PVP2-S Metadata Profile

Contents

1 Audience, Authority and Purpose

Audience. This document is targeted to the federation operator and participants of "Portalverbundvereinbarung" (PVV – Austrian intra-government federation) and other parties having an interface to that federation.
The document is written in English to facilitate review and know-how exchange with other European federation operators.

Authority. This specification is the result of the project "PVP2 zentrale Dienste" of the AG-IZ working group, which is a sub-group of the e-Gov cooperation board "BLSG" that joins federal, state and local government with other public sector bodies.

Purpose. Both federations mentioned above are growing at a considerable pace. To provide clean and efficient technical interfaces for deployments, and support governance, risk and control to essential infrastructure components of these e-government services a centralized repository for federation-wide metadata along with defined management processes is required.

2 Acknowledgement

This document is based on the OASIS SSTC standards. In addition it draws from lists, wikis and workshops in the Shibboleth, Terena/REFEDS, Feide and IIW/EWTI communities.

3 Requirements for Metadata and Trust Management

3.1 General Needs and Objectives

Metadata is used to communicate the configuration of SAML entities (primarily IDP and SP roles) in a trustworthy manner. A key service of federations is the aggregation of the metadata of participating entities. It is achieved by providing a list of entities in a machine-readable format that is conformant to the federation policy. This helps to simplify the deployment of services by providing a single point of acquisition for the metadata of many entities, and by having the federation operator act as a broker of technical trust.
Metadata shall be governed by a policy that defines the responsibilities and limitations of the organizations creating and aggregating metadata.

3.2 Key Requirements

- List entities of Federation Participants in good standing. Federation Participants operate *entities* like IdP and SP.
- For each entity maintain data about:
    - Service endpoints;
    - Valid keys (certificates);
    - Supported cipher suites for flexible choice of algorithm support;
    - Other properties, e.g. a SP's required attributes.
- Link entities to organizations that operate and own entities.
- The federation operator must enforce federation policy by asserting that entity data is authentic.

- Aggregate and distribute above data in a secure and timely manner.
- Organizational and technical controls must be appropriate to the security level, e.g. for audit logging, key management, revocation and business continuity.

3.3 Trust on Business and Technical Levels

The Federation Operator has to operate according to the federation policy, like onboarding and terminating Federation Participants, requiring audits etc. Metadata is a collection of EntityDescriptors (see below) that are created and maintained by Federation Participants and authenticated by the Federation Operator. The trust chain is established like this:

1. Organization applies for participation;
2. Federation Operator (FO) checks and confirms;
3. Organization becomes a (Federation) Participant, and a highly secure method of mutual authentication is established;
4. FO shares the certificate of the metadata signature with the Participant;
5. Participant submits an EntityDescriptor to FO;
6. FO checks if the EntityDescriptor
   a. Is syntactically correct;
   b. Is authentic (bound to the Participant);
   c. Has Entity Attributes that match the registered names and values in the federation participant file (i.e. eligibility to receive attribute bundles like eGovToken or CitizenToken);
   d. Does not contain any elements or attributes that are not contained in the metadata profile (i.e. not agreed upon custom extensions)[1];
   e. Has a validity is between 4 and 24h into the future.
7. FO adds the EntityDescriptor to the Metadata Aggregator, links it to ldap.gv.at and adds a signature;
8. Any Participant may retrieve the Entity Descriptor and validate its signature.

[1] Rationale: The FO should not sign unknown data elements.

4 Structural Overview

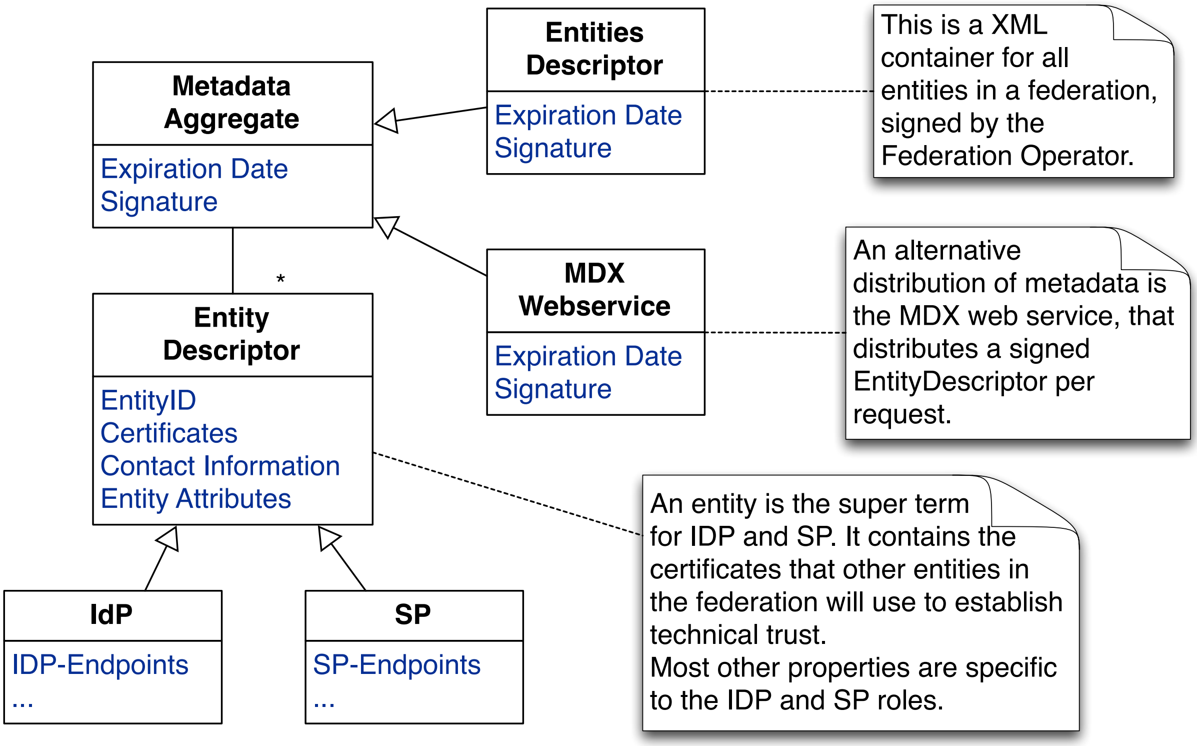
Entities Descriptor
Expiration Date
Signature
This is a XML container for all entities in a federation, signed by the Federation Operator.
Metadata Aggregate
Expiration Date
Signature
*
MDX Webservice
Expiration Date
Signature
An alternative distribution of metadata is the MDX web service, that distributes a signed EntityDescriptor per request.
Entity Descriptor
EntityID
Certificates
Contact Information
Entity Attributes
An entity is the super term for IDP and SP. It contains the certificates that other entities in the federation will use to establish technical trust.
Most other properties are specific to the IDP and SP roles.
IdP
IDP-Endpoints
...
SP
SP-Endpoints
...

Fig. 1 Metadata structure

Metadata can be grouped into two main parts. First the EntityDescriptor that describes a single IDP or SP. Second, the aggregation service that makes those EntityDescriptors available to the federation.

5 Use Cases

5.1 Assumptions

Interfederation is not considered as a separate use case in this document. To implement Interfederation entities may (i) join multiple federations (a.k.a. multi-homing), or (ii) a Federation Operator may accept another federation's metadata feed (a kind of roaming arrangement), or (iii) entities may join a federation due to a law. From the perspective of this document these cases handled equal to entities that are direct federation participants.

Examples of these constellations are (i) a commercially operated service offering services to PVV and USP users with individual contracts, (ii) a government-operated service offering services to PVV and USP users under a common agreement governing USP and all PVV members, and (iii) a Citizen Card IDP authenticating citizens to PVV-member SPs.

Note: While multi-homing is the simplest and most obvious method for an entity to join multiple federations, there might be a non-trivial impact on service and idp discovery services that have not been analyzed yet.

5.2 Actors

The basic actors in a federation are Federation Operator, IdP and SP. To create a more specific design, [FeideMaRequ] introduces additional terms for the metadata management perspective, distinguishing between components and their roles. There are the roles:

- *Metadata Publishers* (MP)
- *Metadata Consumer* (MC)

And the components:

- *Identity Provider* (IdP)
- *Service Provider* (SP)
- *Metadata Aggregator* (MA)
- (Federation) *Metadata Registrar* (MR)

A *Metadata Aggregator* (MA) collects metadata from one or more *Metadata Publishers* (MP) **and** publishes validated and filtered metadata for one or more *Metadata Consumers* (MC) based on configuration and rules for the federation.

In its current version the profile is targeted for a simple federation constellation, with the Federation Operator acting as MR and MA, and entities playing both MP and MC as depicted below in Fig. 2. For other constellations including interfederation see [FeideMaRequ].

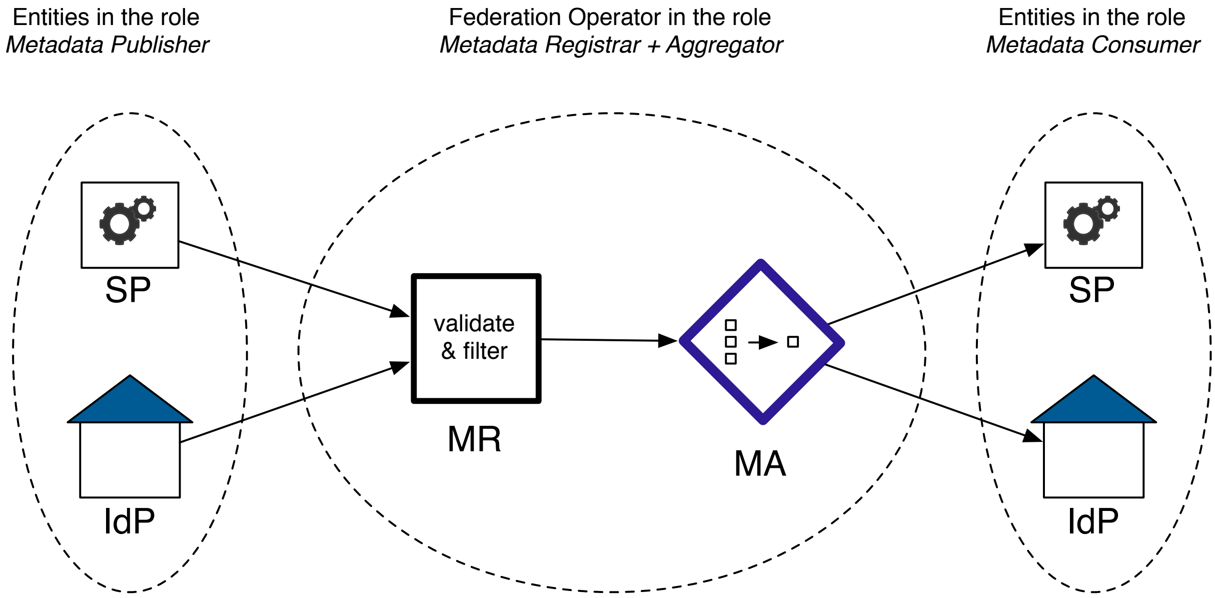

Fig. 2 Metadata flow in a federation

5.3 Use Case Overview

The following use case descriptions (see Fig. 3*)* therefore simplify actors into entities (IdP/SP) and federation operator (FO).

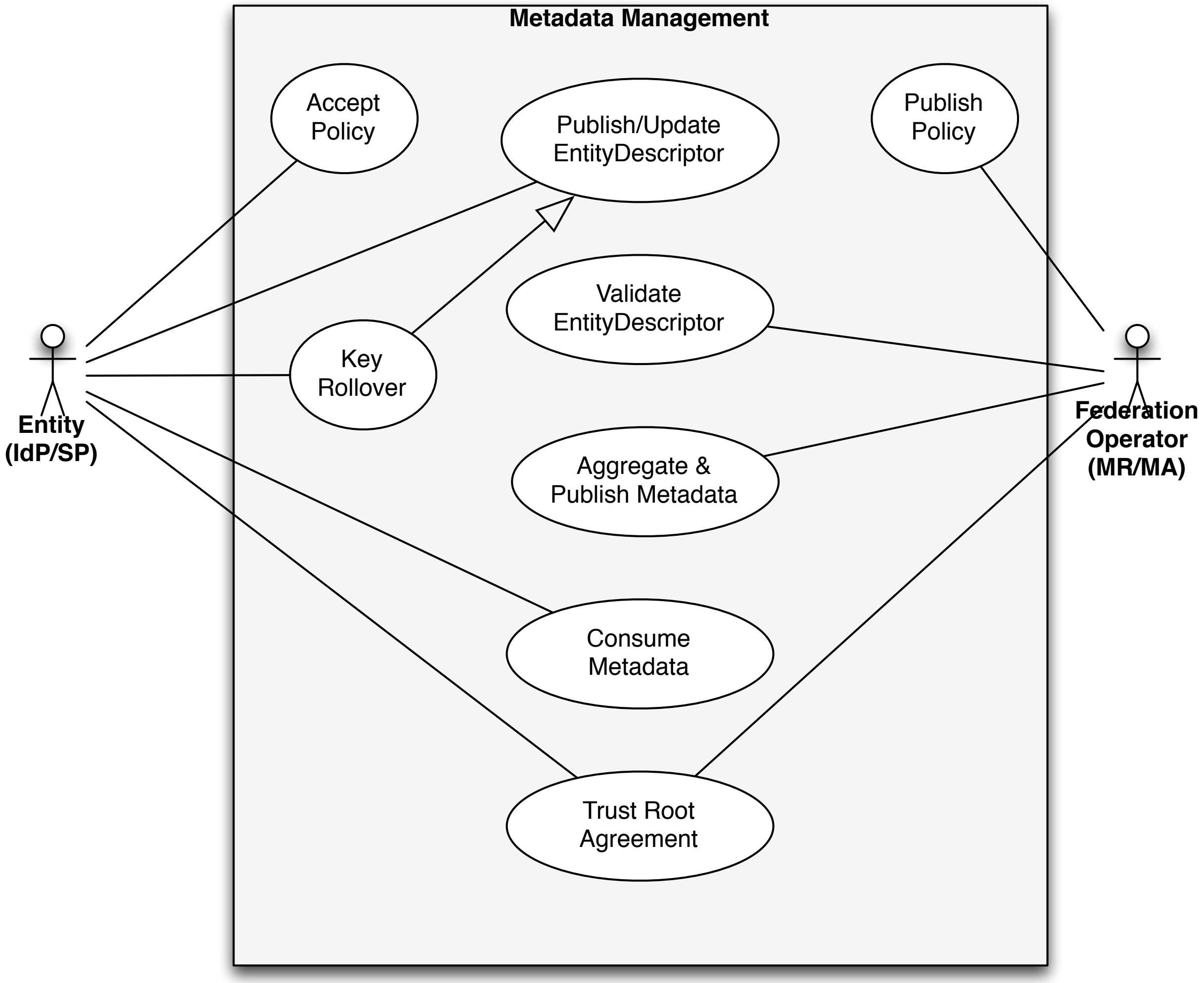
Metadata Management
Accept Policy
Publish/Update EntityDescriptor
Publish Policy
Validate EntityDescriptor
Key Rollover
Aggregate & Publish Metadata
Consume Metadata
Trust Root Agreement
Entity (IdP/SP)
Federation Operator (MR/MA)

Fig. 3 Use Case Overview

5.4 Use Case "Publish/Update Entity Descriptor"

Entities SHOULD publish its metadata using the Well-Known Location method defined in [SAML2Meta][2]. There are 4 options to transfer the EntityDescriptor to the FO in a secure way:
a) Sign and Pull. The Entity will sign the XML infoset, the FO will pull it from the Well-Known Location. (Minor) disadvantage: The FO must manage a list of Well-Known Locations for federation members and refresh in regular intervals.
b) Sign and Push. The Entity will sign the XML document and transmit it via a reliable channel, such as a web service or an electronic delivery service.
c) Pull over HTTPS. The FO will pull the XML infoset from an HTTPS-protected location. Disadvantage: For security reasons the TLS client certificate must be validated against a whitelist.
d) Out-of-Band Channel: Entity and FO agree on some other secure and fast enough transfer.
For pull methods the FO is required to refresh the metadata every 15 minutes.

5.4.1 Key Rollover

Certificates contained in metadata must be valid with respect to the "PKIX Trust Model" with its rules for expiration and path validation. For operational purposes it is strongly recommended to make both the old and new certificate available for a transition period. This period should be long enough to allow for manual importation of certificates in deployments that cannot dynamically load keys from metadata (e.g. 2 weeks).

5.5 Use Case "Validate EntityDescriptor"

Every EntityDescriptor MUST be validated using the XML signature in either the EntityDescriptor or EntitiesDescriptor.
Note: Retrieval from a URL that is authenticated with a TLS is insufficient. Without checking the XML signature.

5.6 Use Case "Aggregate and Publish Metadata"

Metadata MUST be signed by the FO **after** pruning existing signatures.

5.6.1 Publishing the Aggregated Metadata as single XML Infoset

The FO MUST publish the metadata of all federation members as a XML infoset at a federation-defined URL. The FO must sign the metadata at the `<EntitiesDescriptor>` level.

5.6.2 Publishing each `<EntityDescriptor>` via the MDX Web Service

The FO SHOULD publish Entity Descriptors using the Metadata Query Protocol [MDX draft].

5.7 Use Case "Consume Metadata"

Entities are obliged to consume metadata at least daily from the FO, validate the signature and update their services if changes were detected. Update intervals are implied by the `validUntil` attribute.
If a product does not read and refresh metadata directly from the metadata feed, deployers need to implement helper scripts to verify the signature, check for updates, import updates into the product configuration and reload the service on a regular basis.
Note: It is important to understand that the PVP federations are based on a model of explicit trust, i.e. certificates are white-listed in metadata, as opposed to black-listing

[2] I.e. by dereferencing the EntityID

invalid certificates with a revocation mechanism. There is no benefit to read CRLs or OSCP servers for entity certificates.

5.8 Use Case "Trust Root Agreement"

The FO certificate used to sign metadata is the trust anchor for the technical trust in the federation. The certificate needs to be shared in a secure ceremony that is defined in a federation specific policy document. E.g. working group meetings can be used for key sharing parties.

6 Metadata Format, Contents and Rules

Identity Providers and Service Providers MUST provide a SAML 2.0 Metadata document representing its entity. Provided metadata MUST conform to the SAML V2.0 Metadata Interoperability Profile Version 1.0 [SAMLMetaIOP].

Editor Note. This section needs to be cross-verified with the schema definitions for completeness and correctness.

6.1 Metadata Signature

The trust root is established and maintained according to the following rules:

- A MA (metadata aggregator) MUST have a signing key in a form of a X.509 certificate. The MA MUST generate and protect the associated private key using an HSM.
- The MA MUST share the signing key out-of-band using a ceremony based on a written policy. The ceremony MUST include means that allow Metadata Consumers (MC) to verify the signing key with its fingerprint.
- MCs MUST whitelist the public key when validating a metadata signature.
- Metadata Consumers MUST NOT use standard certificate validation mechanisms (path validation and revocation), as this falls back to (weak) commercial grade PKIX.
- The issuer of the certificate is ignored, and hence should be self-signed.
- The MA MUST have an out-of-band mechanism to revoke a compromised metadata signing key.

6.2 Common Elements for Entities (for IdP and SP)

6.2.1 Entity ID

The value of the `EntityDescriptor@entityID` attribute SHOULD be the canonical URL of the entity's metadata document. Canonical URLs follow the semantic-preserving normalization as specified in RFC 3986 section 6.

E.g. https://testsp.xyz.tld/sp.xml, *but not* https://testsp.xyz.tld:443/sp.xml.

Note: A common misconception is that the entity ID must match the endpoint locations in the EntityDescriptor. Unlike the endpoint locations, the entity ID accurately reflects the organization that owns the entity.

6.2.2 Key Material for IDP and SP

A SAML entity uses public key cryptography to secure the data transmitted to trusted partners. Public keys are published in the form of X.509 certificates in metadata whereas the corresponding private keys are held securely by the entity. These keys are used for message-level signing and encryption, and to create secure back channels for transporting SAML messages over SSL/TLS. They are **not** used for browser-facing SSL/TLS transactions on port 443.

6.2.2.1 Trust Models

The primary trust model (as described in [SAMLMetaIOP]) validates certificates by looking up their public key against metadata, thus implementing white-listing. The metadata signature provides the trust anchor for those keys in metadata.

The PKIX trust model is supported in parallel to facilitate compatibility with products that do not support the first model. As it uses black-listing, the first model is considered superior. Certificates are also distributed using SAML metadata in the PKIX model.

6.2.2.2 X.509 Certificates in Federation Metadata

This profile sets the following security and trust requirements around certificates included in federation metadata:

- The use of **long-lived certificates** in federation metadata with lifetimes between 10 and 20 years is strongly RECOMMENDED to reduce unnecessary technically imposed deadlines on key rollover.
- Service providers MAY include a separate encryption key in SP metadata, but it is not used according to the current PVP2 SAML profile.
- Keys and certificates must conform to PVP2-S-Profil V2.1 section 3.1.1. (i.e. mandatory support of RSA-SHA2).
- The decision to generate a new private key and submit a certificate with a new public key is subject to the federation member's policy, or necessity in the event of a suspected key compromise.
- Expired certificates MUST NOT be introduced into federation metadata, and MUST be removed once a certificate migration process to a new certificate has been completed.
- For key management purposes multiple certificates per role descriptor are allowed at any time. Products that do not allow this (reported for ADFS 2.0) must preprocess metadata appropriately to make it interoperable.
- To support ADFS 2.0 IDPs it is necessary to have a unique key and certificate per SP, even if those SPs are operated in the same service context.
- The Federation Operator does not validate Subject information in certificates because this information is irrelevant to the federated security context. However, at its own discretion, the Federation Operator may reject metadata submissions if that submission contains a certificate with fields that contain egregiously misrepresented Subject information. Generally, subject information should express a somewhat reasonable relationship between the certificate and its owner.

6.2.2.3 Key Usage

Using the same key for signing and TLS (usage="sign") and a different key for encryption (usage="encrypt") is recommended.

6.2.2.4 Key Values

For accessible documentation a plain text version of the certificate SHOULD be included as XML comment with the certificate string, like:

```
<ds:X509Certificate>
    MIIGdDCCBVygAwIBAgIDCwUIMA0GCSqGSIb3DQEBBQUAMIGMMQswCQYDVQQGEwJJ
    ...
    0f9WF/FNNfefMLfNVxu3A0XZYXdjYNf7
    <!-- Certificate:
    Data:
       Version: 3 (0x2)
       Serial Number: 722184 (0xb0508)
       Signature Algorithm: sha1WithRSAEncryption
       Issuer: C=IL, O=StartCom Ltd., OU=Secure Digital Certificate
Signing, CN=StartCom Class 1 Primary Intermediate Server CA
       Validity
          Not Before: Jul  6 01:08:03 2013 GMT
          Not After : Jul  7 16:27:49 2014 GMT
       Subject: description=X1mvpNs3C87MSNKw, C=AT,
CN=testshib.portalverbund.at/emailAddress=hostmaster@portalverbund.at
    -->
</ds:X509Certificate>
```

6.2.3 Algorithm Support

In theory weak algorithms should be substituted within reasonable time frames. Yet practice has shown that not only suspicious but also broken algorithms stay in production systems for many years. With the overdue replacement (as of 2013) of SHA1 and CBC and the upcoming move to elliptical curves flexibility in negotiating ciphers is important.

Hence it is necessary to make available all ciphers that (i) are agreed upon in the federation context, (ii) are supported for the product and (iii) and are considered strong. There are 3 different applications of cryptography to be considered: XML Encryption[3], XML Signature and TLS. [SAML2MetaAlgSup] supports the negotiation of ciphers for XML Signature and Encryption. Entities MUST publish their cryptographic capabilities with regards to XML Signature and SHOULD publish them for XML Encryption.

TLS cipher negotiation is out of scope for SAML metadata.

6.2.4 Organization Information

Metadata provided by both Identity Providers and Service Provider SHOULD contain the participant's elements derived from ldap.gv.at. This information is to be used only for documentation.

6.2.5 Contact Information

Metadata provided by both Identity Providers and Service Provider SHOULD contain contact information for support and for a technical contact. The `<md:EntityDescriptor>` element SHOULD contain both a `<md:ContactPerson>` element with a `contactType` of "support" and a `<md:ContactPerson>` element with a contactType of "technical". The `<md:ContactPerson>` elements SHOULD contain at least one `<md:EmailAddress>`. The support address MAY be used for generic support questions about the service, while the technical contact may be contacted regarding technical interoperability problems. The technical contact MUST be responsible for the technical operation of the system(s) reflected in the metadata.

6.2.6 Registration and Publication Information

Registration Information is identical for all entities in Portalverbund. Therefore it SHOULD appear at the root element of a metadata document. It comprises of a RegistrationInfo element (from [MDAttribs]) with the attributes

- RegistrationAuthority and
- RegistrationPolicy.

Registration Information identifies location and version of a metadata document. It comprises the PublicationInfo with the attributes

- publisher (containing the URL where the document was retrieved),
- creationInstant (when this set of data was created or last modified),
- publicationId (a seuqnce number that is increased with each change in contents),
- UsagePolicy (a description of the intended usage for the metadata document).

[3] Currently not used in PVP

6.3 IDP Descriptor

Metadata documents provided by an Identity Provider MUST include an `<md:IDPSSODescriptor>` element containing all necessary `<md:KeyDescriptor>` and `<md:SingleSignOnService>` elements. The metadata SHOULD include one or more `<md:NameIDFormat>` elements indicating which `<saml2:NameID>` Format values are supported.

Each <md:IDPSSODescriptor> element SHOULD contain an errorURL XML attribute pointing to a page hosted by the Federation Participant that operates the IdP.

6.4 SP Descriptor

Metadata documents provided by a SP MUST include an `<md:SPSSODescriptor>` element containing all necessary `<md:KeyDescriptor>` and `<md:AssertionConsumerService>` elements.

The metadata SHOULD also include one or more `<md:NameIDFormat>` elements indicating which `<saml2:NameID>` Format values are supported and one or more `<md:AttributeConsumingService>` elements describing the service(s) offered and their attribute requirements.

The metadata provided by a SP SHOULD also contain a descriptive name of the service that the SP represents (not the company) in at least German (using the `xml:lang="de"` attribute). The name should be placed in the `<md:ServiceName>` in the `<md:AttributeConsumingService>` container.

If a Service Provider expects encrypted assertions, then its metadata MUST include a `<md:KeyDescriptor>` suitable for XML Encryption. Note that use of TLS/SSL is mandatory and XML encryption using CBC-ciphers is broken, so this practice is currently not recommended.

Required Attributes MAY be included to indicate attributes that are actually utilized by the SP. Utilization includes those attributes that are only logged but not otherwise processed to satisfy audit requirements.

6.4.1 Entity Categories

Metadata documents provided by a Service Provider MUST include at least one Entity Category in the `<md:SPSSODescriptor>` element specifying the requested attribute token. Multiple Entity Categories are additive. [draft-macedir-entity-attribute-00.xml]
Currently following values are supported:
http://www.ref.gv.at/ns/names/agiz/pvp/egovtoken (core set)
http://www.ref.gv.at/ns/names/agiz/pvp/egovtoken-charge (cost_center etc.)

6.4.2 Discovery Service

If a Service Provider needs to utilize a Discovery Service supporting the Identity Provider Discovery Service Protocol [IdPDisco], then its metadata MUST include one or more `<idpdisc:DiscoveryResponse>` elements in the `<md:Extensions>` element of its `<md:SPSSODescriptor>` element.

6.5 Signature and Expiration

Depending on the method of distribution either the XML-document containing the EntitiesDescriptor or in case of MDX each EntityDescriptor is signed using a certificate owned by the federation operator. In any case the ValidUntil attribute must be set to 24h into the future. The use of CacheDuration is discouraged.

6.6 Encoding of Special Characters

Predefined XML entity characters that are valid in URLs as well (which are the ampersand and apostrophe characters) SHOULD NOT be used in URIs contained in metadata to avoid encoding mistakes.

Rationale: XML uses entity encoding for special reserved characters, like '<' is encoded as '&lt;'. However, there are other encoding formats, such as URL encoding. XML entity encoding MUST be used within the XML document and never other forms of encoding, in particular in the endpoint URLs.

So the following is wrong because it uses URL encoding (note the %26):

```
<AssertionConsumerService>https://example.gv.at/?foo=value%26bar=value</AssertionConsumerService>
```

It should instead be (note the &amp;):

```
<AssertionConsumerService>https://example.gv.at/?foo=value&amp;bar=value</AssertionConsumerService>
```

6.7 Extensions

As the federation operator vouches for the metadata being published, metadata is restricted to the policy and procedures the entity registration. Hence, the only allowed data elements and attributes are those explicitly defined in this specification, or extensions agreed with the federation operator.

As a result, the Federation Operator SHOULD remove any element or attribute that is not explicitly defined in the metadata specification of a bilateral agreement with a metadata producer.

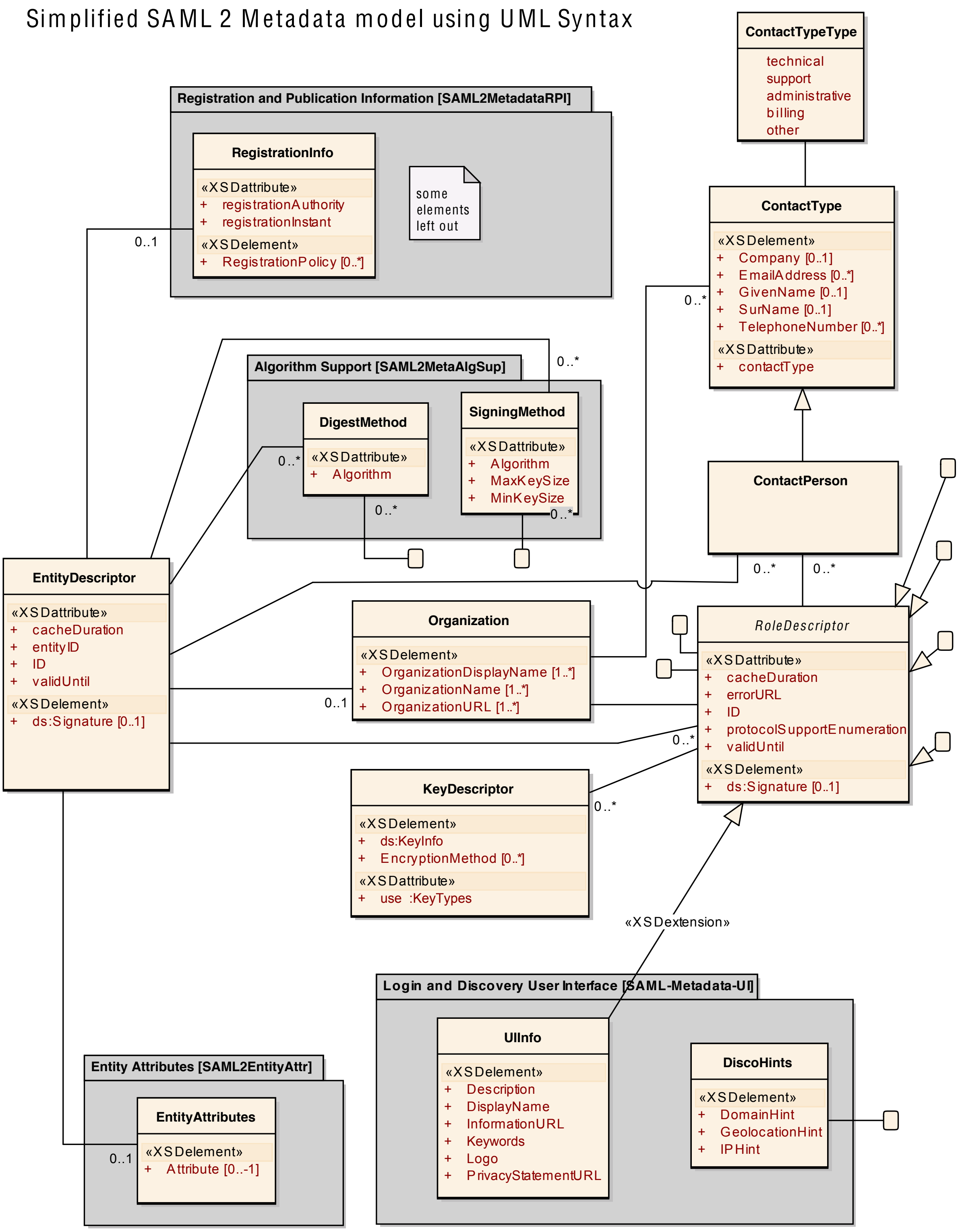

Fig. 4 Metadata Structure (1 of 2)

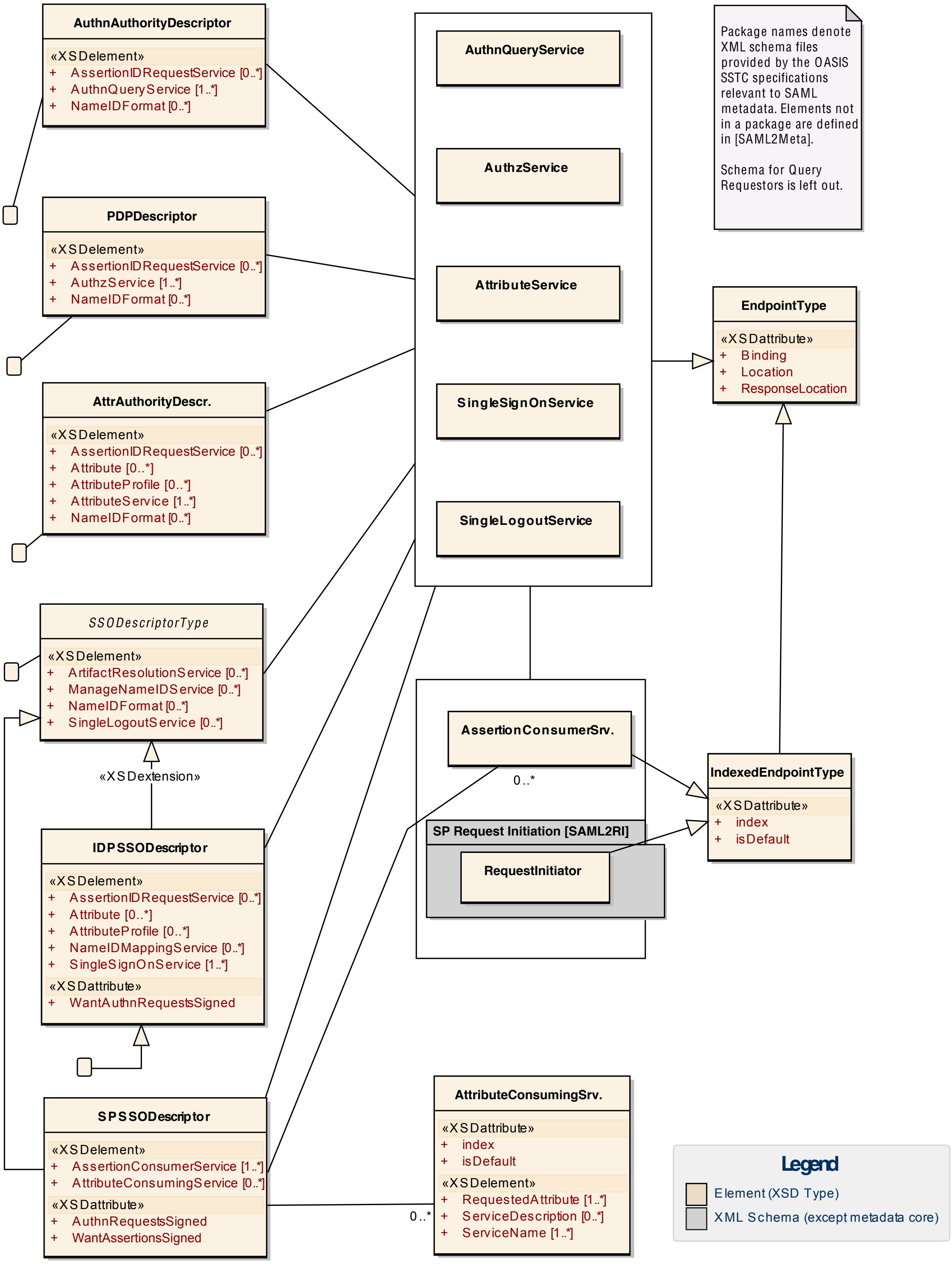
AuthnAuthorityDescriptor
«XSDelement»
+ AssertionIDRequestService [0..*]
+ AuthnQueryService [1..*]
+ NameIDFormat [0..*]
PDPDescriptor
«XSDelement»
+ AssertionIDRequestService [0..*]
+ AuthzService [1..*]
+ NameIDFormat [0..*]
AttrAuthorityDescr.
«XSDelement»
+ AssertionIDRequestService [0..*]
+ Attribute [0..*]
+ AttributeProfile [0..*]
+ AttributeService [1..*]
+ NameIDFormat [0..*]
SSODescriptorType
«XSDelement»
+ ArtifactResolutionService [0..*]
+ ManageNameIDService [0..*]
+ NameIDFormat [0..*]
+ SingleLogoutService [0..*]
«XSDextension»
IDPSSODescriptor
«XSDelement»
+ AssertionIDRequestService [0..*]
+ Attribute [0..*]
+ AttributeProfile [0..*]
+ NameIDMappingService [0..*]
+ SingleSignOnService [1..*]
«XSDattribute»
+ WantAuthnRequestsSigned
SPSSODescriptor
«XSDelement»
+ AssertionConsumerService [1..*]
+ AttributeConsumingService [0..*]
«XSDattribute»
+ AuthnRequestsSigned
+ WantAssertionsSigned
AuthnQueryService
AuthzService
AttributeService
SingleSignOnService
SingleLogoutService
Package names denote XML schema files provided by the OASIS SSTC specifications relevant to SAML metadata. Elements not in a package are defined in [SAML2Meta].
Schema for Query Requestors is left out.
EndpointType
«XSDattribute»
+ Binding
+ Location
+ ResponseLocation
AssertionConsumerSrv.
0..*
SP Request Initiation [SAML2RI]
RequestInitiator
IndexedEndpointType
«XSDattribute»
+ index
+ isDefault
AttributeConsumingSrv.
«XSDattribute»
+ index
+ isDefault
«XSDelement»
+ RequestedAttribute [1..*]
+ ServiceDescription [0..*]
+ ServiceName [1..*]
0..*
Legend
Element (XSD Type)
XML Schema (except metadata core)

Fig. 5 Metadata structure (2 of 2)

7 SAML V2.0 Metadata Specifications

This list provides an overview to the OASIS SSTC metadata specification documents used in this document.

7.1 SAML Metadata 2.0 [SAML2Meta]

Document title: Schema for SAML metadata V2.0, March, 2005
Location: http://docs.oasis-open.org/security/saml/v2.0/
Schema file: saml-schema-metadata-2.0.xsd

SAML profiles require agreements between system entities regarding identifiers, binding support and endpoints, certificates and keys, and so forth. A metadata specification is useful for describing this information in a standardized way. This specification defines an extensible metadata format for SAML system entities, organized by roles that reflect SAML profiles. Such roles include that of IdP and SP.

7.2 SAML Metadata Interoperability [SAML2MDIOP]

Document title: SAML V2.0 Metadata Interoperability Profile V1.0, August 2009.
Location: http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf

This profile is intended to improve and clarify the use of metadata to obtain interoperability in the areas of provisioning federated relationships between deployments, and establishing the validity of cryptographic signatures and handshakes. If an implementation can be shown to rely solely on the acceptance of metadata to derive trust, it can be reasoned about in a much simpler way, and the security exposures can be well understood.

7.3 Algorithm Support [SAML2MetaAlgSup]

Document title: Metadata Profile for Algorithm Support Version 1.0, June 2010
Location: http://docs.oasis-open.org/security/saml/Post2.0/
Schema file: sstc-saml-metadata- algsupport-v1.0.xsd

One of the interoperability challenges in large-scale, and long-term, SAML deployments is the selection of XML Signature [XMLSig] and XML Encryption [XMLEnc] algorithms at runtime when communicating with peer entities. In particular, accounting for software limitations that prevent support of newer algorithms, while supporting those algorithms where possible to gradually strengthen systems, is difficult to manage without knowledge of a peer's capabilities. This profile makes use of SAML metadata to enable deployments to document their algorithm capabilities and preferences. It also allows for future expansion to address the interoperability requirements of more complex algorithms.

7.4 IdP Discovery [IdpDisco]

Document title: Identity Provider Discovery Service Protocol and Profile V1.0, January 2007
Location: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery-cs-01.pdf
Schema file: sstc-saml-idp-discovery.xsd

All redirection-based SSO protocols share a common property in that the service provider is permitted to (and in most cases must) redirect the user agent to the identity provider. This creates opportunities for phishing attacks against the user's authentication credentials when weak (but extremely common) forms of authentication such as passwords are used. This protocol has the potential for creating additional opportunities for phishing if arbitrary web sites are permitted to utilize the protocol and obtain the user's identity provider, the key piece of knowledge required to fake the expected authentication experience. To mitigate this threat, metadata can be used to limit the sites authorized to use a discovery service, without introducing more complex (though stronger) approaches such as message authentication.

7.5 Entity Attributes [SAML2EntityAttr]

Document title: SAML V2.0 Metadata Extension for Entity Attributes V 1.0 (August 2009)
Location: http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr-cs-01.pdf
Schema file: sstc-metadata-attr.xsd

This profile defines a metadata extension element as a container for properties of entities. It allows attribute information to be carried within an entity's metadata to communicate additional information about that entity to a metadata consumer, much as an assertion can carry attributes about a subject.
A possible application of this mechanism is to allow a federation operator to include extensible information if entities adhere to optional federation policies.
This profiles defines no specific attributes to be communicated, but additional profiles might leverage it to do so.

7.6 Login and Discovery User Interface [SAML-Metadata-UI]

Document title: Metadata Extension Schema for SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0, 01 November 2010
Location: http://docs.oasis-open.org/security/saml/Post2.0/
Schema file: sstc-saml-metadata-ui- v1.0.xsd

This metadata extension profile includes several attributes associated with SAML entities that allow for a richer and user-friendly experience when selecting an IdP. Information includes such things as name, logo, privacy statement, geo-location.
As Entity Attributes can not be added specifically per role, but only per entity, this extension allows to add specific information specific to discovery services.

7.7 SP Request Initiation Protocol [SAML2RI]

Document title: Service Provider Request Initiation Protocol and Profile V1.0, March 2010
Location: http://www.oasis-open.org/committees/documents.php?wg_abbrev=security
Schema file: sstc-request-initiation.xsd

For documentation purposes, or as an aid in the dynamic construction of links in support of the Request Initiation Protocol, SP that are described using the SAML V2.0 Metadata specification MAY document endpoints supporting the protocol.

8 Relationship to ldap.gv.at

SAML Metadata provides basic access management:

- Asserting that entities are federation participants in good standing;
- Asserting entity categories that show the set of attributes an entity is entitled to receive.

LDAP.gv.at has an extended capability:

- List rights, right parameters and the minimum security class for application rights (machine readable) and document the legal basis for obtaining that right (human readable)
- List the set of rights and right parameters that a participant (a.k.a. home organization) may use.

The rights management data is *not* duplicated to SAML metadata. LDAP provides links to EntityDescriptors. Applications using PVP2/SAML need to access appropriate ldap entries (like gvApplicationRights) to execute rights management.

9 Operational Considerations

Federation metadata is the hinge point for the operation of a federation. Therefore the classical security objectives of confidentiality (protection of private keys), availability (of valid metadata and revocation information) and integrity (authority of metadata) must be warranted. Usually this is done with a service level agreement.

Given these parameters a data-center grade Hardware Security Module (HSM) to sign metadata should be considered. Compared to signing with smart cards it has the advantage of failover, recovery and more advanced business continuity features.

The freshness of metadata is assured by requiring at least daily updates. It should therefore not be necessary to send notification emails on changes. High-risk application should fetch metadata in short (e.g. 10-minutes) intervals. However, for emergency cases an operating procedure should be established to mail or call all contact persons.

10 References

In addition to the OASIS SSTC documents described in *7.* SAML V2.0 Metadata Specifications following references are noted:

[FeideMaRequ]	Metadata Aggregation Requirements Specification, Andreas Solberg, 2010-01-05. Downloaded from https://rnd.feide.no/2010/01/05/metadata_aggregation_requirements_specification

11 Abbreviations and Terms

TODO: Need to reference or define: AG-IZ, BLSG, HSM, Portalverbund, PVP, PVV, USP.
More general terms like TLS, IDP, SP, SAML are assumed to be known to the reader.