1

PVP Version 2 SAML Metadaten Management Spezifikation		Konvention PVP2-S-MD spec 2.1.3 Ergebnis der AG
Kurzbeschreibung	Um wohldefinierte und effiziente technische Schnittstellen für den Betrieb von Portalen bereitzustellen und Mechanismen für das Sicherheitsmanagement zu unterstützen, wird ein zentrales Repository von Metadaten mit definierten Management-Prozessen spezifiziert.	
Autor(en):	Rainer Hörbe (Wien)	Projektteam / Arbeitsgruppe **AG Integration und Zugänge (AG-IZ)** AG-Leitung: Ing. Dipl.-Ing.(FH) Hannes Wittmann, MSc (Mag. Wien) Stellvertretung: Dipl.-Ing. Dominik Klauser, BSc (BKA)
Beiträge von:	Bernd Zwattendorfer (EGIZ)	

3
4
5

Version	**2.1.3**	**:**	**20.12.2017**	Angenommen: -
Version	**2.1.2**	**:**	**1.6.2015**	Angenommen: -
Version	**2.1.1**	**:**	**26.2.2015**	Angenommen: -
Version	**2.1.**	**:**	**13.6.2014**	Abgelehnt: (muss auf Deutsch übersetzt werden)

6

7

PVP2-S Metadaten Profil

Inhalte

1 Zielgruppe, Auftrag und Zweck

Zielgruppe. Dieses Papier ist bestimmt für den Depositar sowie Teilnehmer der "Portalverbundvereinbarung" (PVV) und anderen Parteien, die eine PVP2-S Schnittstelle (SAML) zu dieser Föderation haben.

Auftrag. Diese Spezifikation ist das Ergebnis des Projekts "PVP2 zentrale Dienste" der AG-IZ Arbeitsgruppe. Diese ist eine Untergruppe der e-Gov Kooperation "BLSG", die Verwaltungen auf Bundes-, Landes- und Gemeindeebene mit anderen Körperschaften des öffentlichen Sektors in Österreich miteinander verbindet.

Zweck. Der Portalverbund ist über eine überschaubare Anzahl von Portalen hinausgewachsen. Um robuste und effiziente technische Schnittstellen bei der Einführung des PVP2-S-Profils für den Einsatz und um die Führung, das Risiko und die Steuerung wesentlicher Infrastruktur-Komponenten dieser e-Government Dienste zu unterstützen, sind ein zentrales Archiv für verbundwiete Metadaten zusammen mit definierten Management-Prozessen ausschlaggebend.

Danksagung. Dieses Papier basiert auf mehreren OASIS SSTC Standards. Weiterhin stützt es sich auf Listen, Wikis und Workshops der Shibboleth, TEREAN/REFEDS, Feide und IIW/EWTI-Gemeinschaften.

2 Allgemeiner Überblick

2.1 Begriffe

Die Begriffe sind im Abschnitt 11 definiert. Weil dieses Dokument auf der Spezifikation der SAML Metadaten aufbaut (siehe Kapitel 7), wird SAML-spezifische Terminologie verwendet. Zusätzlich zu den angegebenen Definitionen soll hier eine kurze Übersicht der Begriffszuordnungen gegeben werden:

SAML-Kontext	*Etablierte PVP Begriffe*
Entity Operator	Portalbetreiber
Föderation	Portalverbundsystem
(System) Entity[1]	Portal
Federation Operator (FO)	Funktion des Depositars
Identity Provider (IDP)	Stammportal
Service Provider (SP)	Anwendungsportal für eine Anwendung

2.2 Grundsätzlicher Bedarf und Ziele

Metadaten werden verwendet, um die Konfiguration von SAML System Entities (im Wesentlichen IdP und SP Rollen) auf vertrauenswürdige Art und Weise zu kommunizieren. Ein Dienst von zentraler Bedeutung in Federations ist die Aggregation

[1] Gebräuchliche Verwendung. Formal ist Entity ein weitergehender Begriff.

von Metadaten der Teilnehmer. Die Aggregation wird erreicht, indem eine Liste der Metadaten der teilnehmenden Entities in einem maschinenlesbaren Format, welches der Policy der Föderation entspricht, vorgelegt wird (siehe 11. Abkürzungen und Begriffe). Die Liste wird zentral vom Federation Operator (FO) verwaltet und gepflegt.

Metadaten unterliegen einer Policy, welche die Verantwortlichkeiten und Einschränkungen der Organisationen festlegt, die die Metadaten erstellen und aggregieren. Die Aggregation hilft dabei, das Deployment von Diensten zu vereinfachen, indem sie eine einzige Stelle für die Übernahme der Metadaten von vielen Entities bereitstellt und indem der Federation Operator als Vermittler technischen Vertrauens agiert.

2.3 Akteure

Die grundlegenden Akteure in einer Föderation sind der Federation Operator (FO), der Identity Provider (IdP) und der Service Provider (SP). Für eine spezifischere Gestaltung führt [FeideMaRequ] [2] zusätzliche Begriffe für die Managementbetrachtung der Metadaten ein, indem er zwischen den Komponenten und ihren Rollen unterscheidet. Die (organisatorischen) Rollen sind:

- *Metadata Publishers* (MP)
- *Metadata Consumer* (MC)
- *Entity Operator* (EO - entspricht dem Portalbetreiber)

Die (technischen) Komponenten sind:

- *Identity Provider* (IdP)
- *Service Provider* (SP)
- *Metadata Aggregator* (MA)
- (Federation) *Metadata Registrar* (MR – das ist eine Funktion des Depositars)

Ein Metadaten Aggregator (MA) sammelt Metadaten von einem oder mehreren Metadata Publishern (MP) und veröffentlicht die gesammelten Metadaten, welche für einen oder mehrere Metadata Consumer (MC) auf Basis der Konfiguration und Regeln der Föderation validiert, gefiltert und aggregiert wurden.

In seiner augenblicklichen Version richtet sich das SAML Metadaten Profil dieser Spezifikation an eine flach strukturierte Föderation, in welcher der Federation Operator als MR und MA agiert und die System Entities die Rollen von sowohl MP und MC spielen, wie in Abbildung 1 dargestellt. Für andere Konstellationen, einschließlich Interfederation, siehe [FeideMaRequ].

[2] [FeideMaRequ] unterscheidet nicht Entities von ihren Betreibern. Aus Gründen der Konsistenz führen wir den Begriff Entity Operator ein.

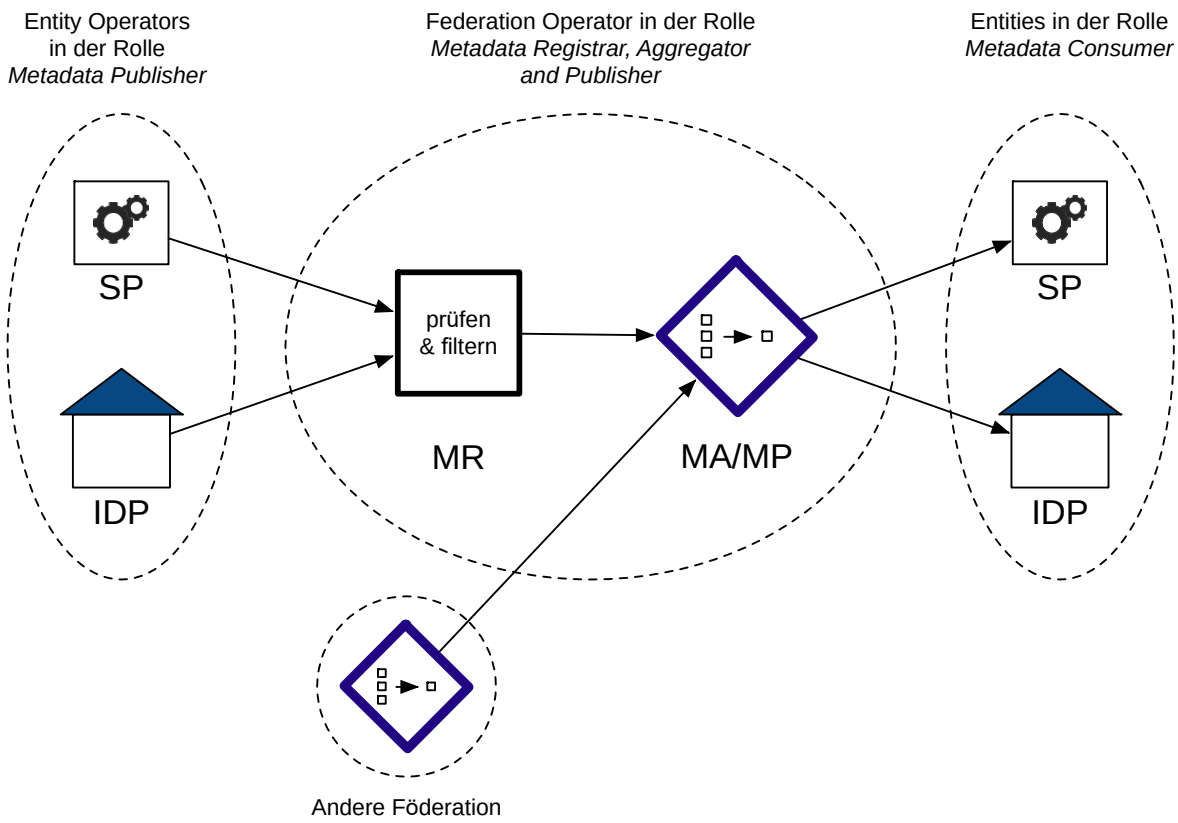


Abbildung 1: Fluss der Metadaten in einer Föderation.

2.4 Interfederation

Interfederation wird in diesem Dokument nicht explizit betrachtet. Dennoch ist Interfederation bei Verwendung der vorgeschlagenen Architektur immer noch möglich. Um Interfederation umzusetzen, könnten Entities (i) an mehreren Federations teilnehmen (auch als Multi-Homing bekannt) [3] oder (ii) ein Federation Operator akzeptiert den aggregierten Metadaten Feed einer anderen Föderation (eine Art Roaming-Vereinbarung) oder (iii) Entities treten einer Föderation aufgrund gesetzlicher Bestimmungen bei. Aus Sicht dieses Papiers werden diese Fälle, d. h. einschließlich System Entities anderer Federations gleich behandelt, falls Entities direkte Föderations-Teilnehmer sind.

Beispiele solcher Konstellationen sind (i) ein kommerziell betriebener Service, der Dienste an PVV und USP Anwender mit Einzelverträgen anbietet (ii) ein von der Verwaltung betriebener Service, der Dienste an PVV und USP Anwender auf Basis einer gegenseitigen Vereinbarung, die USP und alle PVV Mitglieder regelt, bietet und (iii) ein Bürgerkarten-IDP, der Bürger gegenüber PVV-Anwendungen authentifiziert.

[3] Während Multi-Homing die einfachste und offensichtlichste Möglichkeit für eine Entity ist an mehreren Föderationen teilzunehmen, kann es nicht triviale Auswirkungen auf den Service und den IdP Discovery Service haben, die noch nicht analysiert wurden.

3 Metadaten - Struktureller Überblick

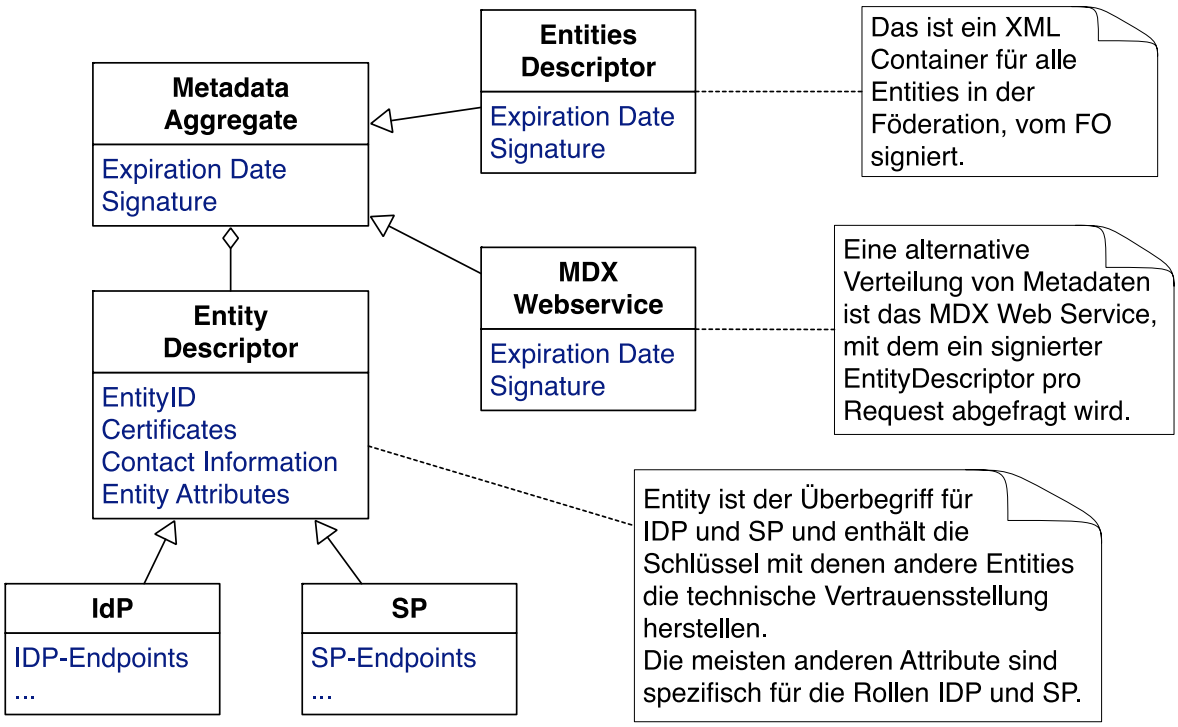


Abbildung 2: Metadaten Struktur

Abbildung 2 gibt einen Überblick über die Struktur der beteiligten Komponenten für das Metadaten-Management. Metadaten können in zwei Hauptgruppen eingeteilt werden. Die erste Gruppe ist das Element *EntityDescriptor*, welches die Entity Metadaten Informationen eines *IdP* oder *SP* beschreibt. Die zweite Gruppe ist das *Metadaten Aggregat,* welches die individuellen *EntityDescriptor* Elemente eines *IdP* oder *SP* sammelt und diese *EntityDescriptor* Elemente für die Föderation verfügbar macht. Dies kann auf zwei Arten erfolgen. Die erste Möglichkeit besteht darin, die einzelnen *EntityDescriptor* Elemente zu aggregieren und eine Liste aller *EntityDescriptor* Elemente in einem speziellen Container (*EntitiesDescriptor*) zu veröffentlichen. Die zweite Möglichkeit ist der *MDX Webservice*, der einen *EntityDescriptor* auf Anforderung verteilt. Das *Metadaten Aggregat*, der *EntitiesDescriptor* und der *MDX Webservice* werden durch den Federation Operator betrieben und verwaltet. Bei einer Aggregation wird der *EntitiesDescriptor* durch den FO signiert, bei einer Verteilung durch den *MDX Webservice* wird der angeforderte *EntityDescriptor* von dem FO signiert. Im weiteren Verlauf des Papiers bezeichnen wir Metadaten, die durch den FO veröffentlicht wurden, als *FO Metadaten*.

4 Anforderungen für Metadaten- und Vertrauensmanagement

4.1 Grunderfordernisse

Die folgenden Grunderfordernisse eines Federation Operators müssen bei der Definition des Metadaten-Managements berücksichtigt werden:

- Anzeige der Entities (Portale) rechtmäßiger Teilnehmer (Entities können vom Teilnehmer selbst oder von einem beauftragten Entity Operator betrieben werden).
- Verwaltung der Metadaten jeder System Entity, einschließlich:

- Service Endpunkte,
- Gültige Schlüssel (Zertifikate);
- Unterstützte Cipher Suites zur flexiblen Wahl des Algorithmen Supports; Die Fähigkeiten von System Entities MÜSSEN beschrieben sein, sie brauchen aber nicht automatisch[4] ausgehandelt werden;
- Erforderliche Attribute des SPs (EntityCategory oder <md:RequestedAttribute> siehe 5.6.1);
- Andere Eigenschaften.

- Verknüpfung von Entities mit PV-Teilnehmern, die eigene Entities betreiben und besitzen.
- Durchsetzung der Federation Policy, indem sichergestellt wird, dass die Metadaten der Entity authentisch sind.
- Aggregation und Verteilung der Metadaten der System Entities in einer sicheren und zeitnahen Art und Weise.
- Organisatorische und technische Maßnahmen müssen mit der Federation Policy übereinstimmen, etwa für Protokollierung, Schlüssel-Management, Widerruf und Aufrechterhaltung des Geschäftsbetriebs (Business Continuity).

4.2 Vertrauen auf Business- und Technik-Ebene

Der Federation Operator muss in Übereinstimmung mit der Federation Policy arbeiten, bspw. bei der Aufnahme und Kündigung von Teilnehmern sowie Veröffentlichung und Aggregation der gesammelten technischen Konfigurations-Metadaten ihrer System Entities. Das verwendete Format, um diese FO Metadaten zu veröffentlichen, entspricht [SAML2Meta]. Die FO Metadaten sind eine Sammlung von *EntityDescriptor*[5] Elementen, die durch Teilnehmer erzeugt und verwaltet werden und durch den Federation Operator authentifiziert werden. Die Kette der Vertrauensstellungen wird entsprechend dem nachstehenden Prozess eingerichtet:

1. Die Organisation beantragt die Teilnahme an der Föderation;
2. Der Federation Operator (FO) prüft und bestätigt den Antrag;
3. Die Organisation wird somit Teilnehmer der Föderation. Eine sehr sichere Art der gegenseitigen Authentifikation wird etabliert, wie in der Erklärung zum Metadata Key Management Practice Statement niedergelegt, (siehe 11 Abkürzungen und Begriffe);
4. Der FO teilt das Zertifikat der *Metadaten Aggregat* Signatur mit dem Teilnehmer;
5. Der Teilnehmer übermittelt einen *EntityDescriptor* an den FO;
6. Der FO prüft, ob der *EntityDescriptor*
   a. die korrekte Syntax aufweist;
   b. Authentisch ist (gebunden an den Teilnehmer);
   c. Über Attribute verfügt, die bezogen (SP) oder geliefert werden können (IdP), spezifiziert entsprechend Abschnitt 5.6.1;
   d. Keine Elemente oder Merkmale enthält, die nicht im Metadaten Profil enthalten sind (d. h. nicht vereinbarte, individuelle Erweiterungen)[6];
7. Der FO fügt den *EntityDescriptor* dem *Metadaten Aggregat* hinzu, verlinkt ihn mit *ldap.gv.at* und fügt Registrierungs- und Veröffentlichungs-Merkmale sowie eine Signatur hinzu;

[4] Es reicht daher aus, wenn der Entity Operator die Fähigkeit manuell in den Metadaten dokumentiert.
[5] Das EntityDescriptor Element spezifiziert Metadaten für eine einzelne Entity. Eine einzelne Entity kann in verschiedenen Rollen agieren, um mehrere Profile zu unterstützen. [SAML2Meta]
[6] Begründung: der FO sollte keine unbekannten Datenelemente signieren.

8. Jeder Teilnehmer kann den *EntityDescriptor* entweder über den *EntitiesDescriptor* oder über *MDX Webservice* abrufen und die Signatur validieren.

4.3 Use Cases

Use Cases für SAML Metadaten sind im [SAML-metadata-guide] beschrieben.

5 Metadaten Format, Inhalte und Regeln

Identity Provider und Service Provider MÜSSEN ein SAML 2.0 Metadaten-Dokument vorlegen, welches ihre Entity beschreibt. Die vorgelegten Metadaten MÜSSEN dem SAML V2.0 Metadata Interoperability Profile Version 1.0 [SAML MetaIOP] entsprechen.

5.1 FO Metadaten-Signatur

Das Trust Root (die oberste Instanz in der kryptographischen Vertrauensstellung) wird eingerichtet und entsprechend den nachstehenden Regeln gepflegt:

- Ein MA (Metadaten Aggregator) MUSS über einen Signatur-Schlüssel in Form eines X.509 Zertifikats verfügen. Der MA muss den dazugehörigen privaten Schlüssel erstellen und schützen, indem er ein HSM verwendet.
- Der MA MUSS das Zertifikat des signierenden Schlüssels mittels einer Schlüsselzeremonie oder eines äquivalenten Verfahrens, die auf einer schriftlichen Policy basiert, out-of-band[7] teilen. Das Verfahren MUSS Möglichkeiten beinhalten, die es den Metadata Consumers (MC) erlauben, den signierenden Schlüssel anhand seines Fingerprints zu verifizieren.
- MCs MÜSSEN den öffentlichen Schlüssel einer Whitelist hinzufügen, wenn sie eine Metadaten Signatur validieren.
- Metadata Consumer SOLLTEN NICHT die üblichen Validierungsverfahren (Pfad-Validierung und Widerruf) verwenden, da diese auf die (schwachen) handelsüblichen PKIX Verfahren zurückgreifen.
- Der MA MUSS über ein Out-of-Band-Verfahren zum Widerruf eines kompromittierten Metadaten Signatur-Schlüssels verfügen. Das Verfahren ist in der Federation Policy beschrieben.

5.2 Gültigkeit

FO Metadaten Wurzelelemente (*EntitiesDescriptor* bei einer Aggregation oder *EntityDescriptor* bei MDX) MÜSSEN ein validUntil Attribut enthalten. Metadata Consumer DÜRFEN FO Metadaten nach Ablauf NICHT mehr verwenden. Die *cacheDuration* kann mittels einer föderationsweiten Policy festgelegt werden oder KANN im *cacheDuration* Merkmal in *EntitiesDescriptor* oder *EntityDescriptor* beinhaltet sein.

5.3 Gemeinsame Elemente für Entities (bei IdP und SP)

5.3.1 Entity ID

Der Wert des `EntityDescriptor@entityID` Attributs SOLLTE die canonical URL des Metadaten Dokuments der Entity sein. Canonical URLs folgen der semantik-erhaltenden Normalisierung, wie in RFC 3986 Section 6, bspw.
https://testsp.xyz.tld/sp.xml **aber nicht** *https://testsp.xyz.tld:443/sp.xml* beschrieben.[8]

[7] Out-of-band: Wenn die in-band Kommunikation erfolgreich angegriffen wird, darf der out-of-band Kanal dadurch nicht korrumpiert werden.
[8] Ein häufiges Missverständnis ist, dass die *entityID* derDomäne Endpoint-URLs des *EntityDescriptor* entsprechen muss. Anders als der Endpoint-locations spiegelt die *entityID* die Organisation, der die Entity gehört, genau und eindeutig wieder.

5.3.2 Schlüssel-Material für IdP und SP

Eine SAML Entity verwendet asymmetrische Kryptographie, um die Daten zu schützen, die an vertrauenswürdige Partner übermittelt werden. Öffentliche Schlüssel werden in Form eines X.509 Zertifikats in Entity Metadaten veröffentlicht, während der entsprechende private Schlüssel von der Entity geheim und sicher verwahrt wird. Diese Schlüssel werden zum Verschlüsseln und Signieren auf der Nachrichtenebene genutzt sowie dazu, um gesicherte Rückkanäle zum Transport von SAML Nachrichten über SSL/TLS zu erzeugen. Sie werden **nicht** für browserseitige SSL/TLS Transaktionen auf Port 443 verwendet. Schlüssel müssen als `<ds:X509Certificate>` (Unterelemente von `<ds:X509Data>`) enthalten sein.

5.3.2.1 Vertrauens-Modelle

Das grundlegende Vertrauens-Modell (wie in [SAMLMetaIOP] beschrieben) validiert Zertifikate, indem es deren öffentlichen Schlüssel mit Metadaten vergleicht, und so ein White-Listing implementiert. Die Metadaten Signatur bietet den Basisschlüssel (Trust Anchor) für diese Schlüssel in den Metadaten.
Das PKIX Vertrauens-Modell SOLLTE als Ausweichmöglichkeit unterstützt werden, um Kompatibilität mit Produkten zu gewährleisten, die das erste Modell nicht unterstützen. Das erste Modell gilt als überlegen, da es White-Listing verwendet. Zertifikate, die unter Verwendung von SAML Metadaten verteilt werden, SOLLTEN dem PKIX Modell entsprechen.

5.3.2.2 X.509 Entity Zertifikate in Federation Metadaten

Dieses Profil bestimmt die nachstehenden Sicherheits- und Vertrauensanforderungen zu Entity Zertifikaten, die in Federation Metadaten enthalten sind:

- Die Verwendung von langlebigen Zertifikaten, mit einer Lebensdauer von 4 Jahren oder mehr in Federation Metadaten wird EMPFOHLEN, um unnötige, durch die Technik aufgezwungene Fristen für Schlüsselübergänge zu vermeiden.
- Wenn ein Schlüssel zur Verschlüsselung enthalten ist, KANN sein *KeyDescriptor* eine oder mehrere *EncryptionMethod* Elemente enthalten.
- Schlüssel und Zertifikate müssen dem PVP2-S-Profil V2.1 Abschnitt 3.1.1 entsprechen.
- Die Entscheidung, einen neuen privaten Schlüssel zu erzeugen und ein Zertifikat für einen neuen öffentlichen Schlüssel auszugeben, hängt von der Federation Policy des Teilnehmers oder von der Notwendigkeit in Fällen, in denen ein kompromittierter Schlüssel vermutet wird, ab.
- Abgelaufene Zertifikate DÜRFEN NICHT in die Federation Metadaten aufgenommen werden und MÜSSEN entfernt werden, sobald der Prozess der Migration auf ein neues Zertifikat abgeschlossen ist.
- Zum Zweck des Schlüssel-Managements sind zu jeder Zeit mehrere Zertifikate pro *RoleDescriptor* zulässig. Produkte, die dies nicht zulassen, MÜSSEN Metadaten in geeigneter Weise vorverarbeiten, um sie interoperabel zu machen.
- Der Federation Operator MUSS die Subject-Attribute in Zertifikaten validieren, da diese Daten im R-Profil verwendet werden. Das Subject-Attribut MUSS den DN des Betreibers des IDP/SP/Portals enthalten.

5.3.2.3 Schlüsselverwendung

Die Verwendung verschiedener Schlüssel zum Signieren und für TLS (usage="signing") einerseits und zur Verschlüsselung (usage="encrypt") andererseits wird empfohlen. In

jedem Fall MUSS das *usage* Attribut vorhanden sein. Daher MÜSSEN Schlüssel zum Signieren und für TLS durch das *usage*="signing" Attribut gekennzeichnet sein. Falls Verschlüsselung unterstützt wird, MÜSSEN Schlüssel zur Verschlüsselung durch das Attribut *usage*="encrypt" gekennzeichnet sein.

5.3.2.4 Schlüsselwerte

Für eine gut lesbare Dokumentation SOLLTE eine einfache Textversion des Zertifikats als XML Kommentar, mit einer Zertifikatzeichenfolge enthalten sein, bspw.:

```
<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>
```

5.3.3 Unterstützung von Algorithmen

Theoretisch sollten schwache Algorithmen innerhalb eines angemessenen Zeitraums ersetzt werden. Die Praxis hat jedoch gezeigt, dass nicht nur verdächtige, sondern auch gebrochene Algorithmen viele Jahre in Produktiv-Systemen verbleiben. Flexibilität im Aushandeln von Algorithmen ist auch wichtig, weil der Austausch von SHA1 und CBC überfällig (Stand 2013) ist, aber auch wegen des absehbaren Wechsels zu elliptischen Kurven. Dies erlaubt den Upgrade der Kommunikation zwischen Teilnehmern, die neuere Algorithmen unterstützen, während gleichzeitig Produkte unterstützt werden, die auf eine einzige historisch gewachsene Einstellung begrenzt sind.
Daher ist es erforderlich, dass jede Entity ihre Fähigkeit Algorithmen, die (i) im Kontext der Föderation vereinbart sind, (ii) für das Produkt unterstützt werden und (iii) als stark angesehen werden, veröffentlicht.
Es sind 3 verschiedenen Anwendungsfälle der Kryptographie zu berücksichtigen: XML Verschlüsselung, XML Signatur und TLS. [SAML2MetaAlgSup] unterstützt die Aushandlung von Chiffren für XML Signaturen und zur Verschlüsselung. Entities MÜSSEN ihre kryptographischen Fähigkeiten in Bezug auf XML Signaturen in ihren Metadaten veröffentlichen. Für die XML Verschlüsselung SOLLTEN sie diese in ihren Metadaten veröffentlichen. Entities DÜRFEN diese Informationen für die automatisierte Aushandlung von Algorithmen verwenden. Siehe in Anhang 1 Unterstützung von Algorithmen
Die Aushandlung von TLS Algorithmen liegt außerhalb des Geltungsbereichs von SAML Metadaten.

5.3.4 Informationen zur Organisation

Metadaten, die von Identity Providern oder Service Providern zur vorgelegt werden, sind mit *ldap.gv.at* über das *entityID* Attribut verlinkt. Die Organisation ist entweder der PV-Teilnehmer, der die System Entity kontrolliert oder der Betreiber, der die System Entity betreibt. Diese Information dient nur Dokumentationszwecken.

5.3.5 Kontakt

Metadaten, die von Identity Providern oder Service Providern zur Verfügung gestellt werden, SOLLTEN Kontaktdaten für den Support und für einen technischen Kontakt enthalten. Das `<md:EntityDescriptor>` SOLLTE beide Elemente enthalten, ein `<md:ContactPerson>` Element mit contactType "`Support`" und ein `<md:ContactPerson>` Element mit contactType "`technical `". Die `<md:ContactPerson>` Elemente SOLLTEN mindestens ein `<md:EmailAddress>` enthalten. Die Support Adresse KANN für allgemeinen Support zu Fragen über den Service genutzt werden, während der technische Kontakt für Fragen zu Problemen der Interoperabilität kontaktiert werden kann. Der technische Kontakt MUSS für den technischen Betrieb des Systems/der Systeme, die in den Metadaten aufgeführt sind, verantwortlich sein.

5.3.6 Registration and Publication Information

Registrierungsdaten SOLLTEN durch den Metadata Registrar für jeden *EntityDescriptor* in den FO Metadaten eingefügt werden. Die Daten bestehen aus einem *RegistrationInfo* Element (aus [MDAttribs]) mit den Attributen:

- *registrationInstant,*
- *RegistrationAuthority* und
- *RegistrationPolicy*.

Die Publication Information identifiziert den Standort und die Version eines Metadaten Dokuments. Sie enthält das *PublicationInfo* Element mit den Attributen

- publisher (enthält die URL, von welcher das Dokument abgerufen wurde),
- *creationInstant* (Zeitstempel, wann dieser Satz an Daten erstellt oder das letzte Mal geändert wurde),
- *publicationId* (eine laufende Nummer, die mit jeder Änderung in den Inhalten erhöht wird),
- *UsagePolicy* (eine Beschreibung über die beabsichtigte Verwendung des Metadaten Dokuments).

5.4 Benutzeroberfläche für das IdP Discovery

Falls IdP Discovery unterstützt werden muss, dann MUSS jede Entity ihren von Menschen lesbaren Namen, Beschreibung und Logo URL, wie in Abschnitt 7.6 beschrieben enthalten. Dies dient der Verbesserung der Bedienbarkeit im Metadaten Management, und um das metadaten-basierende IdP Discovery und Service Discovery für IDP-first Flows zu unterstützen. Die Inhalte des `<mdui:UIInfo>` Elements für SP sollten diesem Beispiel folgen:

```
<mdui:DisplayName xml:lang="de">ZMR SecClass2</mdui:DisplayName>
<mdui:Description xml:lang="de">
      Zentrales Melderegister (SecClass 2)
</mdui:Description>
<mdui:Logo xml:lang="de" height="70" width="79">
      https://awp.bmi.gv.at/xyz/zmrlogo.png
</mdui:Logo>
```

`<mdui:DisplayName>` und `<mdui:Description>` SOLLTEN geeigente Werte enthalten, die einen Endanwender dabei unterstützen, eine Anwendung aus einer Liste auszuwählen. Der Name der Organisation sollte nur im `<mdui:DisplayName>` und `<mdui:Description>` enthalten sein, wenn er zur Orientierung erforderlich ist. `<mdui:DisplayName>` und `<mdui:Description>`

SOLLTEN keine URL enthalten. Das *Logo* SOLLTE quadratisch und klein sein, um eine kompakte Aufstellung vieler Services zu ermöglichen. Für transparente (maskierte) Bilder im .png Format kann ein heller Hintergrund vorausgesetzt werden.
<mdui:DisplayName> und <mdui:Description> sollten Anwendern dabei helfen, ihre Stammorganisation zu identifizieren. Die folgende Tabelle zeigt die Inhalte dieser Elemente in typischen Szenarien:

	Szenario	Inhalte von Name und Beschreibung
1	Die Stammorganisation verfügt über einen einzelnen IDP	*Name der Organisation* Identity Provider
2	Die Stammorganisation verfügt über mehrere IDPs	*Name der Organisation* Identity Provider für die *Benutzergruppe*
3	Die Stammorganisation wird von einem externen IDP gehostet	*IDP Provider* Identity Provider

(Siehe auch das Beispiel in Anhang 1e)

Im 3. Szenario SOLLTE das optionale Element <mdui:Keywords> eine vollständige Liste der Stammorganisationen enthalten, die zum Suchen verwendet werden kann. Nur die Suchbegriffe, welche diese Organisation von anderen unterscheidbar macht, SOLLTEN zur Verfügung gestellt werden. Allgemeine Stichwörter wie "Gemeinde", "Land" "Bundesministerium" etc. DÜRFEN NICHT angegeben werden. <mdui:Keywords> DARF NICHT den gleichen Text enthalten wie <mdui:DisplayName>. Gute Kandidaten für Stichwörter sind abgekürzte und vollständige Namen.

5.5 IDP Descriptor

Metadaten Dokumente, die durch einen Identity Provider vorgelegt werden, MÜSSEN ein <md:IDPSSODescriptor> Element beinhalten, welches alle erforderlichen <md:KeyDescriptor> (siehe Sektion 5.3.2.3 "Schlüsselverwendung" für weitere Details) und <md:SingleSignOnService> Elemente enthält. Ein <md:KeyDescriptor> zum Signieren MUSS immer enthalten sein. Die Entity Metadaten SOLLTEN eines oder mehr der <md:NameIDFormat> Elemente enthalten, um anzugeben, welche <saml2:NameID> Format-Werte unterstützt werden.
Jedes <md:IDPSSODescriptor> Element SOLLTE ein *errorURL* XML Attribut enthalten, welches auf eine Webseite verweist, die von dem Teilnehmer gehostet wird, der den IdP betreibt. Diese Seite SOLL Kontaktinformationen des Helpdesks beinhalten, welches für den IdP zuständig ist und SOLL Orientierung zu Themen, wie bspw. ein Anwender/eine Anwenderin seine/ihre Stammorganisation findet, bieten.

5.5.1 Entity Categories

Metadaten Dokumente, die von einem Identity Provider bereitsgestellt werden, KÖNNEN eine *Entity Category* enthalten, um anzuzeigen, welche Attributsätze vom IdP zur Verfügung gestellt werden können. (Siehe 5.6.1)

5.6 SP Descriptor

Metadaten Dokumente, die durch einen SP vorgelegt werden, MÜSSEN ein <md:SPSSODescriptor> Element beinhalten, welches alle notwendigen <md:KeyDescriptor> (für Einzelheiten siehe Sektion 5.3.2.3 "Schlüsselverwendung") und <md:AssertionConsumerService> Elemente enthält. Ein <md:KeyDescriptor> zum Signieren MUSS immer enthalten sein.

Die Metadaten SOLLTEN auch ein oder mehrere <md:NameIDFormat> Elemente enthalten, die anzeigen, welche <saml2:NameID> Formatwerte unterstützt werden und weiterhin ein oder mehrere <md:AttributeConsumingService> Elemente, die den oder die angebotenen Dienste und ihre Attributanforderungen beschreiben.

Die Metadaten, die von einem SP vorgelegt werden, SOLLTEN auch, (unter Verwendung des xml:lang="de" Attributs) - zumindest in deutscher Sprache - einen beschreibenden Namen des Service enthalten, den der SP verkörpert (nicht das Unternehmen). Der Name sollte in <md:ServiceName> im <md:AttributeConsumingService> Container stehen.
Falls ein Service Provider verschlüsselte Assertions erwartet, dann MÜSSEN seine Metadaten einen <md:KeyDescriptor> enthalten, der für die XML Verschlüsselung geeignet ist. Dies MUSS zusätzlich durch *usage*="encryption" kenntlich gemacht werden.

Erforderliche Attribute oder EntitiyCategories können enthalten sein, um anzuzeigen, welche Attribute vom SP tatsächlich genutzt werden. (Siehe die nächste Sektion 5.6.1 bzgl. der Verwendung)

5.6.1 Entity Categories

Metadaten Dokumente, die von einem Service Provider vorgelegt werden, DÜRFEN eine oder mehr Entity Categories im <md:SPSSODescriptor> Element beinhalten, welches einen Satz verlangter Attribute spezifiziert. [draft-macedir-entity-attribute-00.xml].
Metadaten MÜSSEN entweder <md:RequestedAttribute> Elemente oder eine EntityCategory enthalten.
Zur Spezifikation von geforderten Attributen für einen SP sind folgende Optionen zulässig:
a) EntityCategory für eine definierte Menge von Attributen

Akzeptierte Werte für Entity Categories sind in [PVP2-Attr] definiert, bspw.
http://www.ref.gv.at/ns/names/agiz/pvp/egovtoken (core set)
Die Methode MUSS verwendet werden, um einen PVP eGovToken anzufordern. Sie wird empfohlen, wenn die Attributmenge über mehrere Anwendungen hinweg wiederverwendet wird. Mehrfache Entity Categories sind additiv. Mehrfache Entity Categories MÜSSEN als mehrere <saml:AttributeValue> Elemente in einem einzigen <saml:AttributeName> Element dargestellt werden.
Falls von einem SP keine Attribute verlangt werden, MUSS die folgende EntityCategory spezifiziert werden:
http://www.ref.gv.at/ns/names/agiz/pvp/no-attributes

b) *RequestedAttribute* in Metadaten

Eine Liste von <md:RequestedAttribute> Elementen als untergeordnete Elemente eines <AttributeConsumingService> wird vom SP in den Metadaten vorgelegt. Dies ist empfohlen, wenn eine kleine Zahl von Attributen für einen speziellen Service verwendet wird.

c) *RequestedAttribute* im Authentication Request

Attribute, die selten verwendet werden, können als Erweiterungen im Authentication Request verlangt werden.

5.6.2 Discovery Service

Wenn ein Service Provider einen Discovery Service verwenden muss, der das Identity Provider Discovery Service Protokoll [IdPDisco] unterstützt, dann MÜSSEN seine

Metadaten eines oder mehr <idpdisc:DiscoveryResponse> Elemente im <md:Extensions> Element seines <md:SPSSODescriptor> Elements beinhalten.

5.6.3 Request Initiation Protocol Endpunkt

SPs MÜSSEN die Metadatenerweiterung für das SP Request Initiation Protocol [SAML2RI] beinhalten, falls IDP initiierte Logins unterstützt werden müssen.

5.7 Signatur und Ablauf

Abhängig von der Methode der Metadaten-Verteilung wird bei einer Aggregation der *EntitiesDescriptor* oder bei MDX jeder *EntityDescriptor* unter Verwendung eines Zertifikats vom Federation Operator signiert. In jedem Fall muss das *validUntil* Attribute auf 10 Tage in die Zukunft gesetzt werden.

5.8 Kodierung von Sonderzeichen

Vordefinierte XML Entity Zeichen, die auch in URLs gültig sind (das kaufmännische Und und das Apostroph), SOLLTEN NICHT in URIs, die in Metadaten enthalten sind, verwendet werden, um Kodierungsfehler zu vermeiden.

Begründung: XML verwendet Entity-Kodierung für reservierte Sonderzeichen, bspw. wird '<' als '<' kodiert. Es gibt jedoch andere Kodierungsformate, wie die URL Kodierung. Innerhalb von XML Dokumenten MUSS die XML Entity-Kodierung verwendet werden und niemals andere Arten der Kodierung, insbesondere nicht in der Endpunkt URL.

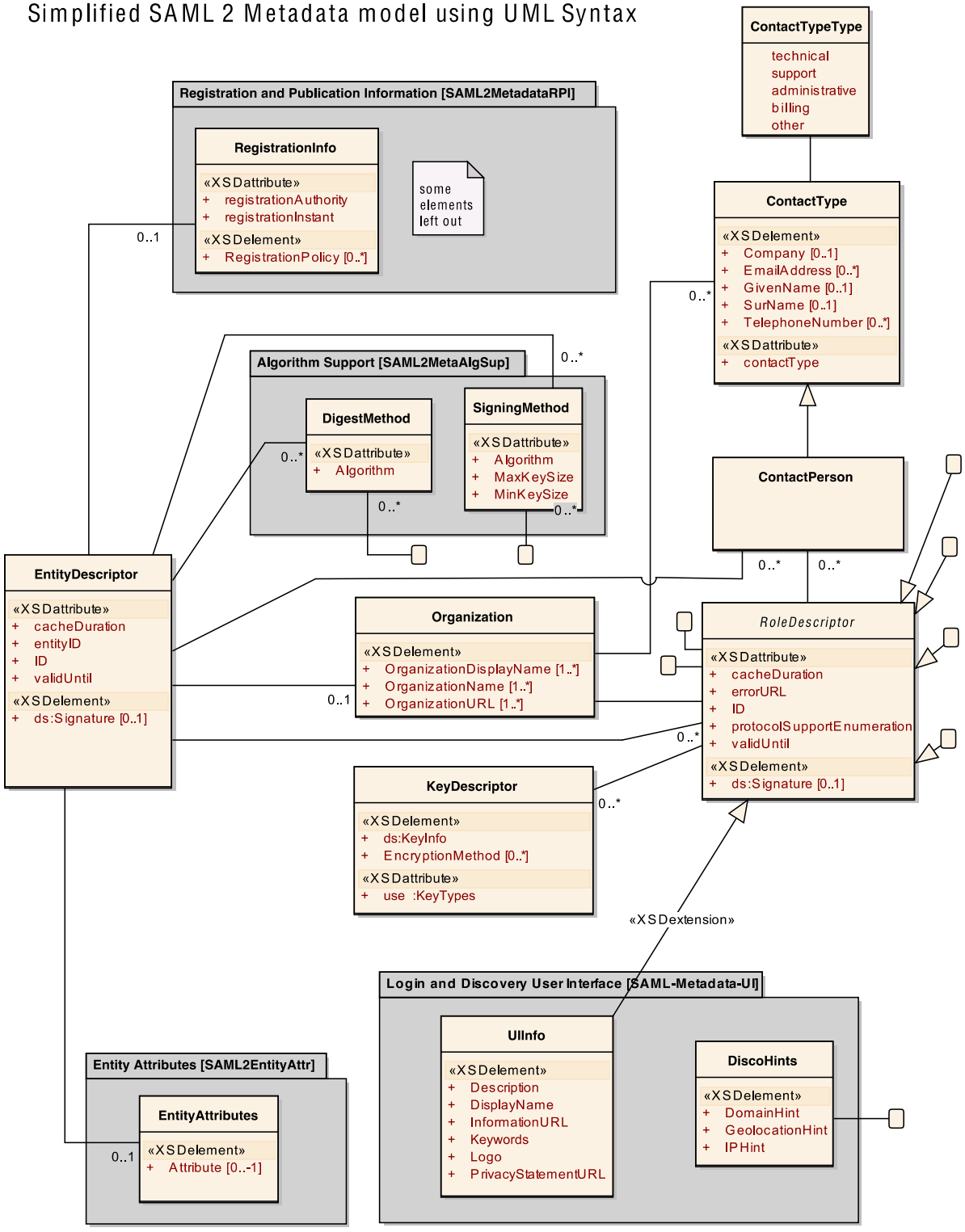
Die nachstehende Kodierung ist daher falsch, weil sie die URL Kodierung verwendet (siehe die %26):
*<AssertionConsumerService>https://example.gv.at/?foo=value**%26**bar=value</Assertion ConsumerService>*
Stattdessen sollte sie so aussehen (siehe das &):
*<AssertionConsumerService>https://example.gv.at/?foo=value**&**bar=value</Assertio nConsumerService>*

5.9 Erweiterungen

Da sich der Federation Operator für die veröffentlichten Metadaten verbürgt, sind Metadaten auf die Policy und die Verfahren der Entity Registrierung beschränkt. Daher sind nur die Datenelemente und Attribute zulässig, die ausdrücklich in dieser Spezifikation festgelegt sind.
Im Ergebnis MUSS der Federation Operator alle Elemente und Attribute entfernen, die nicht ausdrücklich in einer der referenzierten Metadaten-Spezifikationen festgelegt sind.

5.10 SAML Metadaten Struktur

Die nachstehenden Abbildungen Abbildung 3 und Abbildung 4 zeigen das gesamte SAML 2 Metadaten Modell unter Verwendung der UML Syntax. Die Abbildungen geben dem Leser einen umfassenden und kompakten Überblick über die Metadaten Elemente, die in diesem Papier verwendet werden.



528
529 **Abbildung 3: Metadaten Struktur (1 von 2)**

530

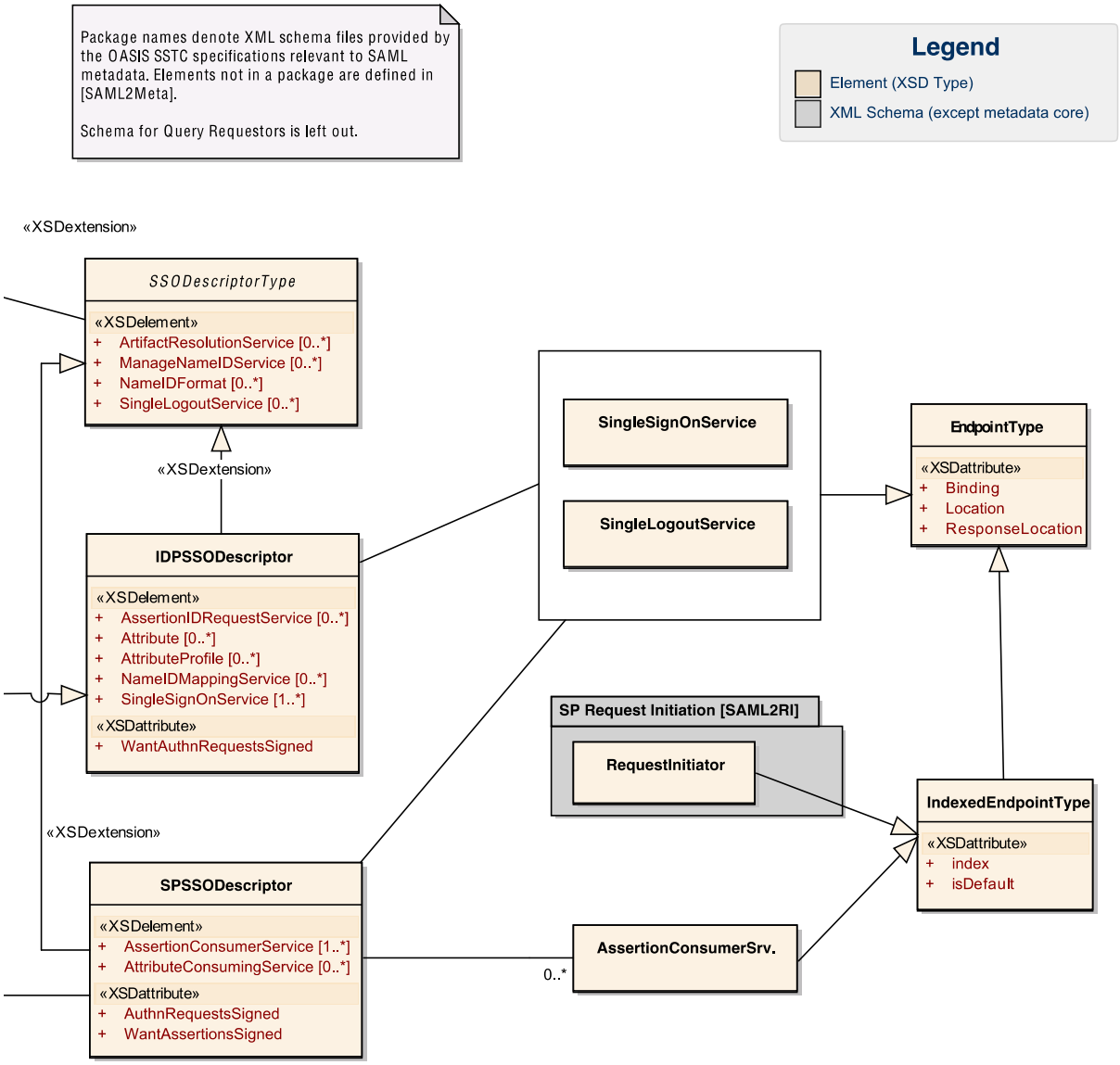


Abbildung 4: Metadaten Struktur (2 von 2)

5.11 Schematron Regeln

Der EntitiesDescriptor (Aggregation) bzw. der *EntityDescriptor* (bei MDX) MUSS konform zu den Regeln sein, die in [PVP2-Schematron] niedergelegt sind.
Ein Validierungsservice ist unter http://md.test.portalverbund.gv.at verfügbar

6 Metadaten Registrierung und Publikation

Von den verschiedenen Möglichkeiten[9], den EntityDescriptor eines IdP/SP sicher an den FO zu übertragen, ist das Sign-after-Upload Verfahren das bevorzugte Verfahren. Andere Verfahren, die eine vergleichbare oder bessere Sicherheit bieten, dürfen auch verwendet werden. Der Use Case umfasst die folgenden Elemente:
Voraussetzungen:

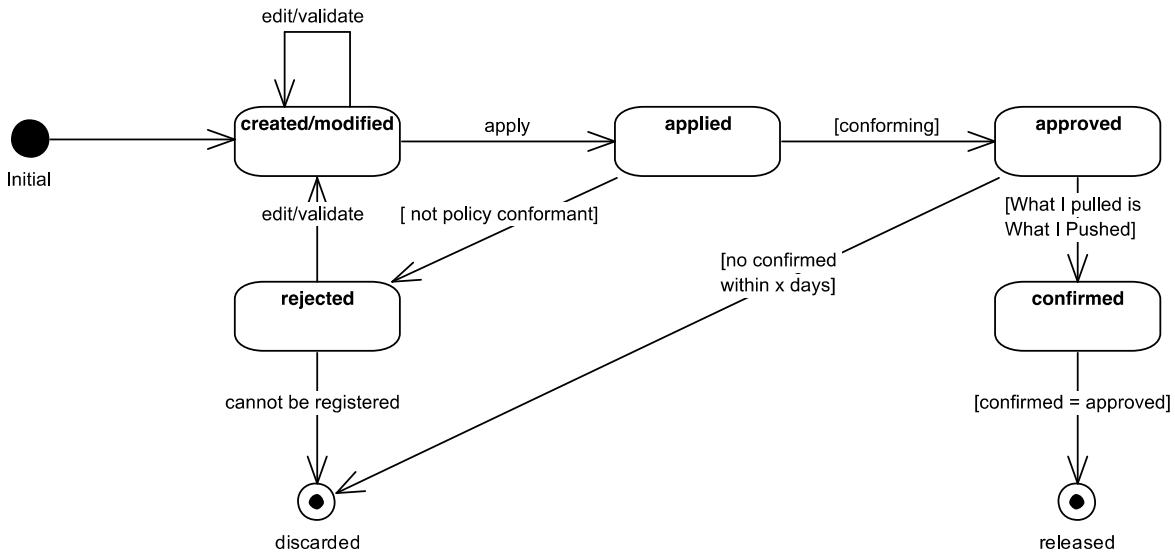
- Der Entity Operator ist ein registrierter Teilnehmer der Föderation.
- Der EO kann die Metadaten Registrierung authentifizieren, z.B. mittels PVP. Der FO stellt diesen Authentifizierungsdienst bereit.
- Der EO kann die Metadaten, unter Verwendung eines administrativen Schlüssels, signieren.

Der Use Case enthält für den Erfolgsfall folgende Schritte:

Arbeitsschritt	Status
Der Entity Operator lädt den *EntityDescriptor* in das Metadaten-Registry. Der Ablauf ist für neue Anträge und Updates identisch.	Created/modified
Der Entity Operator setzt den Status der Entity auf *applied*.	Applied
Der Föderation Operator validiert die hochgeladenen Metadaten entsprechend der Registrierungs-Policy, wie im "Metadata registration practice statement" niedergelegt (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**). Die Validierung ist erfolgreich und der Status wird auf *approved* gesetzt. Der *EntityDescriptor* wird um die Registrierungsdaten, wie in Sektion 5.3.6 niedergelegt, ergänzt.	Approved
Der validierte *EntityDescriptor* wird dem Entity Operator zur Verifizierung vorgelegt. Der Entity Operator muss den endgültigen *EntityDescriptor* verifizieren, signieren und wieder in das FO Metadaten-Registry uploaden.	Confirmed
Der signierte *EntityDescriptor* wird durch den FO authentifiziert, indem er dessen Signatur verifiziert und anschließend dem Aggregator freigibt, der ihn als neuen Eintrag eintragen kann, oder als Update, wenn es bereits eine frühere Version gibt.	released (→ published)

Die nachstehenden Zustandsdiagramme (Abbildung 5 und Abbildung 6) spezifizieren den Lebenszyklus eines *EntityDescriptor* im Registrierungs- und Veröffentlichungsprozess.

[9] Die Entscheidung über das Design ist in dokumentiert



554
555 **Abbildung 5: Metadaten-Registrierungs-Zyklus**

556 Zusätzlich zum erfolgreichen Ablauf im Use Case über die Statusmaschine wird der Ablauf für
557 abgelehnte Uploads und Einträge gezeigt, die „approved“, aber nicht mit einer gültigen Signatur
558 innerhalb der durch die Policy vorgegebenen Zeit bestätigt wurden. Nachstehend sind die
559 individuellen Zustände im Detail beschrieben.
560
561

Zustände	Details
Initial	Der *EntityDescriptor* wird erstellt.
Created/modified	Eine erste oder aktualisierte Version der Entity Metadaten (*EntityDescriptor*) wird in das Metadaten-Registry-Tool hochgeladen. Möglicherweise ist sie nicht gültig.
Applied	Der Betreiber der System Entity beantragte die Registrierung oder Änderung eines *EntityDescriptor*. Die System Entity ist betriebsbereit.
Approved	Der Metadata Registrar hat die notwendigen Überprüfungen gemäß der Registration-Policy durchgeführt. Der *EntityDescriptor* ist gültig.
Confirmed	Der Betreiber der System Entity bestätigt, dass der im Approval Feed veröffentlichte *EntityDescriptor* richtig ist und die Schlüssel und Adressen die gleichen sind, wie im Status *applied*. (Dieser Schritt verhindert, dass ein Angreifer bösartige Daten in den Registrierungsprozess einschleust).
Released	Der FO fügt den *confirmed EntityDescriptor* dem Feed zur Veröffentlichung hinzu. Der *EntityDescriptor* kann von allen Entities in der Federation verwendet werden. Bitte beachten: Der freigegebene *EntityDescriptor* ist jetzt ein anderes Objekt. Änderungen am *EntityDescriptor* in der Zustandsmaschine beeinflussen den *released EntityDescriptor* so lange nicht, bis er wieder freigegeben (*released*) wird.
Rejected	Der *EntityDescriptor* hat die Registrierungsprüfung nicht bestanden und wurde daher vom FO abgelehnt.

Zustände	Details
Discarded	Die Entity kann gemäß der Policy nicht registriert werden, die Bewerbung wird zurückgezogen.

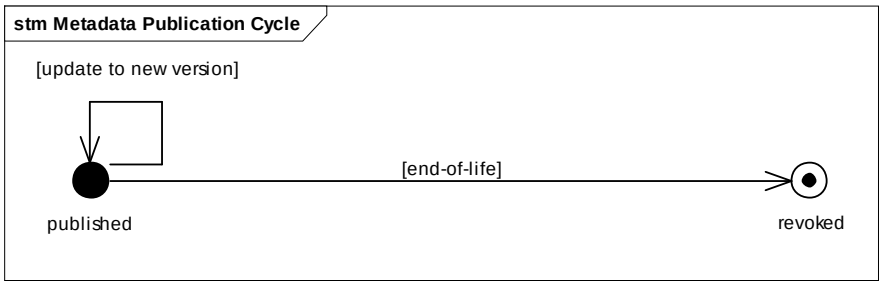


Abbildung 6: Metadaten-Veröffentlichungs-Zyklus

Nachdem ein EntityDescriptor im Registrierungs-Zyklus freigegeben wurde (Released), wird er im Metadaten-Feed eingefügt, wenn er neu ist oder ein Update einer früheren Version gemacht wurde. Wenn eine Entity aus der Föderation entfernt wird, wird ihr Status auf *revoked* gesetzt. Dies entfernt die Entity aus dem Feed. Es gibt keine Regelung um die Annullierung einer Entity rückgängig zu machen. Wenn Sie einmal annulliert (*revoked*) wurde, kann sie nur wieder teilnehmen, indem sie einen neuen Registrierungsprozess durchläuft. Nachstehend sind die individuellen Zustände im Detail beschrieben.

Zustände	Details
Published	Der released EntityDescriptor - entweder eine neue System Entity oder als Ersatz für eine ältere Version - wurde auf den Metadaten Feed des FO veröffentlicht.
Revoked	Der EntityDescriptor darf nicht mehr länger verwendet werden.

7 SAML V2.0 Metadata Spezifikationen

Diese Aufstellung bietet einen Überblick über die Dokumente zur OASIS SSTC Metadaten Spezifikation, die in diesem Papier Verwendung finden.

7.1 SAML Metadata 2.0 [SAML2Meta]

Titel des Dokuments: Schema for SAML metadata V2.0, March, 2005
Webseite: http://docs.oasis-open.org/security/saml/v2.0/
Schema-Datei: saml-schema-metadata-2.0.xsd

SAML Profile erfordern Vereinbarungen zwischen System Entities über Identifikatoren, verbindlichen Support, Endpunkte, Zertifikate, Schlüssel und anderes mehr. Eine Metadaten Spezifikation ist nützlich, um diese Information in standardisierter Form zu beschreiben. Diese Spezifikation definiert ein erweiterbares Metadaten Format für SAML System Entities, geordnet durch Rollen, die die SAML Profile widerspiegeln. Solche Rollen beinhalten auch die des IdP und SP.

7.2 SAML Metadata Interoperability [SAML2MDIOP]

Titel des Dokuments: SAML V2.0 Metadata Interoperability Profile V1.0, August 2009.
Webseite: http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf

Dieses Profil soll die Verwendung von Metadaten verbessern und verdeutlichen, wenn es darum geht, Interoperabilität bei der Provisionierung föderierter Beziehungen zwischen Delpoyments zu erreichen, und dabei auch das Vertrauen in kryptografische Signaturen und Handshakes zu etablieren.
Wenn eine Implementierung ausschließlich darauf beruht Vertrauen aus Metadaten herzuleiten, so ist das leichter verständlich und Sicherheitsrisiken können gut untersucht werden.

7.3 Algorithm Support [SAML2MetaAlgSup]

Titel des Dokuments: Metadata Profile for Algorithm Support Version 1.0, June 2010
Webseite: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-v1.0-cs01.pdf
Schema Datei: sstc-saml-metadata- algsupport-v1.0.xsd

Eine der Herausforderungen an die Interoperabilität in großen und langfristigen SAML Deployments ist die Aushandlung der Algorithmen für XML Signatur [XMLSig] und Verschlüsselung [XMLEnc] zur Laufzeit. Besonders schwierig ist es, wenn neue Algorithmen unterstützt werden sollten, um die Systeme schrittweise zu verbessern, aber das Wissen über die Möglichkeiten der angesprochenen Entity fehlt.
Dieses Profil nutzt SAML Metadaten, um es EOs zu ermöglichen, die Fähigkeiten und Präferenzen der Algorithmen ihrer Entities zu dokumentieren. Es erlaubt auch die zukünftige Erweiterung, um Anforderungen an die Interoperabilität von komplexeren Algorithmen zu adressieren.

7.4 IdP Discovery [IdpDisco]

Titel des Dokuments: Identity Provider Discovery Service Protocol and Profile V1.0, January 2007
Webseite: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery-cs-01.pdf
Schema Datei: sstc-saml-idp-discovery.xsd

Dieses Profil definiert eine Metadaten Erweiterung, um das Ergebnis eines redirect-basierten IDP Discovery Service zu verwenden.

7.5 Entity Attribute [SAML2EntityAttr]

Titel des Dokuments: SAML V2.0 Metadata Extension for Entity Attributes V 1.0 (August 2009)
Webseite: http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr-cs-01.pdf
Schema Datei: sstc-metadata-attr.xsd

Dieses Profil definiert ein Erweiterungselement als Container für Eigenschaften von Entities, das ermöglicht, dass über Attribut-Informationen zusätzliche Daten über die betreffende Entity an einen Metadata Consumer kommuniziert werden können, in der Art wie eine Assertion Attribute über ein Subjekt halten kann. Eine mögliche Anwendung dieses Mechanismus ist es, dass sie es einem Federation Operator ermöglicht, erweiterbare Informationen mit einzuschließen, wenn die Entities die optionalen Federation Policys einhalten. Dieses Profil definiert keine besonderen Attribute, die zu kommunizieren sind, allerdings könnten zusätzliche Profile es einsetzen, um dies zu tun.

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

Titel des Documents: Metadata Extension Schema for SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0, 01 November 2010
Webseite: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-ui/v1.0/sstc-saml-metadata-ui-v1.0.pdf
Schema Datei: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-ui/v1.0/cs01/xsd/sstc-saml-metadata-ui-v1.0.xsd

Dieses Erweiterungsprofil enthält mehrere Attribute, die SAML Entities zugeordnet sind. Diese ermöglichen eine reichhaltigere und bessere Benutzererfahrung bei der Auswahl eines IdP. Zu den Informationen gehören Daten, wie Name, Logo, Datenschutzerklärung und geographischer Standort.
Da Entity Attribute nicht gezielt pro Rolle hinzugefügt werden können, sondern nur pro Entity, erlaubt diese Erweiterung es, dass Informationen spezifisch für Discovery Services hinzugefügt werden können.

7.7 SP Request Initiation Protocol [SAML2RI]

Titel des Dokuments: Service Provider Request Initiation Protocol and Profile V1.0, March 2010
Webseite: http://www.oasis-open.org/committees/documents.php?wg_abbrev=security
Schema Datei: sstc-request-initiation.xsd

SPs können eine Erweiterung nutzen, um den Endpunkt des Request Initiators zu beschreiben. Diese wird als Hilfe für die manuelle oder automatisierte Erstellung von Links zur Unterstützung des Request Initiation Protocol benutzt.

8 Verbindung zu ldap.gv.at

SAML Metadaten bieten ein grundlegendes Zugriffsmanagement:

- Zusichern, dass Entities rechtmäßige PV-Teilnehmer sind;
- Zusicherung von Entity Categories, die Attribute aufweisen, die eine Entity zu erhalten berechtigt ist.

LDAP.gv.at hat eine erweiterte Fähigkeit:

- Aufstellung von Rechten, entsprechender Parameter und Mindest-Sicherheitsklassen für Rechte von Anwendungen (maschinenlesbar) und Dokumentation der rechtlichen Basis, um diese Rechte zu erhalten (für Menschen lesbar).
- Aufstellung des Satzes an Rechten und entsprechender Parameter, die ein PV-Teilnehmer bzw. eine zugriffsberechtigte Stelle verwenden darf.

Die Daten des Rechte-Managements werden nicht in SAML Metadaten dupliziert. LDAP bietet Links zu *EntityDescriptors* unter Verwendung des entityID Attributs. Anwendungen, die PVP2/SAML nutzen, müssen auf entsprechende LDAP Einträge zugreifen (wie bspw. gvApplicationRights), um Zugriffsentscheidungen zu treffen.

9 Überlegungen zum Betrieb

Federation Metadaten sind der Dreh- und Angelpunkt für den Betrieb einer Föderation. Daher müssen die klassischen Sicherheitsziele, Vertraulichkeit (Schutz privater Schlüssel), Verfügbarkeit (gültiger Metadaten und Widerrufsinformationen) und Integrität (Metadaten aus einer maßgeblichen Quelle) gewährleistet sein. Normalerweise wird dies über eine Service-Level-Vereinbarung erreicht.

Mit Blick auf diese Parameter sollte ein Hardware Security Module (HSM) in der Klasse eines Zertifizierungsdiensteanbieters überlegt werden. Im Vergleich zur Signatur mit Smart Cards hat es den Vorteil der Ausfallsicherung, der Wiederherstellung und weiterer Business Continuity Funktionen.

Die Aktualität von FO Metadaten MUSS durch eine mindestens tägliche Anforderung von Updates seitens der System Entity sichergestellt sein. Der FO braucht keine Benachrichtigungs-E-Mails über Änderungen an den EO zu schicken. Anwendungen SOLLTEN Metadaten in kurzen Intervallen (bspw. 10 Minuten) abrufen. Für Notfälle SOLLTE ein Notfallplan als Teil der Federation Policy etabliert werden (siehe 11 Abkürzungen und Begriffe).

10 Referenzen

durch den OASIS SSTC sind in Sektion 6 beschrieben. PVP-Dokumente werden unter http://reference.e-government.gv.at publiziert.

[PVP2-Schematron]	Regeln, die in dem Saml_schematron Projekt definiert sind. https://github.com/rhoerbe/saml_schematron/blob/master/profiles/pvp2.sch (Code) und https://github.com/rhoerbe/saml_schematron/blob/master/doc/ruleset.rtf (Dokumentation)
PVP2-S-Profil	**Portalverbundprotokoll Version 2** S-Profil V2.1.0 http://reference.e-government.gv.at/AG-IZ-PVP2-Version-2-1-0-2.2754.0.html
[PVP2-Attr]	Portalverbundprotokoll Version 2 eGovernment Attribute Profile V2.1.x
[PVP-Glossar]	Identity Management Glossar Version 1.0.0
[PVV]	Portalverbundvereinbarung V1.0 http://reference.e-government.gv.at/AG-IZ-PVP2-Version-2-1-0-2.2754.0.html
[FeideMaRequ]	Metadata Aggregation Requirements Specification, Andreas Solberg, 2010-01-05. Heruntergeladen von https://rnd.feide.no/2010/01/05/metadata_aggregation_requirements_specification
[SAML-metadata-guide]	**SAML Metadata Guidance Version 1.0** OASIS SAML SSTC, propsed working draft vom 21-Jul-2014, noch nicht veröffentlicht.

11 Abkürzungen und Begriffe

Allgemeine Begriffe für den Portalverbund und PVP sind in [PVV] und [PVP-Glossar] definiert.

AG-IZ	Arbeitsgruppe Integration und Zugänge
BLSG	Bund-Länder-Städte-Gemeinden
CBC	Cipher Block Chaining Mode
EO	Entity Operator
Feide	Feide.no is the Norwegian academic access & identity federation

FO	Federation Operator
HSM	Hardware Security Module
IIW/EWTI	Internet Identity Workshop, European Workshop for Trust & Identity
IdP	Identity Provider
MA	Metadata Aggregator
MC	Metadata Consumer
MDX	Metadata Exchange
MP	Metadata Publisher
MR	Metadata Registrar
PVP	Portalverbundprotokoll
PVV	Portalverbundvereinbarung
REFEDS	Research and Education FEDerations community platform
SAMLSAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SPSP	Service ProviderService Provider
SSL	Secure Socket Layer
TERENA	Trans-European Research and Education Networking Association
TLS	Transport Layer Security
USP	Unternehmensserviceportal
XML	Extensible Markup Language

707
708
709 Die nachstehenden Begriffe sind in anderen Papieren beschrieben und definiert.

Begriff	Beschreibung
Federation Policy	Die Federation Policy ist die regulatorische Grundlage für die Föderation. Im Kern umfasst sie die zugrundeliegende "Portalverbundvereinbarung" (PVV) und weitere Dokumente für Servicebetreiber und externe Anbieter von Anwendungen.
Metadata Key Management Practice Statement	Das "Metadata Key Management Practice Statement" ist Bestandteil der Federation Policy, die die Regeln für die Erzeugung, Verteilung, Speicherung, Revocation und Nutzung der Schlüssel definiert, die verwendet werden, um Metadaten zu signieren, also die Metadaten Signierschlüssel des FO und EO.
Metadata Registration Practice Statement	Das "Metadata Registration Practice Statement" ist Bestandteil der Federation Policy, die die Regeln für die Einreichung, Validierung, Aggregation und Veröffentlichung der Metadaten regelt.
Contingency procedures	Die Notfallpläne umfassen einen Satz von Arbeitsabläufen für den Umgang mit Ausnahmen, wie bspw. die Nicht-Verfügbarkeit von FO Metadaten Services oder die Gefährdung des FO Metadaten Schlüssels. Dies könnte bspw. eine Liste enthalten, die sowohl den FOs als auch den EOs zur Verfügung steht, um alle EO und FO Kontaktpersonen anzurufen und anzumailen.

710
711

Anhang 1 Design Entscheidungen

a. Hochgeladene Metadaten authentifizieren

Da Metadaten entscheidend für die Sicherheit einer Föderation sind, muss entsprechend darauf geachtet werden, dass die aggregierten Daten aus einer authentischen Quelle stammen. Es gibt mehrere Optionen, um die Echtheit der Metadaten, die der EO in das Metadaten Management System hochgeladen hat, sicherzustellen:

a) Signieren und downloaden. Der Entity Operator signiert den *EntityDescriptor* und der FO bezieht ihn von der well-known Location. (Geringfügiger) Nachteil: Der FO muss eine Liste der namhaften Speicherorte für die Teilnehmer der Föderation führen und diese regelmäßig aktualisieren.

b) Signieren und verteilen. Der Entity Operator signiert den *EntityDescriptor* und übermittelt ihn über einen verlässlichen Kanal, wie bspw. einen Web-Service oder einen elektronischen Zustelldienst an den FO.

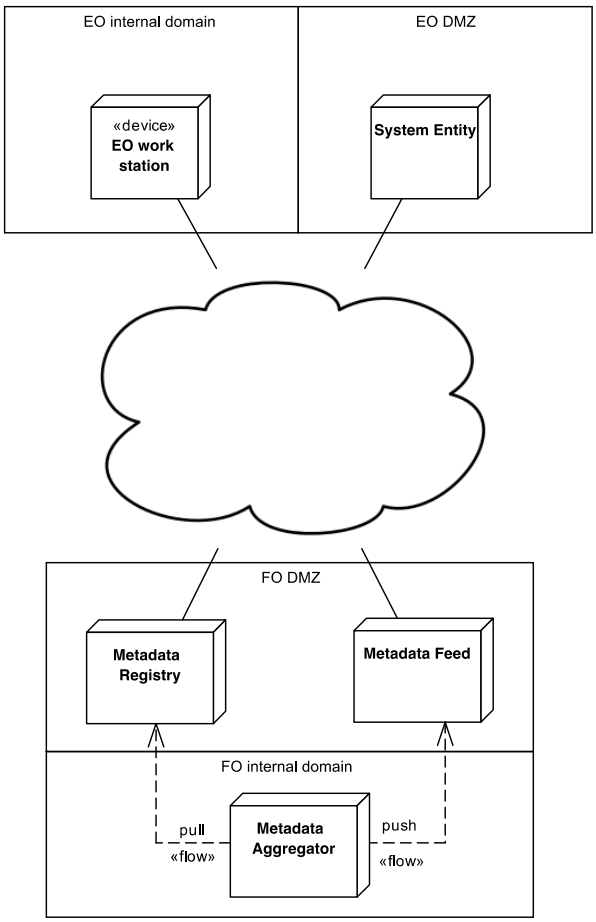
c) HTTPS Download. Der FO bezieht den *EntityDescriptor* von einer mit HTTPS geschützten Website. Nachteil: Aus Sicherheitsgründen muss das Zertifikat des TLS Clients auf einer Whitelist des Entity Operator validiert sein.

d) Out-of-Band Kanal: Der Entity Operator und der FO einigen sich auf eine sichere Übermittlung außerhalb des Netzwerks.

e) Signieren nach dem Upload. Metadaten werden auf eine authentifizierte, öffentliche Webanwendung hochgeladen, auf welcher sie editiert werden können und der FO Daten ergänzen kann. Zusätzliche Sicherheit wird durch eine Validierung des *EntityDescriptor* durch den Entity Operator erreicht, nachdem der *EntityDescriptor* von dem Metadata Aggregator (MA) verarbeitet wurde. Diese umfasst die folgenden Schritte:

- Der EO kann einen *EntityDescriptor* hochladen und editieren.
- Der FO gibt den *EntityDescriptor* zurück an den Entity Operator, nachdem dieser vom MA verarbeitet wurde, aber bevor er auf einem getrennten Feed durch den FO veröffentlicht wird.
- Der Entity Operator vergleicht das Originaldokument mit dem zurückgegebenen Dokument.
- Falls die sicherheitsrelevanten Elemente (Schlüssel, URLs) übereinstimmen, wird diese Zwischenversion des *EntityDescriptor* vom FO mit dem Signaturschlüssel der Entity signiert. Dieser Schritt sollte durch eine vom FO zur Verfügung gestellte XSLT und xmldiff Utility automatisiert werden.
- Der signierte *EntityDescriptor* wird dann an den Aggregator übermittelt, der den *EntityDescriptor* auf dem Veröffentlichungs-Feed veröffentlicht, nachdem die Signatur verifiziert wurde.

Das nachstehende Deploymentdiagramm zeigt ein Szenario für den Workflow des obigen Ablaufs.



Das Deployment-Diagramm zeigt die Systeme, die am Metadaten-Management beteiligt sind.

Die Annahme ist, dass das Metadaten Registry eine Benutzerschnittstelle zum Uploaden und Editieren bietet. Sie ist daher exponierter und verwundbarer als interne Systeme.

Eine Verringerung dieser Gefährdung ist es, wenn die Signaturschlüssel nur bei den internen System verfügbar sind. Im Falle des Metadaten Aggregator sollte der Signaturschlüssel durch ein HSM geschützt werden (nicht dargestellt).

751

752

b. Requested Attributes

Nachstehend sind Optionen für die Anforderung von Attributen durch einen SP festgelegt.

Optionen

Es gibt mehrere Optionen, um eine spezifische Liste von Attributen bei einem Identity Provider anzufordern.

1. Auflistung der Attribute im Authentication Request;
2. Attribute außerhalb von SAML kommunizieren;
3. Auflisten der Attribute in den SAML Metadaten (SPSSODescriptor/Attribute);
4. Referenzieren einer Liste von Attributen in SAML Metadaten durch eine Entity Category.

Kriterien, um die Optionen zu beurteilen, sind

- Support für PVP Use Cases
- Einfache Bereitstellung und Betrieb
- Produktsupport
- Nachrichtengröße

Support für PVP Use Cases

- Die bisher bekannten Use Cases erfordern einen von 3 festen Attribut-Sätzen (PVP-eGov-Token, PVP-eGov-Token-ext oder Citizen-Token).
- Möglicherweise kann STORK in Zukunft Anwendungen unterstützen, die andere Attribute anfordern.

Einfache Bereitstellung und Betrieb

Anhang 2 Wenn ein Service Provider aufgenommen wird, muss der Identity Provider eine Release Policy für Attribute konfigurieren, die in allen Praxisfällen statisch ist. Außer, wenn der Satz an Attributen klein ist, wäre eine Referenz auf eine vordefinierte Liste einfacher zu verifizieren und zu managen.

Anhang 3 Entity Categories können einfach Attribut-Mappings und Release Policies zugeordnet werden.

Produktsupport

- Shibboleth IdP und SimpleSAMLPhp unterstützen den Attribute Release basierend auf Entity Categories. Entity Categories sind jedoch in der Kernspezifikation von SAML nicht spezifiziert und ihre derzeitige Spezifikation (mitte 2014) ist noch im (allerdings fortgeschrittenen) Abstimmungsprozess.

Nachrichtengröße

- Benötigte Attribute sollten über Metadaten weitergegeben werden.

c. Unterstützung von Algorithmen

Es gibt drei generelle Wege, um mit der Migration auf Chiffren mit höherer Sicherheit umzugehen:

1. Out-of-Band Management
2. Obligatorisches automatisiertes Aushandeln
3. Maschinenlesbare Dokumentation mit optionalem automatisiertem Aushandeln

Bei Option 1 hat sich als Nachteil gezeigt, dass in größeren Netzwerken die Migration dadurch behindert wird, dass Entities unterstützt werden müssen, die noch nicht migriert haben. Da es keine formelle Dokumentation darüber gibt, wer was unterstützt, werden alte Chiffren nie abgeschaltet. Damit gibt es keinen Druck, neue Chiffren einzuführen.

Option 2 verlangt 100 % Produktsupport. Für die derzeitigen SAML Implementationen ist dieser nicht verfügbar. Ein Beispiel für diese Option ist das TLS Protokoll, jedoch werden Chiffren nicht statisch angekündigt, sondern in der Handshake-Phase. Daher ist der Support für Migrationen nicht optimal.

Option 3 ist ein Kompromiss zwischen Optionen 1 und 2. Die Produkte müssen nicht das automatisierte Aushandeln von Chiffren unterstützen, müssen aber ihre Fähigkeiten in den Metadaten veröffentlichen. Die Entfernung unerwünschter Algorithmen kann zuverlässig auf einer Metadaten Abfrage basieren.

d. Registration und Publication Information

Der Grund dafür diese Erweiterung in jedem *EntityDescriptor* aufzunehmen ist der mögliche Support für Interfederation. Metadaten Aggregatoren können Policy-Entscheidungen auf diesen Attributen basierend treffen.

e. mdui:DisplayName

Namensanzeigen für System Entities sind eine deutliche Verbesserung der Benutzeroberfläche für beide, Endanwender und Administratoren.
Eine typische Auswahlliste für Endanwender für den IDP wird `<mdui:DisplayName>`, `<mdui:Description>`, `<mdui:Logo>` enthalten. Der Text in `<mdui:DisplayName>` und `<mdui:Description>` sollte in erster Linie dem Endanwender helfen, wenn er IDPs in einer Liste unterscheiden soll.

Die nachstehende Tabelle zeigt Beispiele in drei typischen Szenarien:

	Szenario	Mustereintrag
1	Die Stammorganisation verfügt über einen einzelnen IDP	BMfX Anmeldeportal Portal zur Anmeldung an Webanwendungen für registrierte Benutzer des Bundesministeriums für Xxxx
2	Die Stammorganisation verfügt über mehrere IDPs	BMfY Anmeldeportal Portal zur Anmeldung an Webanwendungen für externe Benutzer des Bundesministeriums für Yyyy
3	Der virtuelle IDP der Stammorganisation gehostet bei einem externen Provider.	LFRZ Anmeldeportal Anmeldeportal für Kunden des LFRZ

▪ Änderungshistorie

PVP 2.1.1

- Dokument neu erstellt

PVP 2.1.2, PVP 2.1.3

- Anpassung an die neue Versionsnummer des Dokumentensets