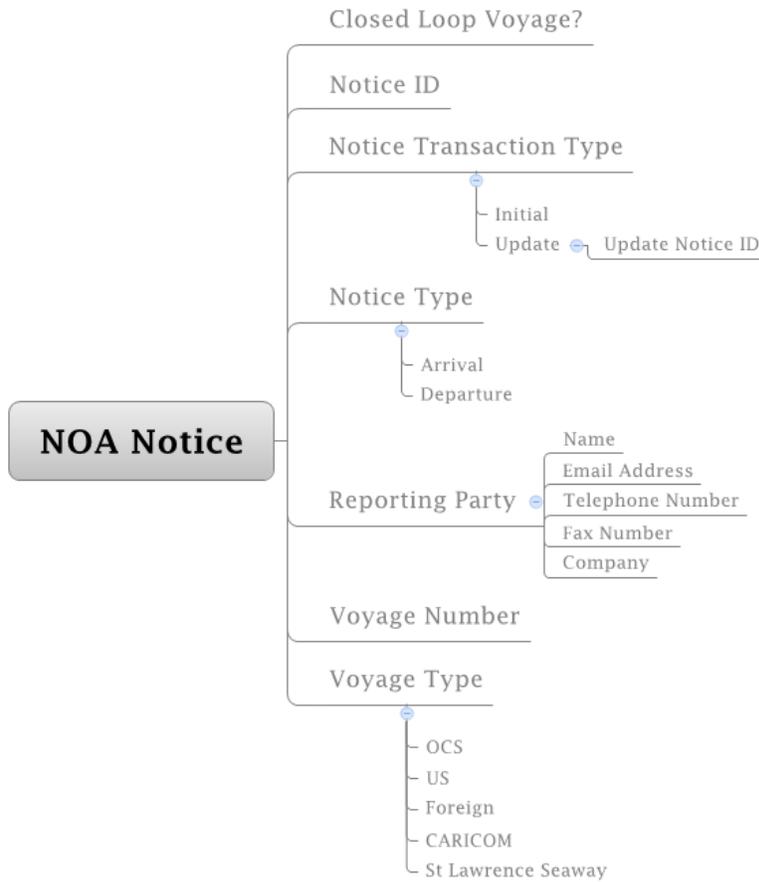
### 5.3.2. CREW & NON-CREW



5.3.3. NOA INFORMATION



### 5.3.4. NOA NOTICE

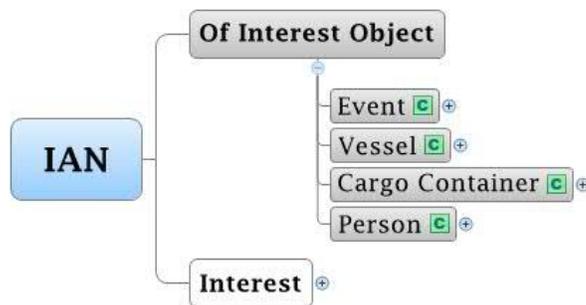


## 6. Indicators and Notifications

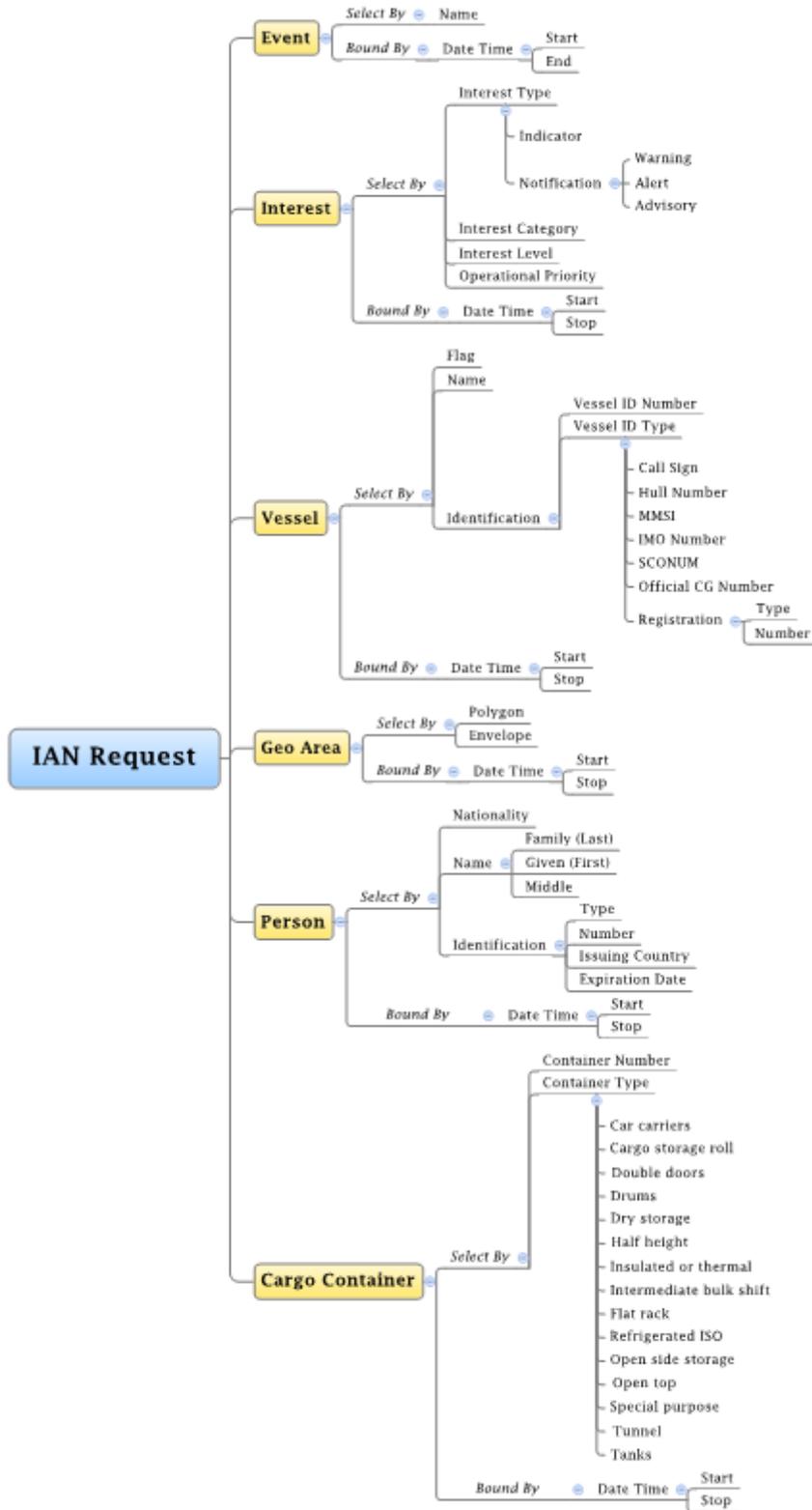
Below are graphical representations of the exchange and request/query logical models for the NIEM-M Indicators and Notification (IAN) IEPD.

### 6.1. EXCHANGE MODEL

The *Of Interest Object* block is not part of the NIEM-M EIEM so the IAN unique blocks are shown in a later section.

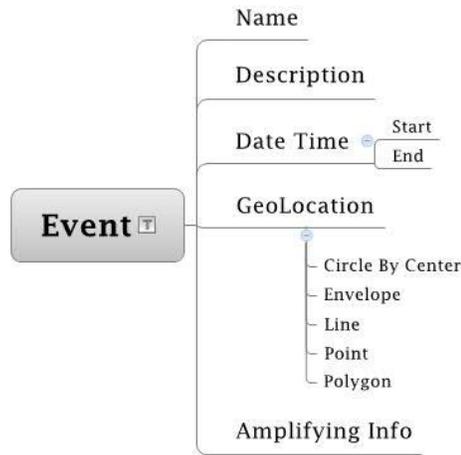


## 6.2. REQUEST/QUERY MODEL



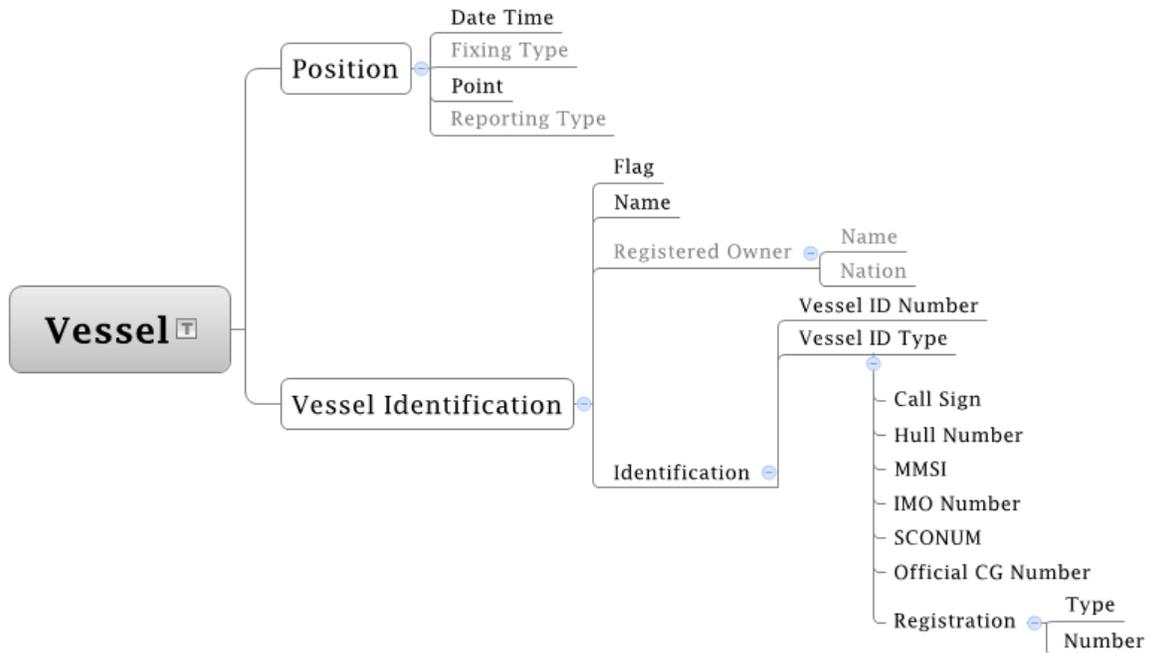
### 6.3. IAN UNIQUE OBJECTS (NON-EIEM BLOCK)

#### 6.3.1. OF INTEREST OBJECT - EVENT

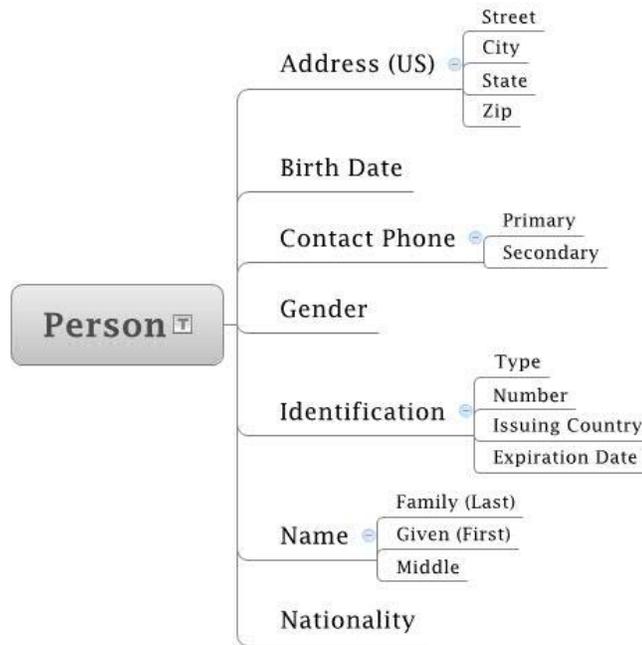


#### 6.3.2. OF INTEREST OBJECT – VESSEL

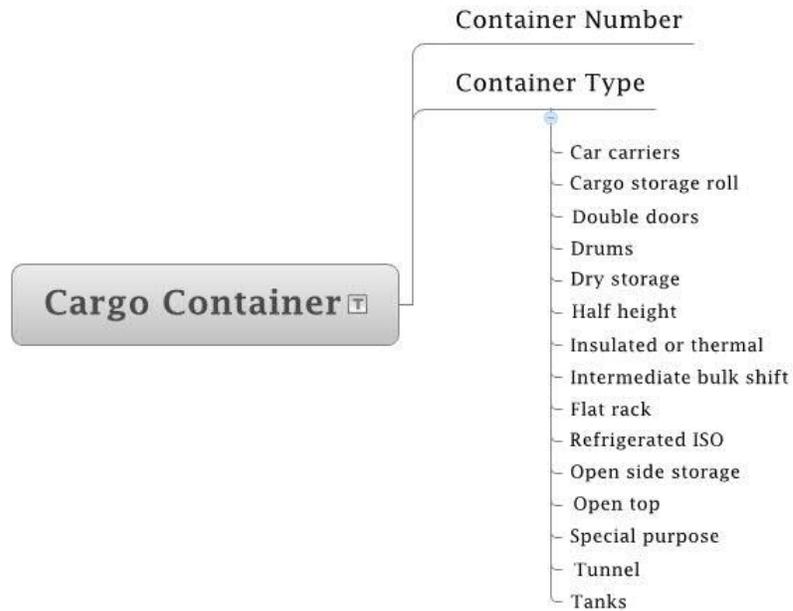
The Vessel block is made up of the *Position* and *Vessel Identification* EEIM blocks.



6.3.3. OF INTEREST OBJECT - PERSON



6.3.4. OF INTEREST OBJECT – CARGO CONTAINER

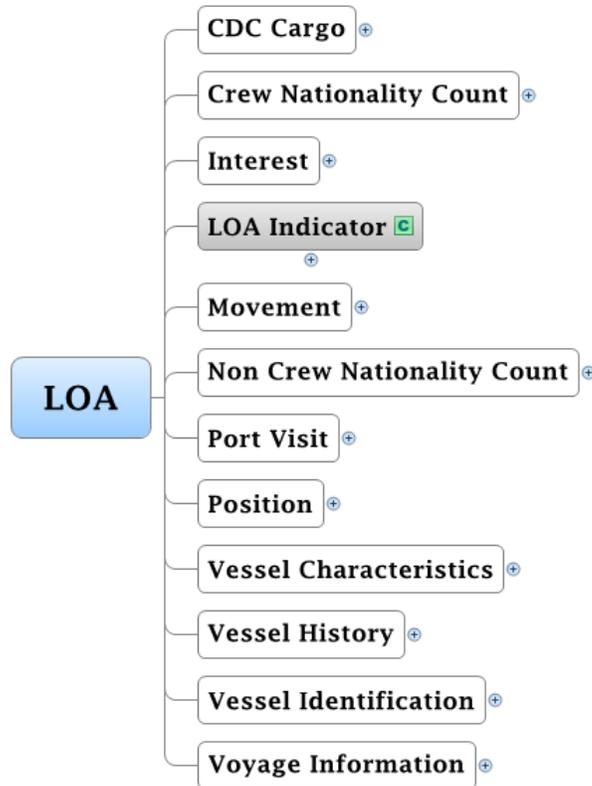


# 7. Levels of Awareness

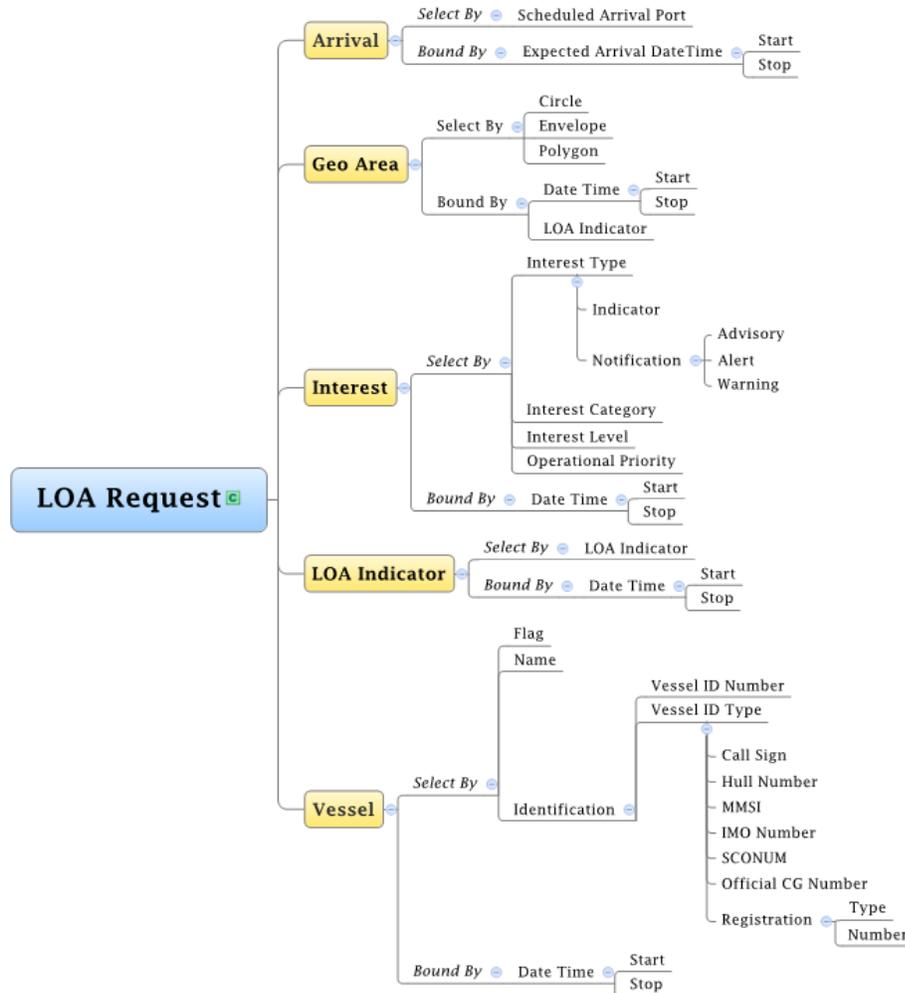
Below are graphical representations of the exchange and request/query logical models for the NIEM-M Levels Of Awareness (LOA) IEPD.

## 7.1. EXCHANGE MODEL

The *LOA Indicator* block is not part of the NIEM-M EIEM so the LOA unique block is shown in a later section.

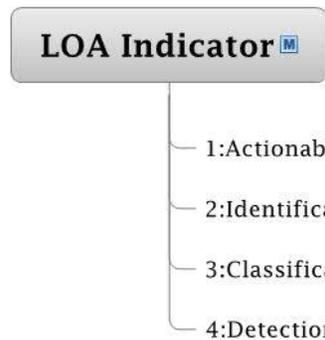


## 7.2. REQUEST/QUERY MODEL



## 7.3. LOA UNIQUE OBJECTS (NON-EIEM BLOCK)

### 7.3.1. LOA INDICATOR





---

# ATTACHMENTS

---

- NIEM-M Position Exchange Summary, Version 1.0
- NIEM-M Vessel Information Exchange Summary, Version 1.0
- NIEM-M Notice of Arrival Exchange Summary, Version 1.0
- NIEM-M Indicator and Notifications Exchange Summary, Version 1.0
- NIEM-M Levels of Awareness Exchange Summary, Version 1.0
- NIEM-M Metadata and Security Summary, Version 1.0

**VERSION 3.0**  
**RELEASE 1**  
**FEBRUARY 2015**