Federation Metadata Document Structure Proposal

Date	Revision Comments	Author
7/31/2008	Initial Draft Proposal	donsch

Contents

1. Problem Statement

The current draft of WS-Federation defines federation metadata as an XML document that contains elements which describe properties of endpoints. These XML elements fall into two general categories, which we will call *service instance* and *service capability* metadata elements:

- *Service instance* element: an XML element which is used to identify a physical instance (or a logically equivalent set of instances) of a specific type of web service. The "type" of the web service is indicated by the QName of the element. Physical instances (endpoints) of the service are indicated by <wsa:EndpointReference> child elements. For example, <fed:TokenIssuerEndpoints> indicates the endpoints(s) of a type of web service that issues security tokens.
- *Service capability* element: an XML element which is used to describe a property of a web service. The "type" of property is indicated by the QName of the element. Values of the property are indicated by child elements. For example, <fed:TokenSigningKeyInfo> and <fed:ClaimDialect> describe a signing key and supported claim syntax, respectively, for a token issuing service.

WS-Federation federation metadata defines three *service instance* elements which can provide meaningful information without any additional qualification in a metadata document. They are <fed:TokenIssuerEndpoints>, <fed:PseudonymServiceEndpoints> and <fed:AttributeServiceEndpoints>. Product development and testing have demonstrated that the remaining *service instance* elements, and all the *service capability* elements, are subject to the following ambiguities as currently defined.

a. There is no mechanism to indicate the correct association if multiple *service instance* and *service capability* elements exist in the same section of a federation metadata document. For example, if a there are elements for two different token issuing services and two different signing keys, there is no mechanism to indicate which key is used by which service.

b. There is no mechanism to indicated related *service instance* elements. Some types of *service instances* must be related to another type of *service instance* to be meaningful. For example, a sign-out subscription service (<fed:SingleSignOutSubscriptionEndpoints>) element is not meaningful unless it is related to a service that enables sign-on, such as an STS (<fed:TokenIssuerEndpoints>).

c. There is no *service instance* element for identifying a generic Relying Party service that is only a consumer of security tokens, attributes or pseudonyms. There are only elements for identifying services that are issuers.

WS-Federation acknowledges that a service provider may participate in multiple federations. The federation section construct allows a service provider to publish distinct metadata for each federation in which it participates in a single federation metadata document. There can be one unnamed default section, and/or multiple sections named using the FederationID attribute.

```
<fed:FederationMetadata xmlns:fed="..." ...>
    <fed:Federation [FederationID="..."] ...> +
                [Federation Metadata]
    </fed:Federation>
                [Signature]
</fed:FederationMetadata>
```

The FederationID attribute MAY be included in security token requests to specify the federation context in which the request is being made by the Relying Party. An Identity Provider MAY use the FederationID attribute to determine the policy or terms and conditions that should be applied to the request. Product development and testing have demonstrated that the federation section construct is subject to the following ambiguities as currently defined.

d. There is no mechanism or naming convention to coordinate the use of a consistent FederationID attribute value between the participants in a federation.

e. There is no mechanism for an Identity Provider to determine that a Relying Party has the right to assert a specific FederationID attribute value.

2. Solution Overview

The first three ambiguities (a, b, c) in WS-Federation federation metadata could be eliminated as follows. Define a generic *service instance* element, such as <fed:FederatedService>. Restate <fed:TokenIssuerEndpoints>, <fed:PseudonymServiceEndpoints> and <fed:AttributeServiceEndpoints> elements as derivations of the generic service element. Restate the remaining *service instance* and all *service capability* elements as child elements of the generic element, or the type-specific derivations, as appropriate.

However, defining a new generic *service instance* element in WS-Federation would appear to partially duplicate work that has already been accomplished in *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0* specification [Samlv2Meta]. The SAML <md:EntityDescriptor> element appears to have almost all of the necessary properties to address the current ambiguities in the WS-Federation definition of federation metadata. Re-using the [Samlv2Meta] constructs will simplify development and improve interoperability for parties that must either implement or deploy products based on both the WS-Federation and SAML protocols.

The remaining ambiguities (d, e) can be addressed by using the grouping property of the SAML <md:EntitiesDescriptor> element. This element can contain the applicable metadata for all the participants in a particular federation.

It is proposed that WS-Federation federation metadata be re-defined to take a normative dependency on the [Samlv2Meta] metadata document syntax and structure, and its <md:EntityDescriptor> element in particular. The following steps provide an overview of the process.

1. Re-define the WS-Federation federation metadata document structure as an extension of the [Samlv2Meta] metadata document structure. That is, eliminate <fed:FederationMetadata> and define the allowed root elements of a WS-Federation metadata document to be either an <md:EntityDescriptor> element or an <md:EntitiesDescriptor> element.

2. Correlate the WS-Federation concept of a federation to the group of service providers identified by an <md:EntitiesDescriptor> element. Map the FederationID attribute to the Name attribute of the <md:EntitiesDescriptor> element.

3. An <md:EntityDesriptor> element must include at least one role descriptor element to identify the role(s) performed by the entity. WS-Federation *service instance* elements for security token, pseudonym and attribute services define logically similar, but technically different, role concepts. The corresponding SAML metadata elements are derived from the md:SSODescriptorType abstract type. Define a parallel fed:WebServiceDescriptorType abstract type to include the WS-Federation *service capability* elements required for the *service instance* elements. Also, eliminate redundant WS-Federation *service capability* elements and use the equivalent constructs from [Samlv2Meta]. For example, cut <fed:ContactInfoAddress> and reference <md:ContactPerson> instead.

4. Derive new [Samlv3Meta] service role types that correspond to the WS-Federation security token, pseudonym and attribute *service instance* elements by extending the WebServiceDescriptorType abstract type. Then extend these new type definitions to specify their relationships to the remaining WS-Federation *service instance* elements (e.g. <fed:SingleSignOutSubscriptionEndpoints>).

3. Harmonized Metadata Details

The Root Element for a SAML metadata document is <md:EntityDescriptor> or <md:EntitiesDescriptor>.

A SAML metadata instance describes either a single entity or multiple entities. In the former case, the root element MUST be <EntityDescriptor>. In the latter case, the root element MUST be <EntitiesDescriptor>.

The <EntitiesDescriptor> element contains the metadata for an optionally named group of SAML entities. Its **EntitiesDescriptorType** *complex type contains a sequence of <EntityDescriptor> elements, <EntitiesDescriptor> elements, or both.*

The <EntityDescriptor> element specifies metadata for a single SAML entity. A single entity may act in many different roles in the support of multiple profiles. This specification directly supports the following concrete roles as well as the abstract <RoleDescriptor> element for extensibility (see subsequent sections for more details):

- *SSO Identity Provider*
- *SSO Service Provider*
- *Authentication Authority*
- *Attribute Authority*
- *Policy Decision Point*
- *Affiliation*

3.1 Define WS-Federation metadata document as an extension of [Samlv2Meta]

3.1.1 Define root elements to match [Samlv2Meta]

Revise section 3.1 Federation Metadata Document of the WS-Federation specification to indicate that the allowed root elements is either <md:EntityDescriptor> or <md:EntitiesDescriptor>.

The federation metadata document is an XML document containing a set of one or more optional XML elements that organizations can publish to proffer information that may be useful to partners for establishing a federation.

The federation metadata document MUST be of the following form:

```
<choice>
    <EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" …/>
    <EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" …/>
</choice>
```

3.1.2 Extend [Samlv2Meta] root element definitions to remove SAML restriction

The <EntitiesDescriptor> element contains the metadata for an optionally named group of SAML entities.
The <EntityDescriptor> element specifies metadata for a single SAML entity.

…

entityID [Required]

Specifies the unique identifier of the SAML entity whose metadata is described by the element's contents.

3.2 Realize WS-Federation federation concept using SAML constructs

WS-federation defines the concept of a federation, but does not define a concrete realization. SAML metadata includes a similar conceptual definition of a federation, and it also defines a concrete realization.

- WS-Federation: *A* federation *is a collection of realms that have established a producer-consumer relationship whereby one realm can provide authorized access to a resource it manages based on an identity, and possibly associated attributes, that are asserted in another realm.*
- SAML: <md:EntitiesDescriptor> is an optionally named group of entities in the context of some deployment

3.2.1 Equate WS-Federation federation concept to SAML to md:EntitiesDescriptor construct

WS-Federation describes how a service provider that participates in multiple federations MAY publish metadata for each federation in separate federation metadata documents, or in a single document. The <fed:federation> *federation section* construct is provided to enable the use of a single document. All of the participants in a particular federation can separately publish their metadata for that relationship. The metadata MAY be correlated using the FederationID label. However, there is no concrete mechanism to collectively publish all of the metadata for all of the participants in the federation.

There are production federations in the SAML community, such as the Shibboleth based InCommon federation, which have found it expedient to publish a single, authoritative metadata document for all of the participants. using the <md:EntitiesDescriptor> element. They claim to have vastly simplified partner configuration and ongoing federation management operations using this approach. Also, this approach allows a service provider to verify the identity of other participants and reliably use a “federation identifier” to determine policy or terms and conditions that should be applied to a service request.

The concrete realization of the WS-Federation federation metadata concept of federation should be defined to be the <md:EntitiesDescriptor> element. The <fed:federation> *federation section* construct should be equated to the <md:EntityDescriptor> element. That is, all of the metadata which a service provider chooses to publish about its participation in a specific federation SHOULD be published in a single <md:EntityDescriptor> element.

3.2.2 Map fed:FederationID to md:EntitiesDescriptor@Name

When a service provider or service consumer participates in multiple federations, its partners must be able to identity which of its metadata corresponds to which federation. The FederationID attribute can be used to “name” a federation and help correlate metadata between participants. But naming conventions and management of FederationID are outside the scope of the WS-Federation specification.

WS-Federation allows the FederationID attribute to be specified in WS-Trust token requests and responses, and supports the analogous wfed parameter for the passive requestor protocol. This can be used to specify the federation context in which the request is being made, and theoretically the responder can use it to determine policy or terms and conditions that should be applied when servicing the request. However, there is no convenient mechanism to verify that a requestor has the right to assert a particular FederationID.

The SAML approach of grouping all federation participants in a single <md:EntitiesDescriptor> element can address both the issues described above. The <md:EntitiesDescriptor> Name attribute provides a simple mechanism to manage a common “federation name” for all the participants. For federations like InCommon – an authority exists that is trusted to publish signed metadata for all the participants – it is straightforward for a service provider to verify that a relying party is authorized to assert the “federation name” in requests.

Therefore, the FederationID attribute is redefined as an extension to the <md:EntityDescriptor> element and mapped to the <md:EntitiesDescriptor> element’s Name attribute as follows

```
<element name="EntityDescriptor" type="md:EntityDescriptorType"/>
<complexType name="EntityDescriptorType">
        …
<attribute name="entityID" type="md:entityIDType" use="required"/>
<attribute name=”fed:FederationID” type="xs:String " use=”optional”/>
<attribute name="validUntil" type="dateTime" use="optional"/>
<attribute name="cacheDuration" type="duration" use="optional"/>
<attribute name="ID" type="ID" use="optional"/>
<anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

md:EntityDescriptorType/@fed:FederationID

This optional string attribute provides an identifier for the federation to which the federation metadata applies. When the metadata for a service provider is published as an <md:EntityDescriptor> element of a Named <md:EntitiesDescriptor> grouping, the value of the fed:FederationID attribute MUST be the same as the value of the md:Name attribute of the <md:EntitiesDescriptor> element.

3.3 Define fed:WebServiceDescriptorType abstract type

All of the concrete role definitions for the <md:EntityDescriptor> element are expressed in terms of SAML profiles and protocols. A parallel **fed:WebServiceDescriptorType** abstract type is defined to facilitate the expression of WS-Federation *service instances*.

3.3.1 Extend <md:RoleDescriptor> supported protocols

The **fed:WebServiceDescriptorType** abstract type will be derived from <md:RoleDescriptor>; its required protocolSupportEnumeration attribute must be extended to support WS-Federation and WS-Trust protocols.

> *A whitespace-delimited set of URIs that identify the set of protocol specifications supported by the role element. For SAML V2.0 entities, this set MUST include the SAML protocol namespace URI, urn:oasis:names:tc:SAML:2.0:protocol. Note that future SAML specifications might share the same namespace URI, but SHOULD provide alternate "protocol support" identifiers to ensure discrimination when necessary.*

3.3.2 Define abstract type by extending <md:RoleDescriptor>

A new abstract type is defined to include the *service capability* elements and related *service instance* elements used to describe the capabilities of WS-Federation based service offerings.

```
<complexType name="WebServiceDescriptorType" abstract="true">
    <complexContent>
        <extension base="md:RoleDescriptorType">
            <sequence>
                <element ref="fed:LogicalServiceNameOffered" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="fed:TokenTypeOffered" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="fed:ClaimDialectOffered" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="fed:ClaimTypeOffered" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="fed:AutomaticPseudonyms" minOccurs="0" maxOccurs="1"/>
                <element ref="fed:TargetScope" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="ServiceDisplayName” type="xs:String" use=”optional”/>
            <attribute name=”ServiceDescription” type="xs:String " use=”optional”/>
        </extension>
    </complexContent>
</complexType>
<element name="fed:LogicalServiceNameOffered" type="xs:anyURI"/>
<element name="fed:TokenTypeOffered" type="fed:TokenType"/>
```

```
<element name="fed:ClaimDialectOffered" type="fed:ClaimDialect"/>
<element name="fed:ClaimTypeOffered" type="auth:ClaimType"/>
<element name="fed:AutomaticPseudonyms" type="xs:boolean"/>
<element name="fed:TargetScope" type=" wsa:EndpointReferenceType "/>
```

fed:WebServiceDescriptorType/@SerivceDisplayName

This optional string attribute provides a friendly name for this service instance that can be shown in user interfaces. It is a human readable label that can be used to index metadata provided for different service instances.

fed:WebServiceDescriptorType/@SerivceDescription

This optional string attribute provides a description for this service instance that can be shown in user interfaces. It is a human readable description that can be used to understand the type of service to which the metadata applies.

fed:WebServiceDescriptorType/fed:LogicalServiceNameOffered

This optional element allows a federation metadata provider to specify to specify a "logical name" that is associated with the service. It MAY be repeated for different names. See section 3.1.5. details.

Note, this element is currently named fed:IssuerNamesOffered in the WS-Federation specification. The name should be changed to the more generic terminology proposed here.

fed:WebServiceDescriptorType/fed:TokenTypeOffered

This optional element allows a federation metadata provider to specify a token type that can be issued by the service. It MAY be repeated for different types. See section 3.1.6 for details.

fed:WebServiceDescriptorType/fed:ClaimDialectOffered

This optional element allows a federation metadata provider to specify a dialect, via an URI, that is accepted by in token requests to express the syntax for requested claims. It MAY be repeated for different dialects. See section 3.1.7 for details.

fed:WebServiceDescriptorType/fed:ClaimTypeOffered

This optional element allows a federation metadata provider to specify an offered claim type, using the schema provided by the common claim dialect defined in this specification, that can be asserted in security tokens issued by the service. It MAY be repeated for different types. See section 3.1.8 for details.

fed:WebServiceDescriptorType/fed:AutomaticPseudonyms

This optional element allows a federation metadata provider to indicate if it automatically maps pseudonyms or applies some form of identity mapping. See section 3.1.9 for details.

3.3.3 Eliminate duplicate WS-Federation elements

Some of the service capability elements currently defined in WS-Federation should be eliminated in favor of equivalent elements that already exist in either the <md:EntityDescriptor> element or the <md:RoleDescriptor> element.

- <md:ContactPerson> should be used in place of <fed:ContactInfoEndpoints>
- <md:KeyDescriptor> should be used in place of <fed:TokenSigningKeyInfo>
- <md:KeyDescriptor> should be used in place of <fed:TokenKeyTransferInfo>
- <md:AdditionalMetadataLocation> should be used in place of <fed:TokenIssuerMetadata>

3.4 Derive types for WS-Federation IP and RP web services

New complex *service types* for Security Token, Attribute and Pseudonym services are derived from **fed:WebServiceDescriptorType** as described in the following sections. These types will be used to extend

<md:RoleDescriptor> to create *service roles* which are similar to <md:IDPSSODescriptor>. A new complex generic application *service type* is also derived from **fed:WebServiceDescriptorType** . This type will be used to extend <md:RoleDescriptor> to create a *service role* which is similar to <md:SPSSODescriptor>.

3.4.1 <fed:SecurityTokenServiceType>

```
<complexType name="SecurityTokenServiceType">
    <extension base="fed:WebServiceDescriptorType">
        <sequence>
            <element ref="fed:SecurityTokenServiceEndpoint" minOccurs="1" maxOccurs="unbounded"/>
            <element ref="fed:SingleSignOutSubscriptionEndpoint" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="fed:SingleSignOutNotificationEndpoint" minOccurs="0" maxOccurs="unbounded"/>
            <element ref="fed:PassiveRequestorEndpoint" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </extension>
</complexType>
<element name="fed:SecurityTokenServiceEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:SingleSignOutSubscriptionEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:SingleSignOutNotificationEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:PassiveRequestorEndpoint" type="wsa:EndpointReferenceType"/>
```

These definitions apply to the derived type listed in the schema outlined above.

fed:SecurityTokenServiceType/fed:SecurityTokenSerivceEndpoint

This required element specifies the endpoint address of a security token service that supports the WS-Federation and WS-Trust interfaces. Its contents MUST an endpoint reference as defined by [WS-Addressing] that provides a transport address for the security token service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

Note this corresponds to the <fed:TokenIssuerEndpoints> from section 3.1.5 of WS-Federation which is renamed and subsumed here.

fed:SecurityTokenServiceType/fed:SingleSignOutSubscriptionSerivceEndpoint

This optional element specifies the endpoint address of a service which can be used to subscribe to federated sign-out messages. Its contents MUST an endpoint reference as defined by [WS-Addressing] that provides a transport address for the subscription service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

fed:SecurityTokennServiceType/fed:SingleSignOutNotificationSerivceEndpoint

This optional element specifies the endpoint address of a service to which push notifications of sign-out are to be sent. Its contents MUST be an endpoint reference as defined by [WS-Addressing] that provides a transport address for the notification service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

fed:SecurityTokenServiceType/fed:PassiveRequestorEndpoint

This optional element specifies the endpoint address of a service that supports the WS-Federation Web (Passive) Requestor protocol. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

3.4.2 <fed:PseudonymServiceType>

```
</complexType>
<complexType name="PseudonymServiceType">
    <extension base="fed:WebServiceDescriptorType">
        <sequence>
            <element ref="fed:PseudonymServiceEndpoint" minOccurs="1" maxOccurs="unbounded"/>
```

```
                <element ref="fed:SingleSignOutNotificationEndpoint" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
    </extension>
</complexType>
<element name="fed:PseudonymServiceEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:SingleSignOutNotificationEndpoint" type="wsa:EndpointReferenceType"/>
```

These definitions apply to the derived type listed in the schema outlined above.

fed:PseudonymServiceType/fed:PseudonymSerivceEndpoint

This required element specifies the endpoint address of a pseudonym service that supports the WS-Federation and WS-Trust interfaces. Its contents MUST an endpoint reference as defined by [WS-Addressing] that provides a transport address for the pseudonym service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

fed:PseudonymServiceType/fed:SingleSignOutNotificationSerivceEndpoint

This optional element specifies the endpoint address of a service to which push notifications of sign-out are to be sent. Its contents MUST be an endpoint reference as defined by [WS-Addressing] that provides a transport address for the notification service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

3.4.3 <fed:AttributeServiceType>

```
<complexType name="AttributeServiceType">
    <extension base="fed:WebServiceDescriptorType">
          <sequence>
                <element ref="fed:AttributeServiceEndpoint" minOccurs="1" maxOccurs="unbounded"/>
                <element ref="fed:SingleSignOutNotificationEndpoint" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
    </extension>
</complexType>
<element name="fed:AttributeServiceEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:SingleSignOutNotificationEndpoint" type="wsa:EndpointReferenceType"/>
```

These definitions apply to the derived type listed in the schema outlined above.

fed:AttributeServiceType/fed:AttributeSerivceEndpoint

This required element specifies the endpoint address of an attribute service that supports the WS-Federation and WS-Trust interfaces. Its contents MUST an endpoint reference as defined by [WS-Addressing] that provides a transport address for the attribute service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

fed:AttributeServiceType/fed:SingleSignOutNotificationSerivceEndpoint

This optional element specifies the endpoint address of a service to which push notifications of sign-out are to be sent. Its contents MUST be an endpoint reference as defined by [WS-Addressing] that provides a transport address for the notification service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

3.4.4 <fed:ApplicationServiceType>

```
<complexType name="ApplicationServiceType">      <extension base="fed:WebServiceDescriptorType">
          <sequence>
                <element ref="fed:ApplicationServiceEndpoint" minOccurs="1" maxOccurs="unbounded"/>
                <element ref="fed:SingleSignOutNotificationEndpoint" minOccurs="0" maxOccurs="unbounded"/>
                <element ref="fed:PassiveRequestorEndpoint" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
    </extension>
```

```
</complexType>
<element name="fed:ApplicationServiceEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:SingleSignOutNotificationEndpoint" type="wsa:EndpointReferenceType"/>
<element name="fed:PassiveRequestorEndpoint" type="wsa:EndpointReferenceType"/>
```

These definitions apply to the derived type listed in the schema outlined above.

fed:ApplicationServiceType/fed:ApplicationSerivceEndpoint

This required element specifies the endpoint address of a Relying Party application service that supports the WS-Federation and WS-Trust interfaces. Its contents MUST an endpoint reference as defined by [WS-Addressing] that provides a transport address for the application service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

fed:ApplicationServiceType/fed:SingleSignOutNotificationSerivceEndpoint

This optional element specifies the endpoint address of a service to which push notifications of sign-out are to be sent. Its contents MUST be an endpoint reference as defined by [WS-Addressing] that provides a transport address for the notification service. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

fed:ApplicationServiceType/fed:PassiveRequestorEndpoint

This optional element specifies the endpoint address of a service that supports the WS-Federation Web (Passive) Requestor protocol. It MAY be repeated for different, but functionally equivalent, endpoints of the same logical *service instance*.

3.5 Extend md:RoleDescriptor for WS-Federation IP and RP service roles

There are concrete service roles defined for <md:EntityDescriptor> which are similar to roles performed by some of the WS-Federation *service instances*. The SAML <md:IDPSSODescriptor> element defines a role similar to that of the WS-Federation <fed:TokenIssuerEndpoints> element and the <md:AttributeAuthorityDescriptor> element corresponds to the <fed:AttributeServiceEndpoints> element. There is no direct [Samlv2Meta] corollary for the WS-Federation <fed:PseudonymServiceEndpoints> element.

The service roles for these three WS-Federation Identity Provider services, and for a generic Relying Party application service, are derived from <md:RoleDescriptor> using the xsi:type extensibility mechanism.

3.5.1 fed:SecurityTokenService Role

An <md:EntityDescriptor> that provides a WS-Federation based security token service is indicated by using the <md:RoleDescriptor> extensibility point as follows.

```
<EntityDescriptor xmlns="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#"
    entityID="…">
    <ds:Signature>...</ds:Signature>
    <RoleDescriptor xsi:type="fed:SecurityTokenServiceType"
        protocolSupportEnumeration="http://docs.oasis-open.org/wsfed/federation/200706"
        "http://docs.oasis-open.org/ws-sx/ws-trust/200512">
        …
    </RoleDescriptor>
    …
</EntityDescriptor>
```

3.5.2 fed:PseudonymService Role

An <md:EntityDescriptor> that provides a WS-Federation based pseudonym service is indicated by using the <md:RoleDescriptor> extensibility point as follows.

```
<EntityDescriptor xmlns="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#"
    entityID="…">
    <ds:Signature>...</ds:Signature>
    <RoleDescriptor xsi:type="fed:PseudonymServiceType"
        protocolSupportEnumeration="http://docs.oasis-open.org/wsfed/federation/200706"
        "http://docs.oasis-open.org/ws-sx/ws-trust/200512">
        …
    </RoleDescriptor>
    …
</EntityDescriptor>
```

3.5.3 fed:AttributeService Role

An <md:EntityDescriptor> that provides a WS-Federation based atribute service is indicated by using the <md:RoleDescriptor> extensibility point as follows.

```
<EntityDescriptor xmlns="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#"
    entityID="…">
    <ds:Signature>...</ds:Signature>
    <RoleDescriptor xsi:type="fed:AttributeServiceType"
        protocolSupportEnumeration="http://docs.oasis-open.org/wsfed/federation/200706"
        "http://docs.oasis-open.org/ws-sx/ws-trust/200512">
        …
    </RoleDescriptor>
    …
</EntityDescriptor>
```

3.5.3 fed:ApplicationService Role

An <md:EntityDescriptor> that provides a WS-Federation based application service is indicated by using the <md:RoleDescriptor> extensibility point as follows.

```
<EntityDescriptor xmlns="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#"
    entityID="…">
    <ds:Signature>...</ds:Signature>
    <RoleDescriptor xsi:type="fed:ApplicationServiceType"
        protocolSupportEnumeration="http://docs.oasis-open.org/wsfed/federation/200706"
        "http://docs.oasis-open.org/ws-sx/ws-trust/200512">
        …
    </RoleDescriptor>
    …
</EntityDescriptor>
```

4. References

[Samlv2Meta]	Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS SSTC, September 2004. Document ID sstc-saml-metadata-2.0-cd-03. http://www.oasisopen.org/committees/security/
[WS-Federation]	Web Services Federation Language (WS-Federation) Version 1.2 Editors Draft-07. OASIS WSFED TC, June 10, 2008. http://docs.oasis-open.org/wsfed/federation/200706/ws-federation-1.2-spec-ed-01.doc