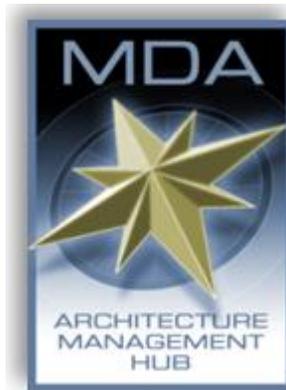


APPENDIX C - INTERFACE SECURITY SPECIFICATION



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 http://www.ietf.org/rfc/rfc2119.txt
GFIPM Trust	Federated Identity and Privilege Management (GFIPM): Cryptographic Trust Model http://www.it.ojp.gov/gist/Document/73
GFIPM Services	Federated Identity and Privilege Management (GFIPM): Web Services System-to-System Profile http://www.it.ojp.gov/gist/Document/122
SAML2 Metadata	“Metadata for the OASIS Security Markup Language (SAML) V2.0” OASIS Standard, 15 March 2005 Document Identifier: saml-metadata-2.0-os http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

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 http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
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 http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf

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.

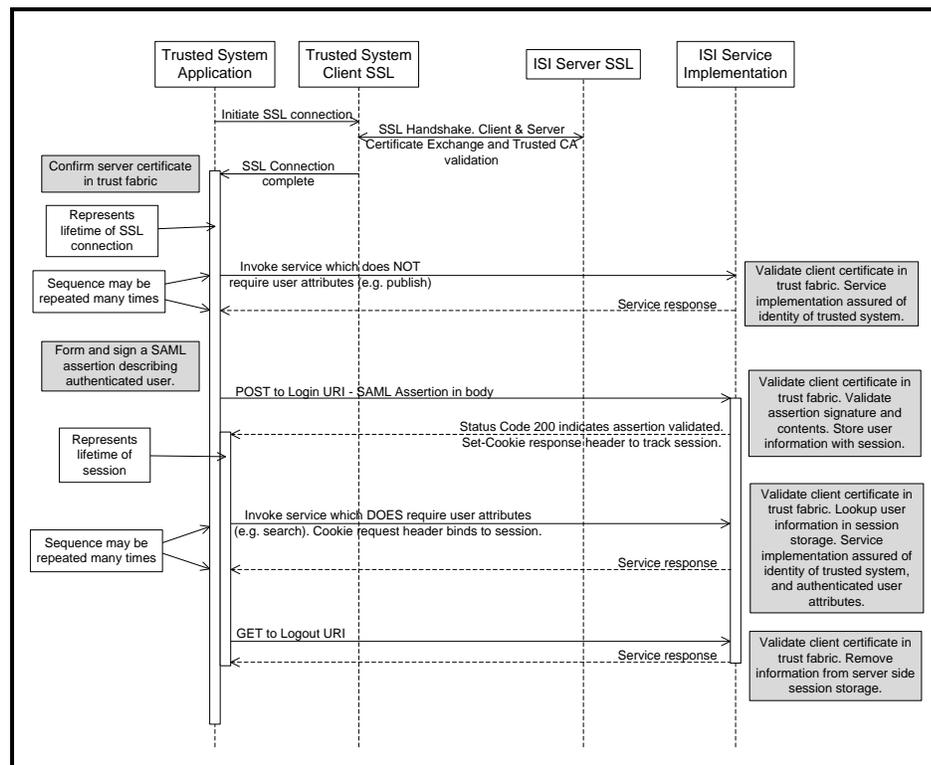


Figure 1 - Service Invocation Process Flow

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.¹ 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 RSA² 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)

¹ See http://en.wikipedia.org/wiki/Public-key_cryptography for more background information.

² 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*³. 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 OpenSSL⁴ 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.

³ See http://en.wikipedia.org/wiki/Root_certificate for more background information.

⁴ <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.

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.
 1. An existing trusted system leaves the MISE.
 2. 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.).
 3. The MISE CA public key certificate expires.
 4. 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:EntityDescriptor 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>
          <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="mise xs" />
          </ds:Transform>
        </ds:Transforms>
      </ds:Reference>
    </ds:SignedInfo>
  </ds:Signature>
</md:EntityDescriptor>
```

```

        <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">
        xsi:type="mise:MISEInfrastructureDescriptorType">
            <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: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>
<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: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>

```

```
        </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>
```

VERSION 3.0
RELEASE 1
FEBRUARY 2015