eCH-0174 – SuisseTrustIAM-Implementierung mit SAML 2.0

Name	SuisseTrustIAM-Implementierung mit SAML 2.0
eCH-Nummer	eCH-0174
Kategorie	Standard
Reifegrad	experimentell
Version	1.0
Status	Aufgehoben
Genehmigt am	2015-11-25
Ausgabedatum	2015-11-26
Ersetzt Standard	-
Sprachen	Deutsch
Beilagen	keine
Autoren / Kontakt	Annett Laube-Rosenpflanzer, BFH, annett.laube@bfh.ch Gerhard Hassenstein, BFH, gerhard.hassenstein@bfh.ch
Herausgeber / Vertrieb	Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Zusammenfassung

SuisseTrustIAM ermöglicht es, digitale Identitäten auf einfache Weise über Unternehmensgrenzen hinweg auszutauschen, um dadurch Geschäftsprozesse zu vereinfachen. SuisseTrustIAM ist eine generische Vermittlerinfrastruktur, auf welcher Identity Federations und IAM-Lösungen abgebildet werden können.

Dieses Dokument beschreibt die Anforderungen und Empfehlungen für die Implementierung einer SuisseTrustIAM-Umgebung mittels SAML 2.0-Protokollen. Dabei stehen Kompatibilität, Transparenz und Interoperabilität zwischen den STIAM-Komponenten im Vordergrund.

Die Vorgabe einer standardisierten Technologie mit wenigen Optionen soll es ermöglichen, dass Applikationen (STIAM-Empfänger mit ihren Ressourcen) einfach, schnell und ohne grossen Aufwand in eine STIAM-Community eingebunden werden können.

Im Rahmen der Spezifizierung von SuisseTrustIAM wurden eine Reihe weiterer Dokumente erarbeitet. Dieses Dokument beschreibt eine vereinfachte Umsetzung der Konzepte aus eCH-0167 ‚SuisseTrustIAM technische Architektur und Prozesse' mit SAML 2.0.

Inhaltsverzeichnis

Status des Dokuments

Aufgehoben: Das Dokument wurde von eCH zurückgezogen. Er darf nicht mehr genutzt werden.

Notation

Die Schlüsselworte MUSS *(MUST)*, DARF NICHT *(MUST NOT)*, ERFORDERLICH *(REQUIRED)*, SOLLTE *(SHOULD)*, SOLLTE NICHT *(SHOULD NOT)*, EMPFOHLEN *(RECOMMENDED)*, KANN *(MAY)* und OPTIONAL in diesem Dokument sind zu interpretieren wie in IETF RFC 2119 beschrieben. [1]

Die in diesem Dokument aufgeführten Präfixe, referenzieren folgende XML-Namensräume:

Präfix	XML-Namensraum
`saml:`	urn:oasis:names:tc:SAML:2.0:assertion
`samlp:`	urn:oasis:names:tc:SAML:2.0:protocol
`md:`	urn:oasis:names:tc:SAML:2.0:metadata
`ds:`	http://www.w3.org/2000/09/xmldsig#
`hoksso:`	urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser

Tabelle 1: Präfixe und referenzierte XML-Namensräume

1 Einleitung

Sprachlicher Hinweis: Aus Gründen der besseren Lesbarkeit werden Personenbezeichnungen lediglich in der männlichen oder weiblichen Form verwendet. Diese Formulierungen schliessen das andere Geschlecht jeweils automatisch mit ein.

1.1 Einordnung

Unter dem Standard eCH-0107 [2] positionieren sich die Konzepte für föderierte IAM-Lösungen und ergänzende Hilfsmittel.

Neben den Federation-Konzepten stellen die Hilfsmittel ergänzende Informationen zur Verfügung. Die dargestellten Qualitäts- und Maturitätsmodelle sind Beispiele für Hilfsmittel, und sind nicht abschliessend.

Der Standard eCH-0167 [3] beschreibt ein IAM-Lösungskonzept zur Bereitstellung einer organisationsübergreifenden Vermittlerinfrastruktur, die im Standard eCH-0168 [4] detailliert, aber weitgehend technologie-unabhängig beschrieben wird.

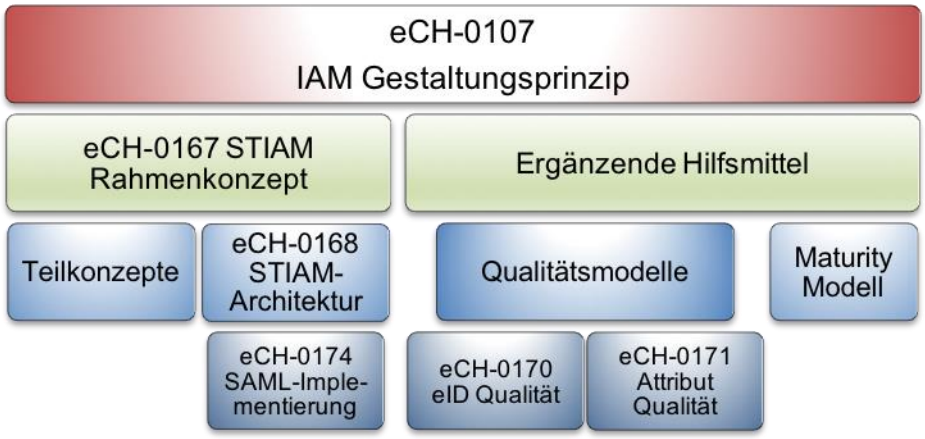

Abbildung 1: Einordung des eCH-0174 Standards

Der vorliegende Standard eCH-0174 reiht sich in die Dokumentenhierarchie der föderierten IAM-Lösungen als Implementierungs-Vorschlag der technischen STIAM-Architektur ein und basiert damit auf dem bereits veröffentlichten Dokument eCH-0168 [4].

1.2 Ziel des Dokuments

Dieses Dokument beschreibt die Anforderungen und Empfehlungen für die Implementierung einer SuisseTrustIAM-Umgebung mittels SAML 2.0-Protokollen.

SuisseTrustIAM beschreibt ein IAM-Gesamtsystem, welches auf einer Hub-'n'-Spoke-Architektur beruht und verfolgt das primäre Ziel, die Anbindung von Applikationen (Ressourcen einer Relying Party) in einen IAM-Verbund möglichst einfach zu gestalten. Eine Relying Party (STIAM-Empfänger) kann damit auf einfache Art und Weise Identitätsinformationen eines Endbenutzers von einer vertrauenswürdigen Instanz (Identity Hub, STIAM-Hub) beziehen. Die benötigten Benutzerinformationen stammen dabei in der Regel aus unterschiedlichen Quellen, werden vom STIAM-Hub zusammengetragen und als Ganzes in Form einer

Assertion dem anfragenden STIAM-Empfänger übergeben. Diese Assertion kann vom STIAM-Empfänger dazu benutzt werden, Zugriffsrechte des Endbenutzers lokal zu steuern.

Mit dem vorliegenden Dokument sollen die umfassenden Möglichkeiten einer IAM-Architektur, wie sie im eCH-0168 [4] beschrieben wurden, stark eingegrenzt werden. Nicht eine Vielfalt von möglichen Optionen steht im Vordergrund, sondern ein möglichst hohes Mass an Kompatibilität, Transparenz und Interoperabilität zwischen den Komponenten. Damit soll eine einfachere und schnellere Integration der peripheren Komponenten basierend auf einer standardisierten Technologie ermöglicht werden.

1.3 Aufbau des Dokuments

In Kapitel 2 werden die Anforderungen an die einzelnen STIAM-Komponenten zur Definitionszeit und zur Laufzeit beschrieben, die für eine SAML V2.0-Implementierung notwendig sind.

Kapitel 3 beschreibt die verschiedenen SAML-Services, die von den STIAM-Komponenten zur Verfügung gestellt werden müssen, sowie die Interaktionen dieser Services.

Kapitel 4 dokumentiert detailliert die Protokolle und deren Inhalte zur Lauf- und Definitionszeit.

Die Definition der Komponenten (STIAM-Empfänger, STIAM-IdPs und STIAM-Sender), Ressourcen und Attribute erfolgt auf dem STIAM-Hub über ein Konfigurationsportal. In Kapitel 5 werden die Informationen der STIAM-Komponenten, die für eine SAML 2.0-Implementierung benötigt werden, standardisiert als SAML-Metadaten beschrieben.

Kapitel 6 zeigt anhand von Beispielen (mit detaillierter Beschreibung) die konkrete Form der SAML 2.0-Nachrichten zwischen den einzelnen STIAM-Komponenten. Damit soll die Interoperabilität und Kompatibilität zwischen den einzelnen Komponenten verstärkt werden.

In Kapitel 7 werden Erweiterungen und Spezialfälle aufgezeigt, die von der beschriebenen Umsetzung abgebildet werden können, aber nicht direkt ersichtlich sind.

1.4 Abgrenzung

Im vorliegenden Dokument wird bewusst darauf verzichtet, Vorgaben zur Umsetzung der einzelnen Komponenten zu machen. Wie die internen Funktionen der einzelnen Komponenten konkret aussehen, ist nicht Teil dieses Dokuments. Der Hauptfokus liegt bei einem möglichst hohen Mass an Standardisierung bezüglich Interaktion und Kommunikation zwischen STIAM-Hub und den peripheren Komponenten STIAM-Empfänger (Relying Party), Identity Provider und STIAM-Sender (Attribute Authority), basierend auf SAML 2.0.

2 Anforderung

Dieses Dokument beschreibt eine STIAM-Implementierung mit SAML 2.0 [5], die den folgenden Einschränkungen gegenüber dem eCH-0168 [4] genügt:

1. Es werden nur die Funktionalitäten beschrieben, die für eine lauffähige STIAM-Lösung auf SAML 2.0 minimal erforderlich sind, d.h. auf alle optionalen Anforderungen und auf Protokolle neben dem SAML 2.0 Web Browser SSO Profile mit HTTP Binding (HTTP POST bzw. HTTP redirect) und SAML 2.0 Assertion Query/Request Profile wird verzichtet.
2. Es werden nur Relying Party-initiierte (SP-First) Szenarien beschrieben. IdP-First Szenarien sind zwar ebenfalls denkbar, werden aber nicht unterstützt.
3. Alle STIAM-Komponenten vertrauen dem STIAM-Hub vollumfänglich. Der STIAM-Hub erfüllt seine Aufgaben entsprechend der definierten Policies und dem Governance Framework (siehe eCH-0169 [6]). Daher wird auf eine Unterstützung des Identity Relaying Modus [4] und der End-to-End-Verschlüsselung [4] verzichtet. Allerdings sollten die ausgetauschten Informationen (Assertions) zwischen STIAM-Hub und STIAM-Empfänger sowie zwischen STIAM-Hub und STIAM-IdP/STIAM-AA verschlüsselt werden.
4. Alle Metainformationen der STIAM-Komponenten werden in der zentralen Datenbasis (Komponenten-Management, Attribut-Management) des STIAM-Hubs abgelegt. Sie bestehen aus SAML-Metadaten (beschrieben in Kapitel 5) sowie zusätzlichen Informationen, wie QAA-Level, und werden von dazu autorisierten Personen (OrgSysAdmin) gepflegt.
5. Der STIAM-Sender ist stateless[1] und delegiert die Authentifizierung des Benutzers an den STIAM-Hub. Im Normalfall vertraut er darauf, dass zum Zeitpunkt der Attribut-Anfrage der Benutzer entsprechend den Richtlinien in der Konfiguration authentifiziert wurde und überprüft dies nicht noch einmal selbst. Optional kann der STIAM-Sender aber die Durchreichung einer gültigen Authentisierungsassertion vom STIAM-Hub verlangen (Extended Attribute Query).
6. Der STIAM-Hub vermittelt Identitäts- und Attributsinformationen an die STIAM-Empfänger. Diese Informationen werden zeitnah eingeholt, vom STIAM-Hub geprüft und können zwecks Aggregation, zur Erhöhung der Benutzerfreundlichkeit oder aus Performancegründen zwischengespeichert werden.
7. Der User Consent MUSS für alle persönlichen Attribute immer am STIAM-Hub eingeholt werden.[2] Dabei sollten alle Attribute gemeinsam bestätigt und auch öffentliche

[1] Das in eCH-0168 [4] beschriebene Verfahren zur Identitätsvermittlung an den STIAM-Sender setzt voraus, dass der STIAM-Sender statefull ist und über ein eigenes (SSO-) Sessionhandling verfügt.

[2] Entsprechend der Definition der „angebotenen Attribute“ (siehe [4], oder Kapitel 4.1.1, S. 25) wird zwischen persönlichen (privaten) und öffentlichen (public) Attributen unterschieden.

oder organisationsinterne Attribute angezeigt werden. Der User Consent SOLLTE nur einmal pro RP eingeholt und nur bei Änderungen des Attribut-Sets erneuert werden.

8. Es werden nur die Funktionen und Nachrichten zur Einbindung von STIAM-IdPs beschrieben, da für diese entsprechende Vorgaben gemacht werden können. Externe IdPs, welche SAML 2.0 unterstützen (z.B. SuisseID), müssen individuell betrachtet werden.
9. Single Logout MUSS vom STIAM-Hub und SOLLTE von den STIAM-Empfängern unterstützt werden.

2.1 Allgemeine Anforderungen an die Vermittlerinfrastruktur

Die in eCH-0168 [4] Kapitel 4, Tabelle 3 beschriebenen allgemeinen Anforderungen an die STIAM-Vermittlerinfrastruktur gelten ohne Ausnahme auch für eCH-0174.

2.2 Generelle Anforderungen

In Tabelle 2 werden generelle Kommunikations- bzw. SAML 2.0-spezifische Anforderungen an das Gesamtsystem beschrieben.

Anforderung	Beschreibung	Kommentar
Data Transfer	Der gesamte Datentransfer zwischen allen Komponenten im STIAM-Verbund MUSS mittels TLS abgesichert werden.	Alle angeschlossenen STIAM-Komponenten verwenden TLS-Zertifikate, ausgestellt von einem anerkannten CSP.
Zeit-Synchronisation	Eine Komponente im STIAM-Verbund, welche SAML-Nachrichten ausstellt oder konsumiert, MUSS ihre Zeit mit zugelassenem Zeitserver synchronisieren.	Synchronisierung über NTP.
Konformität	Alle verwendeten Standards (SAML 2.0, HTTP etc.) MÜSSEN den Spezifikationen entsprechend angewandt werden.	
Sicherheit und Vertrauen	Eine Komponente im STIAM-Verbund MUSS die von ihr ausgestellten SAML-Nachrichten und evtl. enthaltenen Assertions signieren. Die Assertions SOLLTEN verschlüsselt sein. Bei empfangenen SAML-Nachrichten MUSS die Authentizität und Integrität der Nachricht und der enthaltenen Assertions anhand der Signaturen geprüft werden.	

Tabelle 2: Generelle Anforderungen

Für den Fall eines organisationsinternen IAMs (Enterprise-IAM), bei dem Ressourcen und Identitätsdaten der Organisation gehören, ist kein User Consent notwendig.

2.3 STIAM-Empfänger

In Tabelle 3 und Tabelle 4 werden die notwendigen Funktionen eines STIAM-Empfängers aufgelistet.

Funktionen zur Definitionszeit:

Funktion	Beschreibung	Kommentar
E-DZ-01	Der STIAM-Empfänger MUSS ein STIAM-Applikationszertifikat (X.509) bei einer vertrauenswürdigen CA (STIAM-CSP) beziehen und für seine SAML-Services installieren.	Siehe SAML-Services in Kapitel 3.
E-DZ-02	Der STIAM-Empfänger MUSS im Komponenten-Management (KM) des STIAM-Hubs als Relying Party registriert werden.	Siehe eCH-0168 [4].
E-DZ-03	Die Ressourcen des STIAM-Empfängers MÜSSEN im KM des STIAM-Hub definiert werden. Es MUSS pro Ressource ein QAA-Level (gemäss eCH-0170 [7]) festgelegt werden. Es KÖNNEN angeforderte Attribute und deren Qualität angegeben werden.	
E-DZ-04	Der STIAM-Empfänger KANN den Typ der von ihm erwarteten NameID auf Persistent setzen.	Standardmässig gibt der STIAM-Hub eine Transient-NameID an den STIAM-Empfänger zurück.
E-DZ-05	Der STIAM-Empfänger KANN im KM für eine Ressource einen bestimmten Identity Provider vorgeben.	Der gewählte IdP muss den geforderten QAA-Level für diese Ressource erfüllen (E-DZ-03).
E-DZ-06	Der STIAM-Empfänger KANN im KM festgelegen, dass die Assertions vom STIAM-Hub verschlüsselt werden müssen.	

Tabelle 3: Funktionen des STIAM-Empfängers zur Definitionszeit

Funktionen zur Laufzeit:

Funktion	Beschreibung	Kommentar
E-LZ-01	Der STIAM-Empfänger MUSS einen an den STIAM-Hub gerichteten Authentication Request mit seinem STIAM-Applikationszertifikat signieren.	Siehe Kapitel 6.1.2. Die Signatur muss das Zertifikat des STIAM-Empfängers nicht beinhalten.
E-LZ-02	Der STIAM-Empfänger MUSS die - gemäss seiner Konfiguration im KM - vom STIAM-Hub zurückgesendete Response validieren und vollumfänglich interpretieren können.	
E-LZ-03	Der STIAM-Empfänger MUSS das SAML 2.0	

Funktion	Beschreibung	Kommentar
	Web Browser SSO-Profile mit HTTP-Bindung unterstützen.	
E-LZ-04	Der STIAM-Empfänger SOLLTE SSO-Sessions unterstützen, wenn im Gesamtsystem Single Log-out (SLO) unterstützt werden soll.	
E-LZ-05	Wenn der STIAM-Empfänger die Verschlüsselung der Assertions gefordert hat, MUSS er sie entsprechend verarbeiten können.	Siehe E-DZ-06.

Tabelle 4: Funktionen des STIAM-Empfängers zur Laufzeit

2.4 STIAM-IdP

In Tabelle 5 und Tabelle 6 werden die notwendigen Funktionen eines STIAM-IdPs aufgelistet.

Funktionen zur Definitionszeit:

Funktion	Beschreibung	Kommentar
I-DZ-01	Der STIAM-IdP MUSS ein STIAM-Applikationszertifikat[3] bei einer vertrauenswürdigen CA (STIAM-CSP) beziehen und für seine SAML-Services installieren.	
I-DZ-02	Der STIAM-IdP MUSS im Komponenten-Management (KM) des STIAM-Hubs als Identity Provider registriert werden.	Siehe Kapitel 5.
I-DZ-03	Der STIAM-IdP MUSS einen QAA-Level (gemäss eCH-0170 [7]) für seinen Authentifizierungsdienst im KM des STIAM-Hubs eintragen[4].	

Tabelle 5: Funktionen des STIAM-IdPs zur Definitionszeit

Funktionen zur Laufzeit:

Funktion	Beschreibung	Kommentar
I-LZ-01	Der STIAM-IdP MUSS das SAML 2.0 Web Browser SSO-Profile mit HTTP-Bindung unterstützen.	
I-LZ-02	Der STIAM-IdP MUSS die an den STIAM-Hub gerichtete Authentication Response **und** die darin enthaltene Assertion je mit seinem STIAM-Applikationszertifikat signieren.	

[3] Ein in der STIAM-Community anerkannten Zertifikat, typischerweise ein X.509-Zertifikat, das von dem STIAM-CSP ausgestellt wurde.

[4] Der QAA-Level der STIAM-IdPs ist Ergebnis des Governance-Prozesses in der STIAM-Community. Er ist auf keinen Fall selbstdeklariert und muss von dem Community redigiert werden.

Funktion	Beschreibung	Kommentar
I-LZ-03	Der STIAM-IdP MUSS den vom Benutzer verwendeten Authentisierungsgrad (QAA-Level gemäss eCH-0170 [7]) in der Authentication Assertion angeben.	Siehe Kapitel 6.1.4.
I-LZ-04	Der STIAM-IdP MUSS dem STIAM-Hub in seiner SAML-Response als NameID immer denselben, pro Benutzer eindeutigen Identifikator zurücksenden.	Siehe Kapitel 6.1.4.
I-LZ-05	Der STIAM-IdP MUSS das Linking Protokoll zur Registrierung durch den Benutzer unterstützen.	Siehe Kapitel 3.2.1.

Tabelle 6: Funktionen des STIAM-IdPs zur Laufzeit

2.5 STIAM-Sender

In Tabelle 7 und Tabelle 8 werden die notwendigen Funktionen eines STIAM-Senders aufgelistet.

Funktionen zur Definitionszeit:

Funktion	Beschreibung	Kommentar
S-DZ-01	Der STIAM-Sender MUSS ein STIAM-Applikationszertifikat (X.509) bei einer vertrauenswürdigen CA (STIAM-CSP) beziehen und für seine SAML-Services installieren.	
S-DZ-02	Der STIAM-Sender MUSS im Komponenten-Management (KM) des STIAM-Hubs als Attribute Authority registriert werden.	Siehe Kapitel 5.
S-DZ-03	Die vom STIAM-Sender angebotenen Attribute MÜSSEN mit einer Qualitätsangabe nach eCH-0171 [8] im KM des STIAM-Hubs definiert werden. Zu jedem Attribut muss eine Mindestqualitätsstufe (QAA-Level gemäss eCH-0170 [7]) zur Identifizierung des Subjekts angegeben werden.	
S-DZ-04	Der STIAM-Sender KANN für ein Attribut zusätzlich die Authentifizierung des Benutzers bei einem bestimmten Identity Provider vorgeben.	
S-DZ-05	Der STIAM-Sender KANN im KM die Verwendung eines Attributs für einen bestimmten Kreis von Lösungsanbietern (Bereich[5]) eingrenzen.	
S-DZ-06	Der STIAM-Sender KANN im KM angeben, ob er zwecks Nachvollziehbarkeit eine Extended Attribu-	Ohne diese Option sendet der STIAM-Hub

[5] Als Bereich (Teilföderation) kann eine begrenzte Gruppe von Informationsbezügern und -lieferanten angesehen werden, welche ein bestimmtes Set an Attributen und eine gemeinsame Policy teilen (siehe Definition in eCH-0168 [4]).

Funktion	Beschreibung	Kommentar
	te Query mit einer gültigen Authentication Assertion des authentisierenden IdPs erwartet.[6]	eine Attribute Query ohne Authentication Assertion. Siehe Kapitel 4.1.1.

Tabelle 7: Funktionen des STIAM-Senders zur Definitionszeit

Funktionen zur Laufzeit:

Funktion	Beschreibung	Kommentar
S-LZ-01	Der STIAM-Sender prüft die Attribute Query vom STIAM-Hub und liefert die geforderten Attribute.	
S-LZ-02	Wenn der STIAM-Sender vom STIAM-Hub eine Extended Attribute Query erhält, MUSS er die darin enthaltene Authentication Assertion verifizieren, bevor er die geforderten Attribute ausliefert.	
S-LZ-03	Der STIAM-Sender MUSS dem STIAM-Hub in seiner SAML-Response als NameID immer denselben, pro Benutzer eindeutigen Identifikator zurücksenden.	
S-LZ-04	Der STIAM-Sender MUSS das SAML 2.0 Assertion Query/Request Profile unterstützen. Er KANN das Web Browser SSO-Profile mit HTTP-Bindung unterstützen.	
S-LZ-05	Der STIAM-Sender MUSS die an den STIAM-Hub gerichtete Attribute Response **und** die darin enthaltene Assertion je mit seinem STIAM-Applikationszertifikat signieren.	
S-LZ-06	Der STIAM-Sender MUSS Standard Attribute Queries ODER Extended Attribute Queries unterstützen.	Siehe S-DZ-06 in Tabelle 7.
S-LZ-07	Der STIAM-Sender DARF NICHT den „User Consent“ vom Benutzer einholen.	Der User Consent wird immer am STIAM-Hub eingeholt.
S-LZ-10	Der STIAM-Sender MUSS das AA-Linking Protokoll unterstützen.	Siehe Kapitel 3.2.2.
S-LZ-11	Der STIAM-Sender MUSS die vom STIAM-Hub angefragten Attribute interpretieren und auf die intern verwendeten Attribute abbilden. Er MUSS die Identitäten, zu denen die Attribute gehören, verwal-	

[6] Wenn der STIAM-Sender die original ausgestellten Authentication Assertions vom authentisierenden IdP selbst prüfen will, muss er über die Metadaten des entsprechenden IdPs verfügen. Der STIAM-Hub ist dafür verantwortlich, eine zeitlich gültige Authentication Assertion vom IdP zu besorgen und wenn notwendig den Benutzer vorgängig erneut authentifizieren zu lassen.

Funktion	Beschreibung	Kommentar
	ten.	
S-LZ-12	Der STIAM-Sender SOLLTE eine Funktionalität zum Ein-/Ausschliessen der von ihm verwalteten Identitäten an der Identity-Federation bereitstellen (User Filtering).	

Tabelle 8: Funktionen des STIAM-Senders zur Laufzeit

2.6 STIAM-Hub

In Tabelle 9 und Tabelle 10 werden die notwendigen Funktionen eines STIAM-Hubs aufgelistet.

Funktionen zur Definitionszeit:

Funktion	Beschreibung	Kommentar
H-DZ-01	Der STIAM-Hub MUSS ein STIAM-Applikationszertifikat (X.509) bei einer vertrauenswürdigen CA (STIAM-CSP) beziehen und für seine SAML-Services installieren.	
H-DZ-02	Der STIAM-Hub MUSS im eigenen Komponenten-Management (KM) mit seinen Services erfasst werden.	Siehe Kapitel 5.
H-DZ-03	Der STIAM-Hub MUSS seine eigenen Metainformationen in Form von SAML-Metadaten signiert publizieren.	
H-DZ-04	Der STIAM-Hub MUSS die Metainformationen der **registrierten** STIAM-IdPs in Form von SAML-Metadaten aggregieren und signiert publizieren.	
H-DZ-05	Der STIAM-Hub MUSS die Metainformationen von **externen** SAML-IdPs in Form von SAML-Metadaten aggregieren und signiert publizieren.	
H-DZ-06	Der STIAM-Hub MUSS einen Dienst zur Verwaltung von Benutzeraccounts zur Verfügung stellen.	
H-DZ-07	Der STIAM-Hub MUSS Linking Services zur Anbindung von STIAM-IdPs und STIAM-Sendern für das Benutzermanagement zur Verfügung stellen.	Siehe Kapitel 3.2.
H-DZ-08	Der STIAM-Hub MUSS einen Dienst zur Verwaltung der Organisationsdaten und –berechtigungen sowie deren Komponenten zur Verfügung stellen.	Siehe eCH-0168 [4], Kapitel 7.4 und 7.5.
H-DZ-09	Der STIAM-Hub MUSS einen Dienst für das Attributmanagement (Definition,	Siehe eCH-0168 [4], Kapitel 7.6.

Funktion	Beschreibung	Kommentar
	Pflege und Bekanntmachung der Attribute) zur Verfügung stellen.	
H-DZ-10	Der STIAM-Hub MUSS alle administrativen Prozesse zur Organisations- und Subjektverwaltung nachvollziehbar loggen.	

Tabelle 9: Funktionen des STIAM-Hubs zur Definitionszeit

Funktionen zur Laufzeit:

Funktion	Beschreibung	Kommentare
H-LZ-01	Der STIAM-Hub MUSS anhand des Authentication Requests den anfragenden STIAM-Empfänger und die angeforderte Ressource eindeutig identifizieren.	
H-LZ-02	Der STIAM-Hub MUSS dem Subjekt in Abhängigkeit des QAA-Levels in der Ressourcen-Definition eine Liste möglicher IdPs zur Auswahl geben, oder zum einzig möglichen IdP direkt weiterleiten (Discovery Service).	
H-LZ-03	Der STIAM-Hub MUSS das SAML 2.0 Web Browser SSO-Profile mit HTTP-Bindung und das SAML 2.0 Assertion Query/Request Profile unterstützen.	
H-LZ-04	Der STIAM-Hub MUSS zur Laufzeit für anfragende STIAM-Empfänger (RPs) die Funktion eines Authentication Proxies und Attributaggregators übernehmen.	
H-LZ-05	Der STIAM-Hub MUSS die an den STIAM-Empfänger gerichtete Authentication Response und die darin enthaltene Assertion je mit seinem STIAM-Applikationszertifikat signieren.	
H-LZ-06	Der STIAM-Hub MUSS die Assertion in einer Authentication Response verschlüsseln, wenn der STIAM-Empfänger dies fordert. Er MUSS dann auch dafür sorgen, dass die Assertions zwischen ihm und dem gewählten IdP ebenfalls verschlüsselt werden.	Siehe E-DZ-06.
H-LZ-07	Der STIAM-Hub MUSS immer das Einverständnis des Subjekts (User Consent)	

Funktion	Beschreibung	Kommentare
	zur Herausgabe von nicht öffentlichen Attributen einholen.[7]	
H-LZ-08	Der STIAM-Hub MUSS als Authentifizierungsproxy ein eigenes Session Handling für authentifizierte Benutzer unterhalten.	Siehe Kapitel 2.7.
H-LZ-09	Der STIAM-Hub MUSS anonymisierte zufällige Identifikatoren (Transient ID) gegenüber einem STIAM-Empfänger ausgeben können.	
H-LZ-10	Der STIAM-Hub MUSS anonymisierte RP-spezifische Identifikatoren (Persistent ID) gegenüber einem STIAM-Empfänger ausgeben können.	
H-LZ-11	Der STIAM-Hub MUSS Single Logout unterstützen.	
H-LZ-12	Der STIAM-Hub MUSS alle Kommunikationsvorgänge im Rahmen der Authentisierungs- und Attributvermittlung zwecks Nachvollziehbarkeit loggen.	

Tabelle 10: Funktionen des STIAM-Hubs zur Laufzeit

2.7 Session Handling

Die vom STIAM-Hub an den STIAM-Empfänger vermittelten Identitäts- und Attributinformationen werden zeitnah von den zuständigen Quellen eingeholt und vom STIAM-Hub selbst geprüft. Der STIAM-Hub legt dabei grundsätzlich keine Identitäts- oder Attributinformationen bei sich dauerhaft ab, aber er kann zur Laufzeit Identitäts- und Attributinformationen eines Subjekts zwecks Aggregation für die Dauer einer Abfrage oder zur Kontrolle der (SSO-) Sessions zwischenspeichern.

Der STIAM-Hub erstellt eine Authentication Assertion mit der gleichen Dauer wie die empfangene Authentication Assertion vom IdP und stellt ein entsprechendes Session Cookie aus.

Der STIAM-Empfänger entscheidet eigenständig, wie er mit der empfangene Authentication Assertion umgeht und wie lange seine Benutzer-Sessions daraufhin dauern.

Session Refreshing

Nach Ablauf seiner lokalen Benutzer-Session kann der STIAM-Empfänger einen erneuten Authentication Request zum STIAM-Hub senden (**Session Refreshing**). Der STIAM-Hub unterscheidet dabei mit Hilfe des Session-Cookies die folgenden Fälle:

[7] Bei der Definition der angebotenen Attribute wird festgelegt, ob ein User Consent für ein Attribut notwendig ist bzw. ob der STIAM-Hub diesen einholen muss.

1. **Keine aktive SSO-Session am Hub**
   Ist die Session am STIAM-Hub abgelaufen, präsentiert der STIAM-Hub dem Benutzer die Liste der möglichen IdPs, die für den STIAM-Empfänger definiert wurden, und fährt wie bei einem initialen Authentication Request fort.
2. **Aktive SSO-Session am STIAM-Hub**
   2.1. Aktive Session mit gleichem oder höherem QAA-Level
   Existiert am STIAM-Hub eine aktive Session mit gefordertem oder höheren QAA-Level, retourniert der STIAM-Hub eine normale Authentication Response mit einer Authentication Assertion, ohne eine erneute Authentifizierung des Benutzers. Die Authentication Assertion hat eine Gültigkeit, die der Restdauer der aktiven Session entspricht.
   2.2. Aktive Session mit zu geringem QAA-Level (Step-Up-Authentification)
   Am STIAM-Hub ist eine SSO-Session aktiv, aber der QAA-Level der bestehenden Authentifizierung ist nicht ausreichend. In diesem Fall präsentiert der STIAM-Hub dem Benutzer die Liste der möglichen IdPs, die für den STIAM-Empfänger definiert wurden, und fährt wie bei einem initialen Authentication Request fort.

Mehrere STIAM-Empfänger

Verwendet der Benutzer nun einen weiteren STIAM-Empfänger aus seinem Browser (user agent) heraus, sendet dieser STIAM-Empfänger zunächst wie gewohnt einen Authentication Request an den STIAM-Hub. Dieser kann wie beim Session Refreshing mit Hilfe seines Session-Cookies entscheiden, ob eine aktive SSO-Session mit ausreichendem QAA-Level existiert. Ist das der Fall, retourniert der STIAM-Hub eine Authentication Assertion mit eine Gültigkeit, die der Restdauer der Session entspricht.

Existiert keine aktive SSO-Session am Hub für diesen Benutzer, läuft das normale Protokoll ab und der Benutzer muss sich an einem passenden IdP authentifizieren.

Das ist auch der Fall, wenn der Benutzer einen anderen User Agent (Browser) verwendet.

Attributinformationen

Attributinformationen können gemeinsam mit den Session-Informationen am STIAM-Hub zur Laufzeit zwischengespeichert werden. Die Gültigkeit der vom STIAM-Hub ausgestellten Attribute Assertion entspricht dabei jeweils der Gültigkeit der Session mit gefordertem QAA-Level bzw. der maximalen Gültigkeit der originären Attribut Assertion des STIAM-Senders.

3 SAML-Services

Im Folgenden werden die SAML-Services beschrieben, die von den verschiedenen STIAM-Komponenten unterstützt werden müssen bzw. sollen, um die in diesem Dokument beschriebenen Protokolle (Definitions- und Laufzeit) zu implementieren.

Die SAML-Profile [9] (Web Browser SSO, Single Logout und Assertion Query/Request) definieren die folgenden Services:

- **Single Sign-On Service (SSO):** Der SSO Service beschreibt einen SAML Protokoll-Endpunkt, der einen Authentication Request entgegennimmt.

- **Assertion Consumer Service (ACS):** Der Assertion Consumer Service beschreibt einen SAML Protokoll-Endpunkt, der die Antwort `(<samlp:Response>)` auf einen Authentication Request (`<samlp:AuthnRequest>`) oder eine Attribut-Abfrage `(<samlp:AttributeQuery>)` entgegennimmt.

- **Attribute Service (AS):** Der Attribute Service beschriebt einen SAML Protokoll-Endpunkt, der eine Attribut-Abfrage `(<samlp:AttributeQuery>)` entgegennimmt.

- **Single Logout Service (SLO):** Der Single Logout Service beschreibt einen SAML Protokoll-Endpunkt, der `<samlp:LogoutRequest>` oder `<samlp:LogoutResponse>` Nachrichten entgegennimmt. Dabei kann derselbe oder verschiedene SLO Services zum Entgegennehmen der Anfragen oder Antworten verwendet werden.

Tabelle 11 zeigt in der Übersicht, welche SAML-Services von welcher STIAM-Komponente implementiert werden müssen bzw. sollten.

		SAML-Service			
		SSO	**ACS**	**AS**	**SLO**
STIAM-Komponente	**STIAM-Hub**	MUSS	MUSS	-[8]	MUSS
	STIAM-IdP	MUSS	-	-	KANN[9]
	STIAM-Sender	MUSS[10]	KANN[11]	MUSS	-
	STIAM-Empfänger	-	MUSS	-	SOLLTE

Tabelle 11: Zuordnung STIAM-Komponenten zu SAML-Services

[8] Um die Interoperabilität der Identity Federation zur erhöhen, wird auf den AS beim STIAM-Hub verzichtet. Der STIAM-Empfänger (RP) muss dann nur den Authentication Request unterstützen.

[9] Ein STIAM-IdP kann SLO unterstützen, aber er darf niemals das SLO initiieren.

[10] Der STIAM-Sender muss auch einen SSO Service anbieten, um das AA-Linking zu unterstützen.

[11] Der STIAM-Sender benötigt nur dann einen ACS, wenn er den Benutzer beim AA-Linking extern authentisieren lassen will.

3.1 Zur Laufzeit

Im Folgenden wird die Interaktion der SAML-Services und der STIAM-Komponenten zur Laufzeit beschrieben.

3.1.1 Authentifizierung mit und ohne Attributabfrage

Abbildung 2 zeigt auf, wie ein STIAM-Empfänger (Relying Party) einen Authentication Request (1) an den SSO-Service (SSO) des STIAM-Hubs sendet. Dieser agiert als Proxy und sendet einen neuen Request (2) an den SSO des authentisierenden STIAM-IdP. Dieser authentifiziert das Subjekt, generiert eine Response und sendet diese an den ACS des STIAM-Hubs zurück (3).

OHNE Attributabfrage:

Der STIAM-Hub validiert die Response vom STIAM-IdP und generiert eine neue Response, die er an den anfragenden STIAM-Empfänger (RP) zurücksendet (6). Die Schritte 4 und 5 fallen weg.

MIT Attributabfrage:

Nachdem der SSO des Identity Hubs die Identität des Subjekts kennt, kann er diese an einen STIAM-Sender (AA) übergeben, um dann eine Attributabfrage (4) an den Attributservice zu senden. Der Attributlieferant sendet die gewünschten Attributinformationen (5) an den Hub zurück, welcher diese aggregiert und an den anfragenden STIAM-Empfänger (RP) zurücksendet (6).

Auf diese Art und Weise können auch mehrere STIAM-Sender abgefragt werden. Der STIAM-Hub kann die entsprechenden Attribute Queries parallel versenden und dann die Antworten aggregieren.

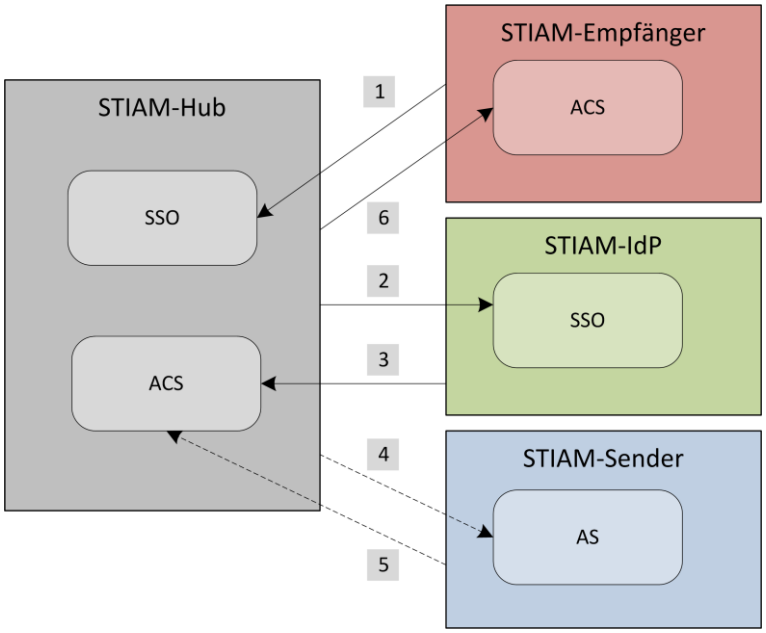

Abbildung 2: Interaktion der SAML-Services bei einem Authentication Request

Die Entscheidung, ob der Authentication Request mit oder ohne Attributabfrage durchgeführt wird, ist von der Konfiguration des STIAM-Empfängers und seinen Ressourcen in der Datenbasis (KM) des STIAM-Hubs bzw. in den Metadaten abhängig.

Da mit einem Authentication Request und den entsprechenden Konfigurationen zu den Attributen alle Anforderungen des STIAM-Empfängers abgedeckt werden können, ist es nicht notwendig, dass der STIAM-Empfänger auch *Attribute Queries* zur Attribut-Abfrage unterstützt. Sollte eine Ressource – nach einem bereits erfolgreichen Authentication Request des STIAM-Senders – noch weitere Attribute des Benutzers benötigen, kann der STIAM-Empfänger einen weiteren Authentication Request mit Angabe der Ressource abschicken. In einem solchen Fall überprüft der STIAM-Hub, ob eine erneute Authentifizierung u.U. mit einem höheren QAA-Level notwendig ist und führt diese ggf. durch. Die funktionalen Anforderungen an den STIAM-Empfänger werden dadurch stark vereinfacht und die Interoperabilität des Gesamtsystems erhöht.

Relay-State

In SAML 2.0 wird vom STIAM-Empfänger ein Relay-State-Parameter verwendet, um die ursprüngliche Anwendungs-URL, welche vom Benutzer aufgerufen wurde, wiederherstellen zu können. Bei einem von der Relying Party initiierten Web-SSO-Protokollablauf legt dieser in der Regel die aufgerufene URL ab und setzt den Namen des Cookies in den Relay-State seiner Anfrage beim IdP.

Folgendes Verhalten muss der STIAM-Hub unterstützen: Wenn in einem Authentication Request eines STIAM-Empfängers der Relay-State Parameter verwendet wird, so muss der STIAM-Hub diesen Parameter in seiner Response wieder zurücksenden.

3.1.2 Single Logout

Single Logout (SLO) ermöglicht es einem Benutzer in einer STIAM-Domäne, alle für ihn erstellten Sessions mit einem Klick zu beenden. Für STIAM ist diese Funktionalität ein wichtiger zusätzlicher Dienst, da ein Benutzer über den STIAM-Hub einfach mehrere Sessions zu unterschiedlichen STIAM-Empfängern erstellen kann. SLO hilft ihm dabei, diese Sessions einfach zu beenden.

Der Benutzer kann das SLO entweder von einem STIAM-Empfänger[12] aus (in Abbildung 3 STIAM-Empfänger 1) oder vom STIAM-Hub aus initiieren.

Im ersten Fall sendet der initiierende STIAM-Empfänger einen Logout-Request (1) an den STIAM-Hub, nachdem er die lokale Benutzer-Session beendet hat. Der Hub identifiziert alle zu diesem Benutzer gehörenden Sessions bei anderen STIAM-Empfängern (in Abbildung 3 STIAM-Empfänger 2). Diese bekommen parallel einen Logout-Request (2) zugestellt, den sie nach Beenden der Benutzer-Sessions mit einer Logout-Response (3) beantworten müssen. Das Single-Logout ist abgeschlossen, wenn der STIAM-Hub dem initiierenden STIAM-Empfänger eine Logout-Response (4) schickt.

Beim STIAM-Hub-initiierten Single Logout fallen die Schritte 1 und 4 weg.

Falls gewünscht und wenn vom STIAM-IdP unterstützt, kann auch ein Logout-Request an einen STIAM-IdP gesendet werden, um die dort aktive Session zu beenden. Dabei muss

[12] Die Applikation muss dazu dem Benutzer ein entsprechendes Bedienelement zur Verfügung stellen und ihn darüber informieren, dass auch Sessions in anderen Applikationen beendet werden.

aber sichergestellt werden, dass dieser IdP nicht von anderen STIAM-Communities oder externen Relying-Parties verwendet wird.

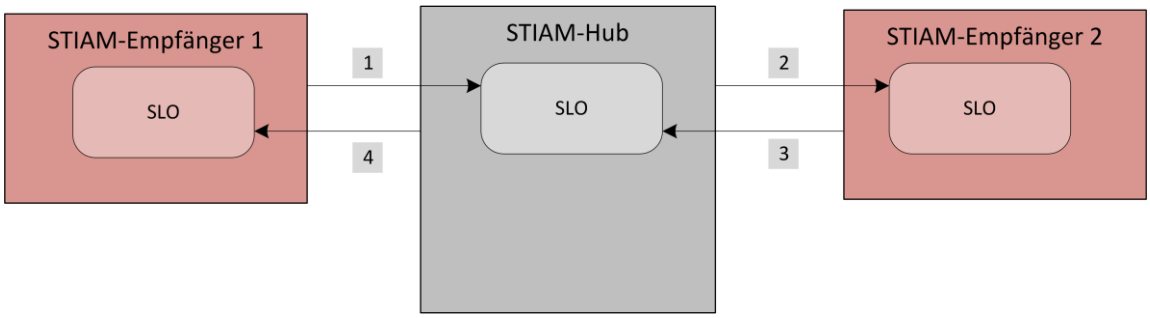

Abbildung 3: Interaktion der SAML-Services beim Single Logout

3.2 Zur Definitionszeit

Im Folgenden wird die Interaktion der SAML-Services der STIAM-Komponenten zur Definitionszeit beschrieben.

3.2.1 IdP-Linking

Beim IdP-Linking kann ein Benutzer interaktiv einen STIAM-IdP in seiner Link Table erfassen (siehe Abbildung 4).

Der STIAM-Hub gibt auf seiner Webseite einem Benutzer die Möglichkeit, einen möglichen IdP aus einer Liste zu wählen und diesen zu verlinken. Der Benutzer wird vom STIAM-Hub auf den SSO-Service des STIAM-IdP weitergeleitet (1). Der Benutzer wird von diesem durch die ihm bekannte und lokal übliche Methode authentifiziert. Der STIAM-IdP gibt in seiner Assertion einen eindeutigen, nicht-transienten Identifikator des Benutzers an den STIAM-Hub zurück (2). Damit kann der STIAM-Hub den erhaltenen Identifikator in der Link Table des Benutzers ablegen.

Zur Unterstützung des IdP-Linking-Protokolls muss ein STIAM-IdP keine zusätzlichen Funktionalitäten implementieren, da ein normaler Authentication Request verwendet wird (siehe IdP-Linking-Protokoll in Kapitel 4.2.1, sowie das Beispiel eines Authentication Requests in Kapitel 6.1.2).

Beim Erstellen und der Pflege der Identity Links muss der Hub sicherstellen, dass der Benutzer keine Veränderung der Link Table vornehmen kann, ohne sich zuvor dem von ihm verlinkten IdP mit dem höchsten QAA-Level authentisiert zu haben.[13]

[13] Wenn der Benutzer Änderungen in seiner Link Table vornehmen will, so muss er sich zuvor bei einem von ihm verlinkten IdP mit höchstem QAA-Level authentisieren. Optional kann zusätzlich noch eine Bestätigungs-Email oder -SMS vom Hub eingefordert werden. Kann sich ein Benutzer so nicht erfolgreich authentisieren, bleibt nur der Weg über die Hotline.

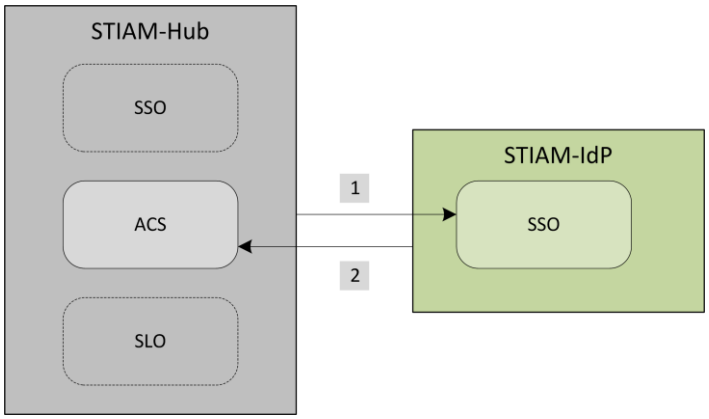

Abbildung 4: Interaktion der SAML-Services bei einem IdP-Linking

3.2.2 AA-Linking

Beim AA-Linking kann ein Benutzer interaktiv einen STIAM-Sender (AA) in seiner Link Table erfassen (siehe Abbildung 5). Der STIAM-Hub gibt auf seiner Webseite einem Benutzer die Möglichkeit, eine mögliche AA aus einer Liste zu wählen und diese zu verlinken.

Bei der Verlinkung gibt es zwei prinzipiell verschiedene Möglichkeiten:

- Verlinkung mittels Authentifizierung

und

- Verlinkung über identifizierendes Attribut.

3.2.2.1 Verlinkung mittels Authentifizierung

Der STIAM-Hub sendet dazu einen *Authentication Request* zum STIAM-Sender (1). Der STIAM-Sender erfordert eine Authentifizierung des Benutzers.

Variante 1 *– Interner lokaler IdP*

Der Benutzer wird vom internen IdP der Attribute Authority (AA) durch die ihm bekannte und lokal übliche Methode authentifiziert.

Variante 2 *– Externer STIAM-IdP*

Der STIAM-Sender leitet den Authentication Request über den Browser des Benutzers an einen geeigneten IdP (2) weiter. Der Benutzer wird vom gewählten IdP durch die ihm bekannte und lokal übliche Methode authentifiziert. Der IdP retourniert eine digital signierte Response Message (3), welche die Identität des Benutzers bestätigt.

Nach erfolgreicher Authentifizierung bestimmt der STIAM-Sender den lokalen Identifikator des Benutzers und übergibt diesen signiert und mit einem Zeitstempel versehen in einer Authentication Assertion über den Browser des Benutzers dem STIAM-Hub (4). Damit kann der STIAM-Hub den erhaltenen Identifikator in der Link Table des Benutzers ablegen.

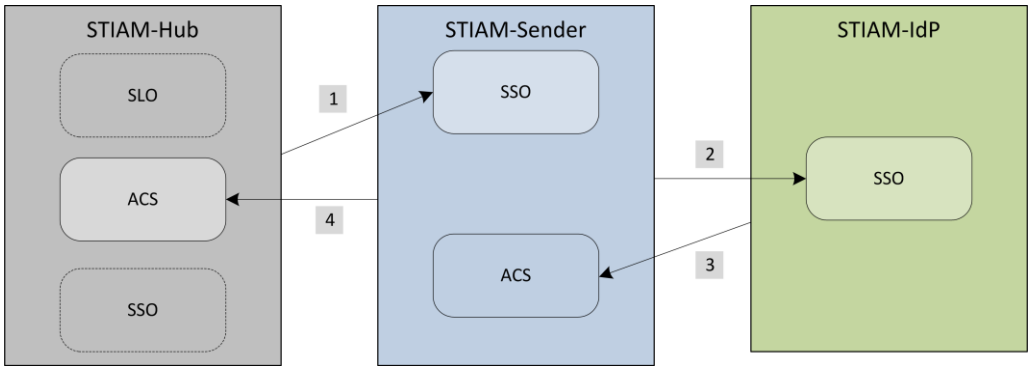

Abbildung 5: Interaktion der SAML-Services bei einem AA-Linking mittels Authentifizierung

Variante 3 *– Externer STIAM-IdP über Hub*

Der STIAM-Sender leitet den Authentication Request über den Browser des Benutzers an den STIAM-Hub (2) weiter. Der Hub leitet den Authentication Request zu einem geeigneten IdP weiter (3) und hat so Kontrolle über den zum Linking verwendeten IdP. Der Benutzer wird vom gewählten IdP durch die ihm bekannte und lokal übliche Methode authentifiziert. Der IdP retourniert eine digital signierte Response Message (4) an den Hub, der diese an den STIAM-Sender weitergibt (5).

Nach erfolgreicher Authentifizierung bestimmt der STIAM-Sender den lokalen Identifikator des Benutzers und übergibt diesen signiert und mit einem Zeitstempel versehen in einer Authentication Assertion über den Browser des Benutzers dem STIAM-Hub (6). Damit kann der STIAM-Hub den erhaltenen Identifikator in der Link Table des Benutzers ablegen.

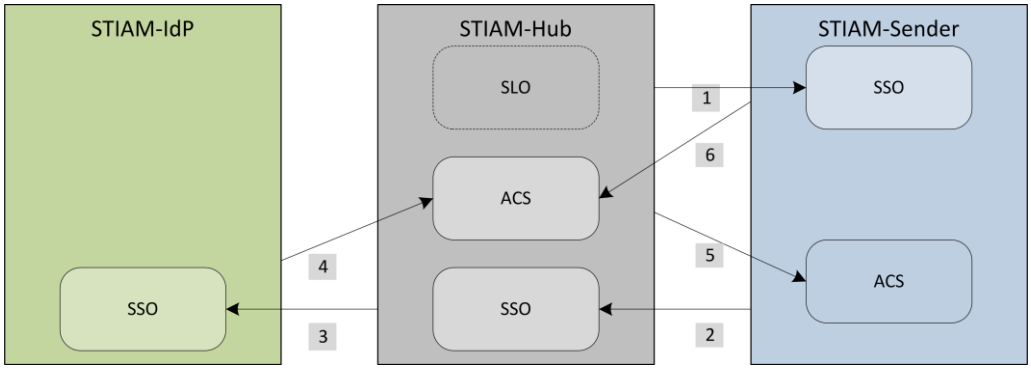

Abbildung 6: Interaktion der SAML-Services bei einem AA-Linking mittels Authentifizierung über den Hub

Die Anforderungen an eine Authentifizierung des Benutzers mit höchstmöglichem QAA-Level gilt auch beim Erstellen und der Pflege von Attribute Links. Der Hub muss sicherstellen, dass sich der Benutzer zuvor mit seinem höchsten verlinkten QAA-Level authentisiert hat, bevor er die Anbindung oder Pflege von AA-Links zulässt.

3.2.2.2 Verlinkung über identifizierendes Attribut

Bei einem STIAM-Sender der über ein identifizierendes Attribut, d.h. über einen extern definierten Identifikator, angebunden wird, MUSS im Komponenten-Management angegeben

werden, welches Attribut mit welcher Qualität zur Verlinkung verwendet werden soll. Es kann auch direkt ein STIAM-Sender angegeben werden, der dieses Attribut zur Verfügung stellt.

Die Verlinkungen läuft dann in den folgenden Schritten ab:

Schritt	Beschreibung
1	• Falls kein STIAM-Sender, der das Attribut bereitstellt, vorgegeben wurde, evaluiert der STIAM-Hub, welche STIAM-Sender das Attribut in der gewünschten Qualität zur Verfügung stellen und einen AA-Link für den Benutzer besitzen (bei mehreren Möglichkeiten muss der Benutzer wählen).
2	• Der STIAM-Hub überprüft, ob der Benutzer mit einer genügend hohen Qualität authentifiziert wurde und führt bei Bedarf eine Step-Up-Authentifizierung durch.
3	• Der STIAM-Hub sendet eine Attribut-Query an den gewählten STIAM-Sender, um das Attribut abzuholen. War das erfolgreich, kann der STIAM-Hub dieses Attribut als Identifikator für den neuen STIAM-Sender in der AA-Linking-Tabelle ablegen.
4	• Optional kann die Anbindung des neuen STIAM-Senders getestet werden.

3.2.3 RP-Linking

In der Regel verfügt eine Applikation vor der Integration in STIAM über ein lokales Identity Management, in welchem Benutzername, Credentials und Attribute zu den Identitäten gespeichert sind. Eine Relying Party kann die lokal gespeicherten Informationen zu einem Benutzer auch nach einer Integration in STIAM verwenden und z.B. nur die Authentifizierung des Benutzers an den STIAM-Hub delegieren. Dazu müssen folgende Bedingungen erfüllt sein:

- Der STIAM-Hub muss der RP bei jeder Response denselben Identifier als *NameID* zusenden, damit diese ein Identity-Mapping durchführen kann. Dazu muss in der Konfiguration des STIAM-Empfängers auf dem STIAM-Hub das *NameID*-Format auf **persistent** gesetzt werden können.
- Die Relying Party muss die lokalen Identitäten mit den persistenten STIAM-Identitäten verbinden können (Identity Mapping).

Das Verlinken dieser Identitäten auf Seiten RP kann auf verschiedene Art und Weise erfolgen. In der Folge wird eine Möglichkeit eines Protokollablaufs beschrieben, wie über das in Abbildung 8 dargestellte Authentisierungs-Protokoll mit Attribut-Aggregation ein Identity Linking zwischen STIAM-Empfänger und STIAM-Hub, basierend auf einem RelayState Parameter und einem zweifachen Login des Benutzers, realisiert werden kann.

Schritt	Beschreibung
5	• Der Benutzer loggt sich wie üblich auf der Webseite der Applikation der Relying Party mit seinen lokalen Credentials ein.
1	• Der STIAM-Empfänger leitet den Benutzer an den STIAM-Hub zur

Schritt	Beschreibung
	Anbindung der STIAM-Identität weiter. Der STIAM-Empfänger gibt im RelayState die lokale SessionID des Benutzers an den Hub mit.
2	• Der Benutzer muss sich über den STIAM-Hub gegenüber einem adäquaten IdP authentisieren.
3	• Der STIAM-Hub berechnet die Persistent-ID des Benutzers für diese Relying Party und sendet eine Authentication Response an den STIAM-Empfänger zurück. Der Hub legt die generierte Persistent-ID mit der RP-ID in der Link Table des Benutzers ab.
4	• Der STIAM-Empfänger identifiziert die Response anhand des RelayState-Parameters und verlinkt die Persistent-ID mit der lokalen ID des Benutzers.
5	• Von nun an kann die Relying Party den Benutzer durch den STIAM-Hub authentifizieren lassen und anhand der Persistent-ID identifizieren.

Alternativ kann das Linking anhand eines redundant vorhandenen Attributwerts vollzogen werden. In diesem Fall muss sich der Benutzer nicht vorgängig lokal bei der Applikation authentisieren, dafür muss im KM der Relying Party ein Attribut definiert werden, welches der Hub für den Identifizierungsprozess mitliefern muss. Anhand des Attributwerts kann die Relying Party den Benutzer dann lokal identifizieren und das Identity-Mapping vornehmen.

4 Protokolle

4.1 Zur Laufzeit

4.1.1 Authentifizierung ohne Attributabfrage

Im Folgenden wird das Protokoll zur Authentifizierung **OHNE** Attributabfrage des Subjektes bei dem STIAM-Empfänger beschrieben. Es basiert auf dem SAML Web Browser SSO Profile mit HTTP POST Binding. Die verwendeten Messages sind in Kapitel 6.1.2 detailliert beschrieben und durch Beispiele ergänzt.

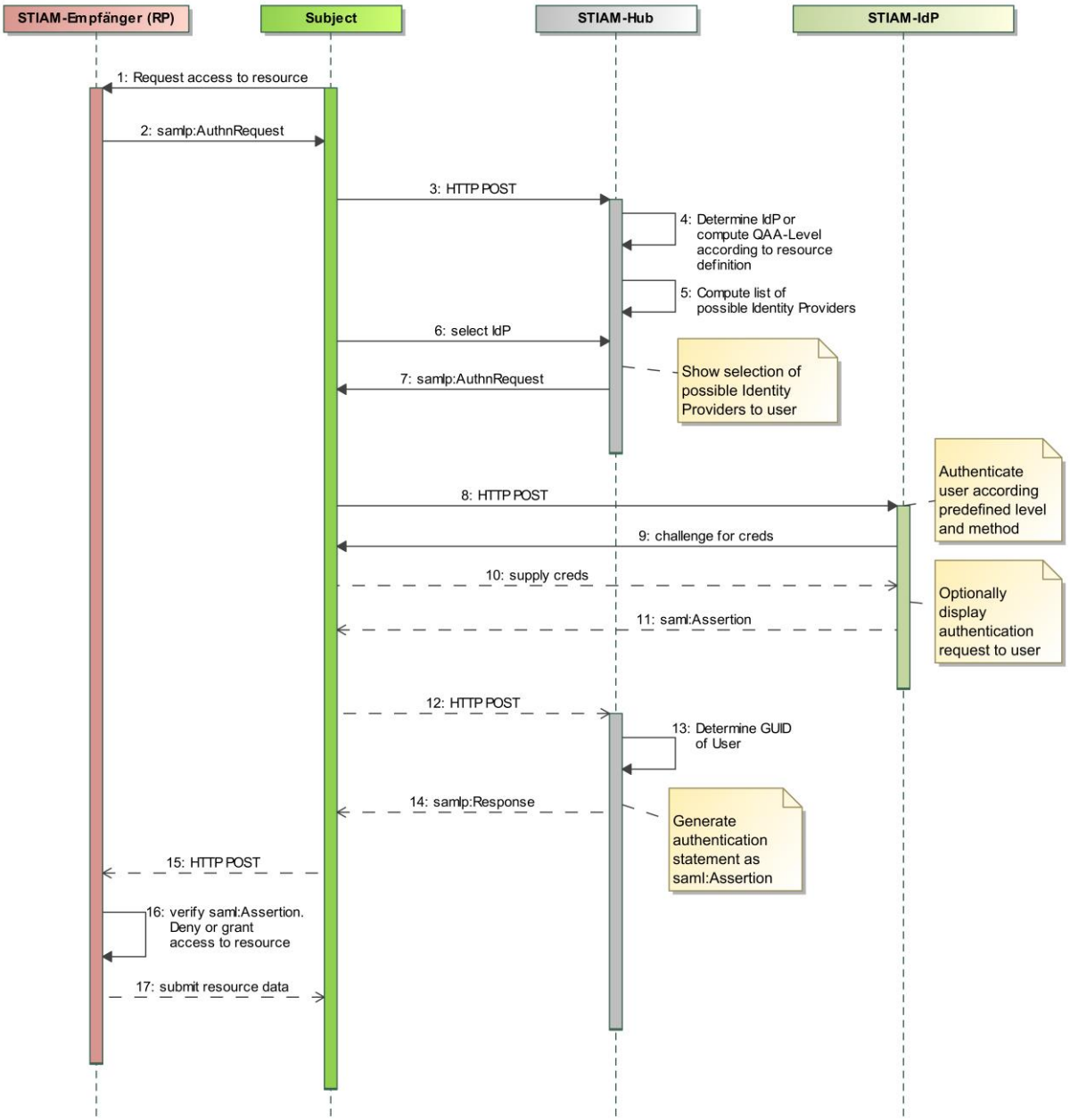

Abbildung 7: Authentisierungs-Protokoll

Beschreibung zu Abbildung 7:

Schritt	Bemerkung
1	• Der Benutzer möchte auf eine Ressource zugreifen.
2	• Der STIAM-Empfänger (RP) sendet einen digital signierten `<samlp:AuthnRequest>` mit Angaben zur Ressource, auf die der Benutzer zugreifen will, zum Browser des Benutzers (Subject). Die Ressource im `<samlp:AuthnRequest>` wird über den `<samlp:AttributeConsumingServiceIndex>` (siehe Kapitel 6.1.2) identifiziert. Wird die Ressource nicht angegeben, gilt der Default[14]. • Der STIAM-Empfänger sendet den `<samlp:AuthnRequest>` in einer self-submitting HTML Form zurück an den Browser des Benutzers.
3	• Der Browser sendet den `<samlp:AuthnRequest>` zum `<md:SingleSignOnService>` des STIAM-Hubs als HTTP POST-Nachricht.
4	• Der STIAM-Hub ermittelt die Relying Party (RP-ID) des Absenders anhand des `<saml:Issuer>` Elements (welches im Request enthalten ist) in seiner Datenbasis (`<md:entityID>` Eintrag). • Der Hub verifiziert die Signatur des `<samlp:AuthnRequest>` mittels X.509 Zertifikat des STIAM-Empfängers aus der Datenbasis.[15] • Der Hub ermittelt anhand des `<samlp:AttributeConsumingServiceIndex>` die Ressource (RES-ID) in der Datenbasis.
5	• Der STIAM-Hub stellt anhand der *Ressourcen Definition* in seiner Datenbasis eine Liste von möglichen Authentifizierungsdiensten (IdPs) zusammen. Dabei sind je nach Ressourcen Definition folgende Szenarien möglich: a) Die Ressource hat einen (oder mehrere) IdPs (IdP-ID) vorgegeben, die den definierten QAA-Level erfüllen. Der STIAM-Hub zeigt auf seiner Webseite dem Benutzer die vordefinierten IdPs zur Wahl an oder leitet ihn direkt zum einzig möglichen IdP weiter. b) Die Ressource hat einen erwünschten Authentifizierungslevel (QAA-Level) vorgegeben. Anhand dieses Werts erstellt der STIAM-Hub eine Liste der in Frage kommenden IdPs und gibt diese dem Benutzer zur Auswahl. Es ist möglich, weitere allgemeine Regeln zur IdP-Auswahl auf dem STIAM-Hub zu definieren, z.B. basierend auf IP-Range oder

[14] Damit ein Authentication Request ohne Attribut-Abfrage durchgeführt werden kann, muss die entsprechende Ressource ohne die Angabe von angeforderten Attributen im Komponentenmanagement konfiguriert sein (entspr. einem `md:AttributeConsumingService` ohne Attribute in den Metadaten des STIAM-Empfängers).

[15] Schlägt die Verifizierung des STIAM-Empfängers fehl, so sendet der STIAM-Hub eine Authentication Response mit entsprechendem Fehler im `<samlp:Status>` zurück.

Schritt	Bemerkung
	User Agent.
6	• Der Benutzer wählt anhand der (in Punkt 5 evaluierten) zur Auswahl stehenden Authentifizierungsdienste einen Identity Provider aus.
7	• Der STIAM-Hub erstellt einen `<samlp:AuthnRequest>` zuhanden des ausgewählten Identity Providers und sendet diesen in einer HTML Form zurück zum Browser.
8	• Der Browser sendet eine HTTP POST-Nachricht mit dem `<samlp:AuthnRequest>` zum SSO-Service des IdPs.
9	• Der IdP fordert den Benutzer zur Authentisierung auf (Username/Password, PKI basierend, 2FA, etc.).
10	• Der Benutzer authentisiert sich gegenüber dem Identity Provider.
11	• Der IdP authentifiziert den Benutzer. • Wenn der IdP das Einverständnis vom Benutzer selbst einholt, wird dies in der IdP-Konfiguration des KM vermerkt und der Hub holt kein weiteres Einverständnis ein. • Dann erzeugt der IdP eine `<samlp:Response>`, welche ein `<saml:AuthnStatement>` in einer `<saml:Assertion>` enthält. • Der IdP muss die `<samlp:Response>` und die `<saml:Assertion>` digital signieren. • Abschliessend sendet der IdP seine Response an den Browser des Benutzers.
12	• Der Browser leitet die `<samlp:Response>` des IdPs in einer HTTP POST-Nachricht zum STIAM-Hub weiter.
13	• Der STIAM-Hub kann anhand der `<saml:NameID>`, welche mit dem IdP-Identifier der Link Table im User Identifier Repository (UIR) übereinstimmen muss, die GUID des Users bestimmen. Ist der IdP-Identifier mit mehr als einer GUID verbunden, so muss der Hub dem User übereinstimmende Accounts zur Auswahl vorlegen.[16] • Aus der GUID wird je nach Konfiguration des STIAM-Empfängers eine Persistent ID oder Transient ID generiert.
14	• Der STIAM-Hub erzeugt eine `<samlp:Response>` mit einer `<saml:Assertion>`, die ein `<saml:AuthnStatement>` enthält. • Der STIAM-Hub muss die `<samlp:Response>` und die `<saml:Assertion>` digital signieren. • Abschliessend sendet der Hub die `<samlp:Response>` an den Browser des Benutzers zurück.

[16] Es soll möglich sein, dass ein User mehrere Accounts in SuisseTrustIAM führen kann und in diesen Accounts denselben Identifier verwenden kann (z.B. in zwei Accounts ein und dieselbe SuisseID mit einer SuisseID-Nr. als Identifier).

Schritt	Bemerkung
15	• Der Browser leitet die `<samlp:Response>` an den STIAM-Empfänger (RP) weiter.
16	• Der STIAM-Empfänger (RP) muss die Signaturen der `<samlp:Response>` und der `<saml:Assertion>` verifizieren und kann die Antwort des STIAM-Hubs für den Zugriffsentscheid auf die Ressource verwenden.
17	• Die STIAM-Empfänger (RP) verwehrt oder gewährt den Zugriff auf die Ressource anhand lokal definierten Policy.

Tabelle 12: Protokollschritte Authentisierung

4.1.2 Authentifizierung mit Attribut-Abfrage

In diesem Abschnitt wird das Protokoll zur Authentifizierung **MIT** Attributabfrage des Subjektes durch die Relying Party beschrieben. Dieses Protokoll ähnelt weitgehend der Variante OHNE Attributabfrage, wobei einige Schritte neu hinzukommen.

Das hier beschriebene Protokoll mit Attributabfrage kann in zwei Varianten durchgeführt werden. Die beiden Varianten unterscheiden sich nicht im Protokollablauf, sondern im Inhalt der `<samlp:AttributeQuery>`, welche der STIAM-Hub dem STIAM-Sender zusendet.

Variante 1 - Standard Attribute Query: Der STIAM-Hub sendet eine Standard Attribute Query an den STIAM-Sender (siehe Listing 6). Dieser vertraut der erfolgten Authentifizierung des Benutzers durch den Hub und liefert diesem die angeforderten Attribute aus. Diese Variante setzt ein entsprechendes Vertrauen zwischen STIAM-Sender und STIAM-Hub voraus, ist dafür aber performanter und von Seiten STIAM-Sender einfacher zu handhaben.

Variante 2 - Extended Attribute Query: Der System Administrator des STIAM-Senders kann im Komponenten-Management angeben, ob er jeweils die Prüfung einer Authentication Assertion selbst durchführen will. Dies verpflichtet den STIAM-Hub dazu, die `<saml:Assertion>` des authentifizierenden IdP an den STIAM-Sender weiterzugeben. Dies kann durch eine Extended Attribute Query[17] erfolgen, welche im `<samlp:Extensions>` Element die vom IdP original ausgestellte Assertion einschliesst (siehe Listing 7). In diesem Fall kann der STIAM-Sender den „Logon Security Context“ des Benutzers selbst prüfen, bevor er mit einer `<samlp:AttributeResponse>` die angeforderten Attribute ausstellt. Dies bedingt aber, dass der STIAM-Sender über alle notwendigen Metadateninformationen des authentifizierenden STIAM-IdP verfügt.

Das beschriebene Protokoll basiert auf dem SAML Web Browser SSO Profile mit HTTP Binding [9] für die Authentifizierung und auf dem SAML Assertion Query/Request Profile [9] mit SOAP Binding entsprechen den Empfehlungen aus den SAML 2.0 Conformance Requirements [10]. Die Attribut-Abfrage beim STIAM-Sender entspricht damit einer Backchannel-Abfrage, die ausser TLS-Absicherung keine weitere Verschlüsselung erfordert.

[17] Eine ebenfalls mögliche Alternative ist die Verwendung von WS-Security und die Übertragung der Authentication Assertion in Form eines Security Tokens im WSS Header.

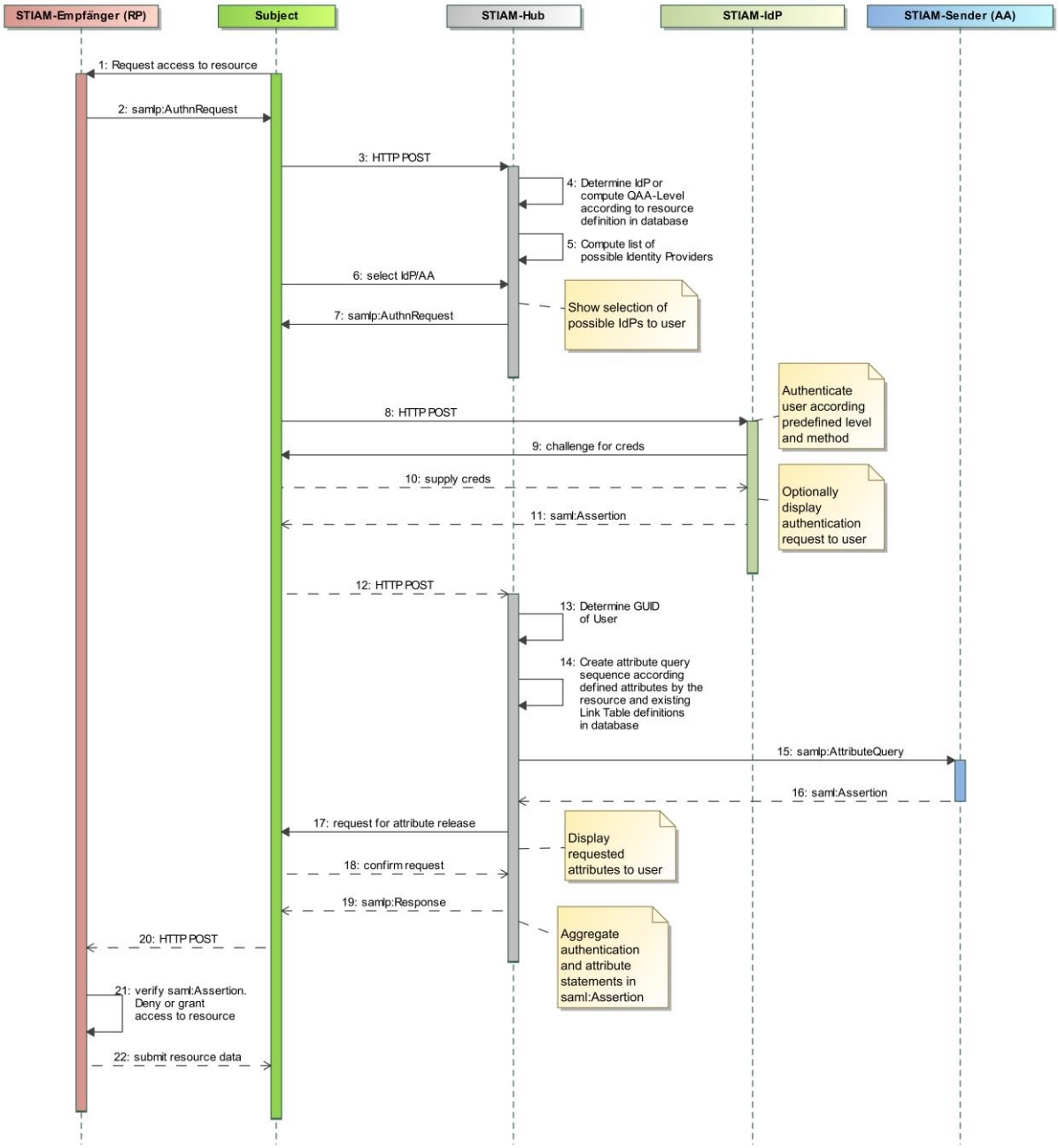

Abbildung 8: Authentisierungs-Protokoll mit Attribut-Aggregation

Beschreibung zu Abbildung 8 (auf die Schritte, die identisch zu denen in Abbildung 7 sind, wird verzichtet):

Schritt	Bemerkung
1 - 4	Bleiben identisch
5	• Der STIAM-Hub stellt anhand der *Ressourcen Definition* in seiner Datenbasis eine Liste von möglichen Authentifizierungsdiensten (IdPs) zusammen. Dabei sind je nach Ressourcen Definition folgende Szenarien möglich: a) Die Ressource hat einen (oder mehrere) IdPs (IdP-ID) vorgegeben, die den definierten QAA-Level erfüllen. Der STIAM-Hub zeigt auf seiner Webseite dem Benutzer die vordefinierten

Schritt	Bemerkung
	IdPs zur Wahl an oder leitet ihn direkt zum einzig möglichen IdP weiter. b) Die Ressource hat einen erwünschten Authentifizierungslevel (QAA-Level) vorgegeben. Anhand dieses Werts erstellt der STIAM-Hub eine Liste der in Frage kommenden IdPs und gibt diese dem Benutzer zur Auswahl. c) Die Ressource hat ein (oder mehrere) angebotene(s) Attribut(e) (*Attr-ID*) definiert. Der STIAM-Hub bestimmt anhand der *Attr-ID* den darin verlangten QAA-Level. Anhand des höchsten QAA-Levels aller angeforderten Attribute zeigt der STIAM-Hub dem User eine Liste der in Frage kommenden IdPs an. Optional können in der *Attr-ID* auch direkt ein oder mehrere Identity Provider (*IdP-ID*) vorgegeben sein. In diesem Fall verfährt der Hub wie in Punkt a). Es ist möglich, auf dem STIAM-Hub weitere allgemeine Regeln zur IdP-Auswahl zu definieren, z.B. basierend auf IP-Range oder User Agent.
6-13	Bleiben identisch
	• Der STIAM-Hub stellt in Abhängigkeit der in der Ressource geforderten Attribute eine *Attribute Query Sequenz*[18] zusammen. Dabei muss er folgende Kriterien berücksichtigen: a) Die Ressource hat ein (oder mehrere) angebotene(s) Attribut(e) (*Attr-ID*) definiert. Der STIAM-Hub bestimmt anhand der *Attr-ID* die *AA-ID* und überprüft, ob der Benutzer in seiner Link Table über einen entsprechenden AA-Link verfügt. b) Die Ressource hat ein (oder mehrere) Attribut(e) mit einer Attributqualität (*Attr-Quality*) definiert. Der STIAM-Hub sucht in seiner Datenbasis nach Attribut-Autoritäten (*AA-ID*), welche das Attribut in der geforderten Qualität zur Verfügung stellen können und überprüft, ob der Benutzer in seiner Link Table über einen entsprechenden AA-Link verfügt. c) Die Ressource hat nur ein (oder mehrere) Attribut(e) (*OID*) definiert. Der STIAM-Hub sucht in der Datenbasis nach Attribut-Autoritäten (*AA-ID*), welche das Attribut anbieten und überprüft, ob der Benutzer in seiner Link Table über den entsprechenden AA-Link verfügt. • Falls der Benutzer nicht über einen notwendigen AA-Link verfügt und das in der Ressource Definition angegebene Attribut ist als *required* bezeichnet, muss der STIAM-Hub dem Benutzer eine entsprechende Fehlermeldung anzeigen. Hat der Benutzer mehrere AA-Links zu einem Attribut definiert, so

[18] Attribute Query Sequenz: Die Liste der AAs, welche Lieferanten der Attribute in geforderter Qualität sind.

Schritt	Bemerkung
	muss der Hub dem Benutzer die in Frage kommenden Attribut-Autoritäten zur Auswahl anzeigen.
15	• Der STIAM-Hub sendet einen digital signierten *Attribute Request* an den Attribute Service (AS) des STIAM-Senders (AA). • Dieser Request beinhaltet die angeforderten Attribute, die *NameID* des Benutzers und optional die Authentication Assertion des IdPs.
16	• Der STIAM-Sender (AA) überprüft den Status des Benutzers. Ist der Benutzer gültig und aktiv, stellt er eine `<saml:Assertion>` zusammen, welche ein `<saml:AttributeStatement>` mit `<saml:Attribute>` und `<saml:AttributeValue>` Elementen beinhaltet und signiert diese digital. In allen anderen Fällen enthält die Antwort des STIAM-Senders einen entsprechenden Status mit Fehlercode (siehe [11]), z.B. `urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal` für einen deaktivierten Benutzer.
17	• Der STIAM-Hub holt sich vom Benutzer für die angeforderten persönlichen Attribute die Freigabe zur Weitergabe (user consent)[19] an den STIAM-Empfänger.
18	• Der Benutzer bestätigt die Freigabe der Attribute.
19	• Der STIAM-Hub stellt eine optional verschlüsselte `<saml:Assertion>` mit einem `<saml:AuthnStatement>` und einem `<saml:AttributeStatement>` zusammen und sendet diese als `<samlp:Response>` an den Browser zurück. • Der STIAM-Hub muss diese `<saml:Assertion>` digital signieren.
20-22	Bleiben identisch

Tabelle 13: Protokollschritte Authentifizierung mit Attributabfrage

4.1.3 Single Logout

In der Abbildung 9 ist das SLO-Protokoll dargestellt. Ein Benutzer, der auf dem STIAM-Empfänger (RP1) arbeitet, entscheidet seine SSO Session zu beenden (1). Der STIAM-Empfänger (RP1) beendet die lokalen Sessions des Benutzers (2) und sendet dem STIAM-Hub eine `<samlp:LogoutRequest>` Message (3).

Der STIAM-Hub identifiziert anhand der *NameID* und *SessionIndex* alle zu diesem Benutzer gehörenden Sessions bei anderen STIAM-Empfängern (4) und sendet diesen, im Beispiel dem STIAM-Empfänger (RP2), parallel eine `<samlp:LogoutRequest>` Message (5).

Der STIAM-Empfänger (RP2) beendet die lokalen Sessions des Benutzers ebenfalls (6) und sendet danach eine entsprechende `<samlp:LogoutResponse>` Message zum STIAM-Hub (7) zurück.

[19] Der User Consent wird nur einmal pro Ressource bzw. bei Änderungen des Attribut-Sets eingeholt und dann auf dem STIAM-Hub abgelegt.

Wenn der STIAM-Hub alle `<samlp:LogoutResponse>` Messages der betroffenen STIAM-Empfänger erhalten hat, terminiert er die lokalen Benutzer-Sessions (8). Anschliessend sendet er eine `<samlp:LogoutResponse>` Message (9) zum STIAM-Empfänger (RP1), der das Logout initiiert hat, und dieser informiert den Benutzer, dass seine SSO-Session beendet wurde (10).

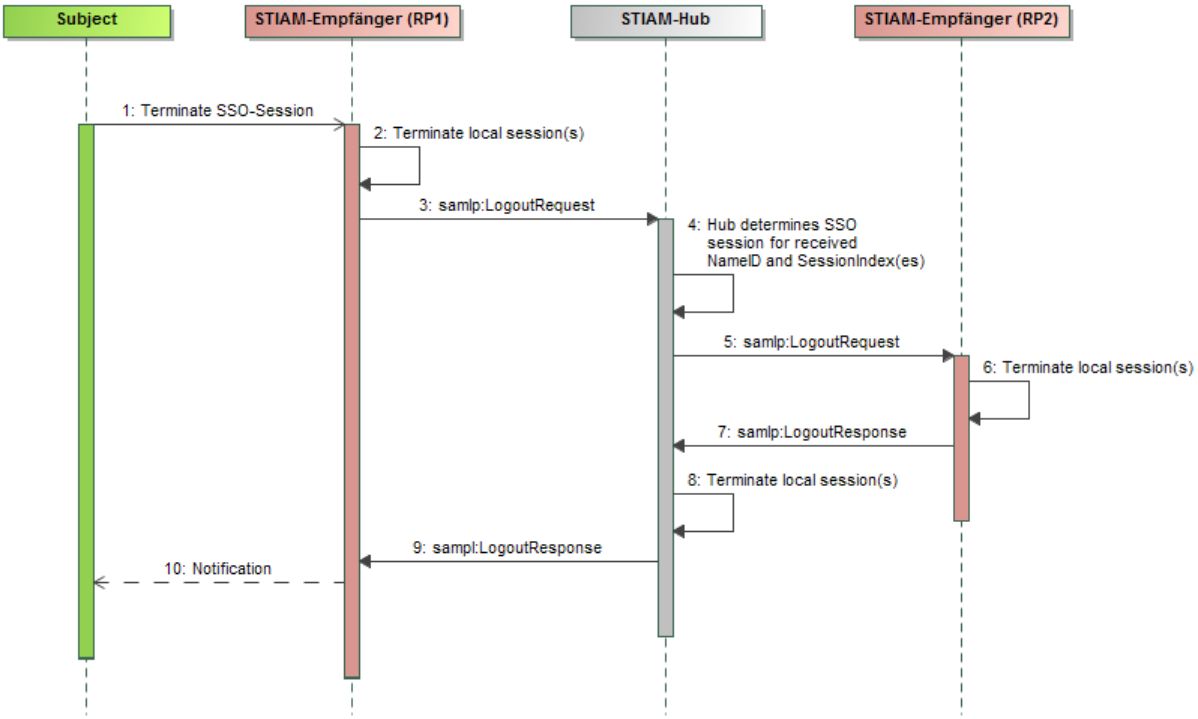

Abbildung 9: SLO-Protokoll

Schritt	Bemerkung
1	• Ein Benutzer auf dem STIAM-Empfänger (RP1) entscheidet, seine SSO Session zu beenden.
2	• Der STIAM-Empfänger (RP1) beendet die lokalen Sessions des Benutzers.
3	• Der STIAM-Empfänger (RP1) sendet eine digital signierte `<samlp:LogoutRequest>` Message zum STIAM-Hub (siehe Beispiel in Kapitel 6.1.6).
4	• Der STIAM-Hub identifiziert anhand der <NameID> alle zu diesem Benutzer gehörenden Sessions bei anderen STIAM-Empfängern.
5	• Der STIAM-Hub sendet dem STIAM-Empfänger (RP2) eine digital signierte `<samlp:LogoutRequest>` Message (ähnlich wie diejenige, die er erhalten hat).
6	• Der STIAM-Empfänger (RP2) beendet die lokalen Sessions des Benutzers anhand der Informationen im `<NameID>` und im `<SessionIndex>` Element.
7	• Der STIAM-Empfänger (RP2) sendet eine digital signierte `<samlp:LogoutResponse>` Message zum STIAM-Hub. Diese

Schritt	Bemerkung
	Message enthält ein `<StatusCode>` Element mit Informationen über den Status der SLO-Abfrage (siehe Beispiel in Kapitel 6.1.7).
8	• Der STIAM-Hub terminiert die lokalen Benutzer-Sessions.
9	• Der STIAM-Hub sendet dem STIAM-Empfänger (RP1) eine digital signierte `<samlp:LogoutResponse>` Message mit Informationen über den Status (`<StatusCode>` Element) der beim STIAM-Empfänger (RP1) ursprünglich erstellten SLO-Abfrage.
10	• Schlussendlich informiert der STIAM-Empfänger (RP1) den Benutzer, dass seine SSO-Session beendet wurde und er sich von der STIAM-Community (STIAM-Hub, RP, RP2) abgemeldet hat.

Tabelle 14: Protokollschritte Single Logout

Da das SAML-SLO-Standard-Protokoll nicht von allen STIAM-Empfängern unterstützt werden kann und vor allem die LogoutRequests nicht immer empfangen werden können, wird noch eine Alternative zum SLO vorgeschlagen. Dabei kann ein solcher STIAM-Empfänger durch ein „Session Refresh“ (siehe Kapitel 2.7) feststellen, ob die SSO-Session am Hub noch gültig ist. Dazu sendet der STIAM-Empfänger in regelmässigen, selbstdefinierten Abständen einen Authentication Request an den Hub (siehe RP2 in Abbildung 10). Aus Performancegründen sollte dazu am STIAM-Hub ein expliziter SSO-Endpoint mit SOAP-Bindung für eine Backchannel-Kommunikation definiert werden.

Der Authentication Request wird vom STIAM-Hub mit einer Authentication Assertion mit der verbleibenden Dauer der SSO-Session beantwortet, bzw. mit einer Response ohne Authentication Assertion und dem Wert `urn:oasis:names:tc:SAML:2.0:status:AuthnFailed` im `<samlp:StatusCode>` des `<samlp:Status>` Elementes. Zusätzlich kann noch der StatusCode `urn:oasis:names:tc:SAML:2.0:status:PartialLogout` mitgegeben werden.

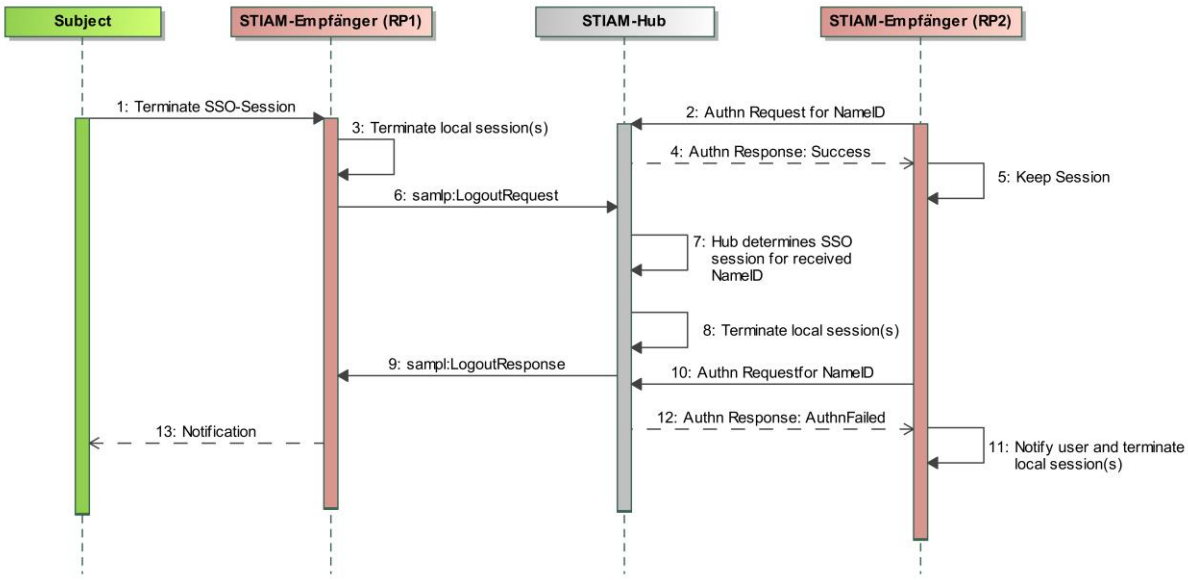

Abbildung 10: Zusammenspiel SLO und Session Refreshing

4.2 Zur Definitionszeit

Zur Definitionszeit kann man zwei Linking-Protokolle unterscheiden. Das IdP-Linking-Protokoll wird zwischen dem STIAM-Hub und einem STIAM-IdP verwendet (siehe Abbildung 11). Das AA-Linking-Protokoll ist in Abbildung 12 aufgezeigt. Es wird zwischen dem STIAM-Hub und einem STIAM-Sender (AA) verwendet.

4.2.1 IdP-Linking Protokoll

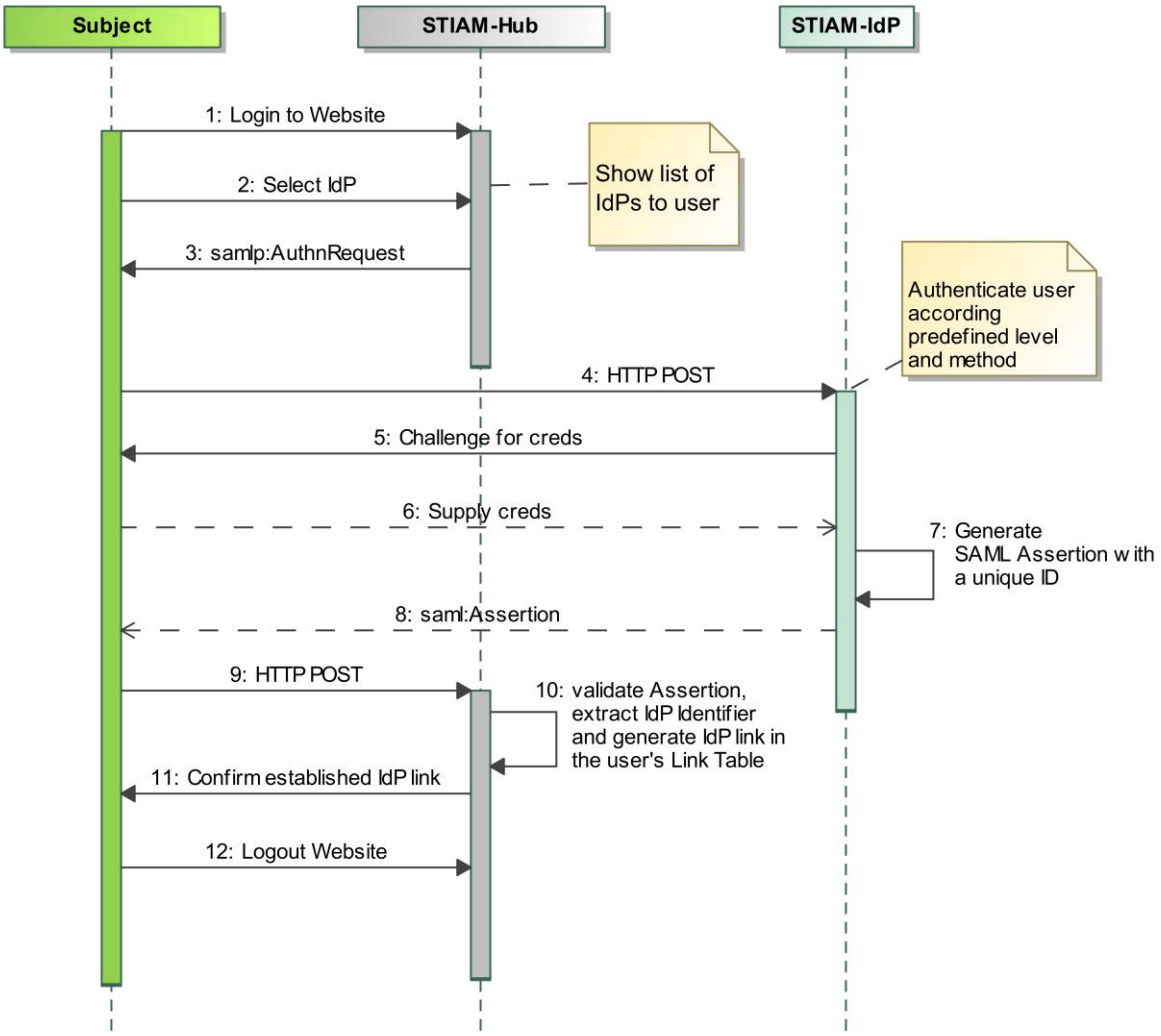

Abbildung 11: IdP-Linking Protokoll

Beschreibung zu Abbildung 11:

Schritt	Bemerkung
1	• Der Benutzer authentisiert sich am STIAM-Hub und wählt das IdP-Linking im Account Management an. • Wenn sich der Benutzer nicht bereits bei einem in seiner Link Table registrierten IdP mit höchstem QAA-Level authentisiert hat, so muss der Hub eine Step-up-Authentifizierung verlangen.
2	• Der Benutzer wählt einen IdP zum Verlinken aus der Liste aus.
3	• Der STIAM-Hub sendet einen digital signierten `<samlp:AuthnRequest>` in einer HTML Form an den Browser des Benutzers.
4	• Der Browser sendet den `<samlp:AuthnRequest>` zum `<md:SingleSignOnService>` des STIAM-IdPs als HTTP POST-Nachricht.
5	• Der IdP fordert den Benutzer zur Authentisierung auf (Username/Password, PKI basierend, 2FA, etc.)
6	• Der Benutzer authentisiert sich gegenüber dem Identity Provider.
7	• Der IdP generiert einen eindeutigen (nicht-transienten) Identifier und generiert eine SAML-Assertion mit diesem Identifier als NameID.
8	• Der IdP sendet dem Benutzer eine `<samlp:Response>` zurück, welche ein `<saml:AuthnStatement>` in einer `<saml:Assertion>` enthält. • Optional kann der IdP das Einverständnis vom Benutzer einholen, das Authentication Statement an den anfragenden STIAM-Hub auszuliefern. • Der IdP muss die `<saml:Assertion>` digital signieren.
9	• Der Browser leitet die `<saml:Assertion>` des IdP in einer HTTP POST-Nachricht zum STIAM-Hub weiter.
10	• Der STIAM-Hub validiert die `<saml:Assertion>` und erzeugt einen Eintrag in der Link-Tabelle des Benutzers.
11	• Der STIAM-Hub informiert den Benutzer über die Erstellung des IdP-Links.
12	• Der Benutzer verlässt die Webseite des Hubs und loggt sich aus.

Tabelle 15: Protokollschritte IdP-Linking

4.2.2 AA-Linking Protokoll

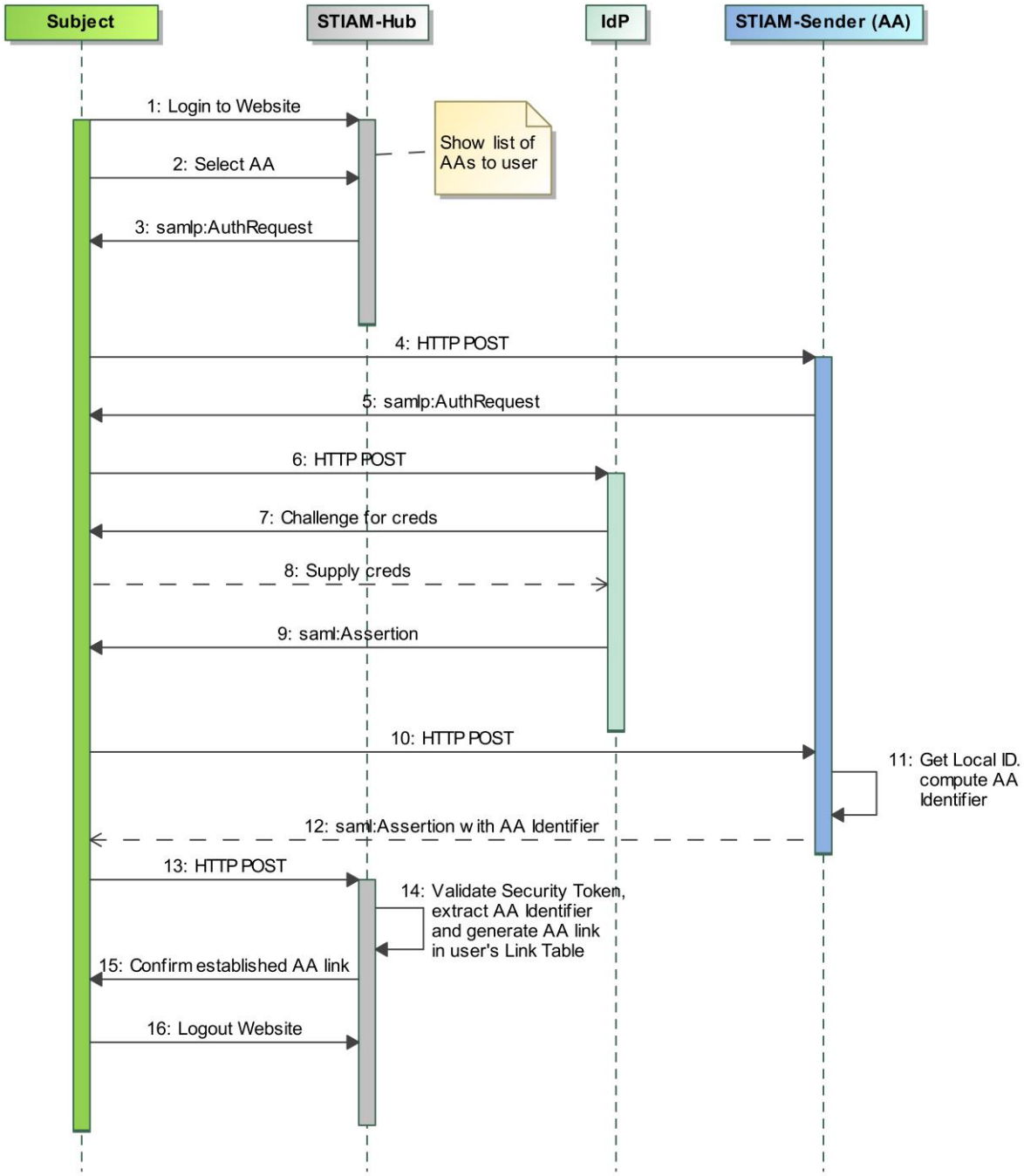

Abbildung 12: AA-Linking Protokoll

Beschreibung zu Abbildung 12:

Die Abbildung beschreibt die Variante 2 des AA-Linking (siehe Kapitel 3.2.2), bei welcher der Benutzer bei einem externen IdP authentifiziert wird. Für Variante 1 mit AA-internem IdP fallen die Schritte 5, 6, 9 und 10 weg, die Schritte 7 und 8 werden vom STIAM-Sender übernommen.

Schritt	Bemerkung
1	• Der Benutzer authentisiert sich am STIAM-Hub und wählt das AA-Linking im Account Management an. • Der Hub präsentiert dem Benutzer eine Liste von Attribute Authorities.
2	• Der Benutzer wählt einen AA zum Verlinken aus der Liste aus. Je nach (unter Schritt 1) erfolgtem QAA-Level bei der Authentifizierung, verlangt der STIAM-Hub eine Step-up-Authentifizierung mit dem gleichhohen oder höheren QAA-Level der gewählten AA.
3	• Der STIAM-Hub sendet einen digital signierten `<samlp:AuthnRequest>` in einer HTML Form an den Browser des Benutzers.
4	• Der Browser sendet den `<samlp:AuthnRequest>` zum `<md:SingleSignOnService>` des STIAM-Senders als HTTP POST-Nachricht.
5	• Der STIAM-Sender leitet den `<samlp:AuthnRequest>` zum STIAM-IdP über den Browser des Benutzers (per HTML Form) um.
6	• Der Browser sendet den `<samlp:AuthnRequest>` zum `<md:SingleSignOnService>` des STIAM-IdP als HTTP POST-Nachricht.
7	• Der IdP fordert den Benutzer zur Authentisierung auf (Username/Password, PKI basierend, 2FA, etc.)
8	• Der Benutzer authentisiert sich gegenüber dem Identity Provider.
9	• Der IdP authentifiziert den Benutzer und sendet ihm eine `<samlp:Response>` zurück, welche ein `<saml:AuthnStatement>` in einer `<saml:Assertion>` enthält. • Optional kann der IdP das Einverständnis vom Benutzer einholen, das Authentication Statement an den anfragenden STIAM-Sender auszuliefern. • Der IdP muss die `<saml:Assertion>` digital signieren.
10	• Der Browser leitet die `<saml:Assertion>` des IdP in einer HTTP POST-Nachricht zum STIAM-Sender weiter.
11	• Der STIAM-Sender validiert die `<saml:Assertion>` und berechnet den AA-Identifier (NameID der AA).
12	• Der STIAM-Sender erzeugt eine `<saml:Assertion>` mit dem berechneten eindeutigen AA-Identifier (unique ID[20]), signiert diese und sendet sie als `<samlp:Response>` an den Browser des Benutzers.

[20] Sowohl Distributed IDs als auch Persistent IDs garantieren, dass der STIAM-Hub immer denselben eindeutigen Identifier für den Benutzer erhält. Persistent IDs verbergen zusätzlich die Identität des Benutzers, setzen aber einen zusätzlichen Mapping-Schritt beim STIAM-Sender voraus.

Schritt	Bemerkung
13	• Der Browser leitet die `<samlp:Response>` an den STIAM-Hub weiter.
14	• Der STIAM-Hub validiert die `<saml:Assertion>` und erzeugt einen Eintrag in der Link Table des Benutzers.
15	• Der STIAM-Hub informiert den Benutzer über die Erstellung des AA-Links.
16	• Der Benutzer verlässt die Webseite des Hubs und loggt sich aus.

Tabelle 16: Protokollschritte AA-Linking

5 Metadaten

Wie in Kapitel 3 beschrieben, muss jede Komponente der STIAM-Community (STIAM-Hub, STIAM-IdP, STIAM-Sender und STIAM-Empfänger) über bestimmte SAML-Services verfügen. Die notwendigen Informationen zu diesen Services werden in der Datenbasis des STIAM-Hubs abgelegt. Diese komponentenspezifischen Informationen werden in der Regel manuell über ein GUI von einer dazu autorisierten Person (OrgSysAdmin) im Komponenten-Management (KM) (siehe eCH-0168 [4]) eingegeben.[21]

Im Komponenten-Management werden die Metainformationen aller internen und externen SAML-Komponenten von den zuständigen Administratoren erfasst.

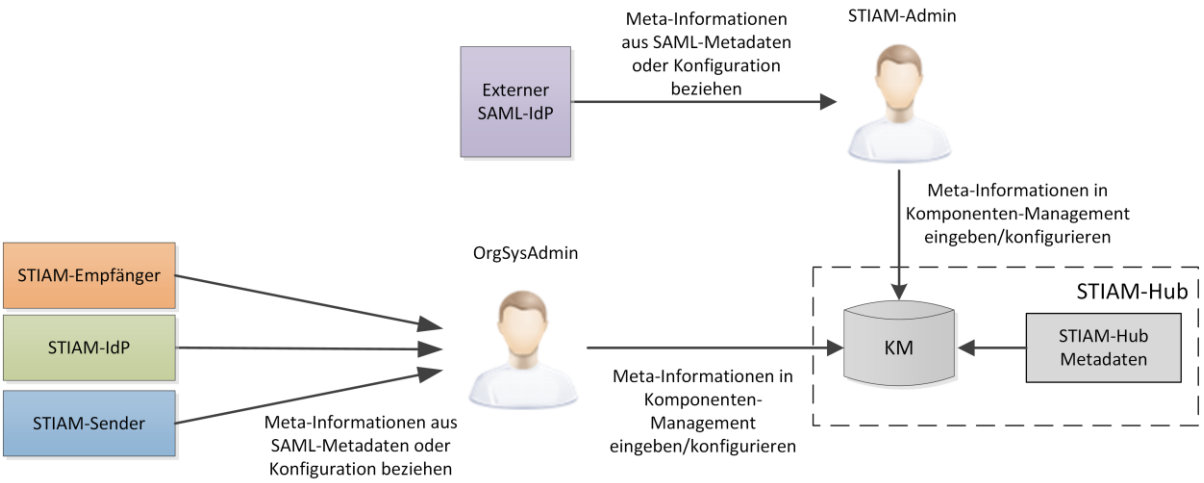

Abbildung 13: Erfassen der Meta-Informationen in KM

Der STIAM-Hub muss über die Metainformationen aller STIAM-Komponenten verfügen. Diese kann er direkt aus dem KM lesen. Die peripheren STIAM-Komponenten benötigen aber auch bestimmte Informationen einzelner STIAM-Komponenten aus diesem zentral geführten KM:

- Der STIAM-Empfänger muss die Metainformationen der SSO-Services des STIAM-Hubs kennen.
- Der STIAM-IdP muss die Metainformationen der ACS-Services des STIAM-Hubs kennen.
- Der STIAM-Sender muss die Metainformationen der SSO-Services des STIAM-Hubs und kann die Metainformationen der SSO-Services der STIAM-IdPs kennen.

Da alle peripheren Komponenten SAML 2.0 unterstützen, kann die Verteilung dieser Metainformationen mittels SAML-Metadaten erfolgen. Die Aufbereitung der notwendigen SAML-Metadaten erfolgt durch den STIAM-Metadata-Aggregator-Dienst (STIAM-MA).[22] Dazu muss

[21] Alternativ könnten diese Informationen auch auf der STIAM-Komponente lokal erzeugt und in den STIAM-Hub über eine vorgegebene Schnittstelle hochgeladen werden. Diese Uploadfunktion und die Verarbeitung dieser Metadaten auf dem Hub sind optional.

[22] Der STIAM-MA Dienst kann auch unabhängig vom STIAM-Hub als separate Komponente implementiert werden.

dieser Dienst periodisch die notwendigen Informationen[23] aus dem KM extrahieren und signiert publizieren. Anschliessend können diese Community-Metadaten von den STIAM-Mitgliedern abgeholt, validiert und bei sich integriert werden.

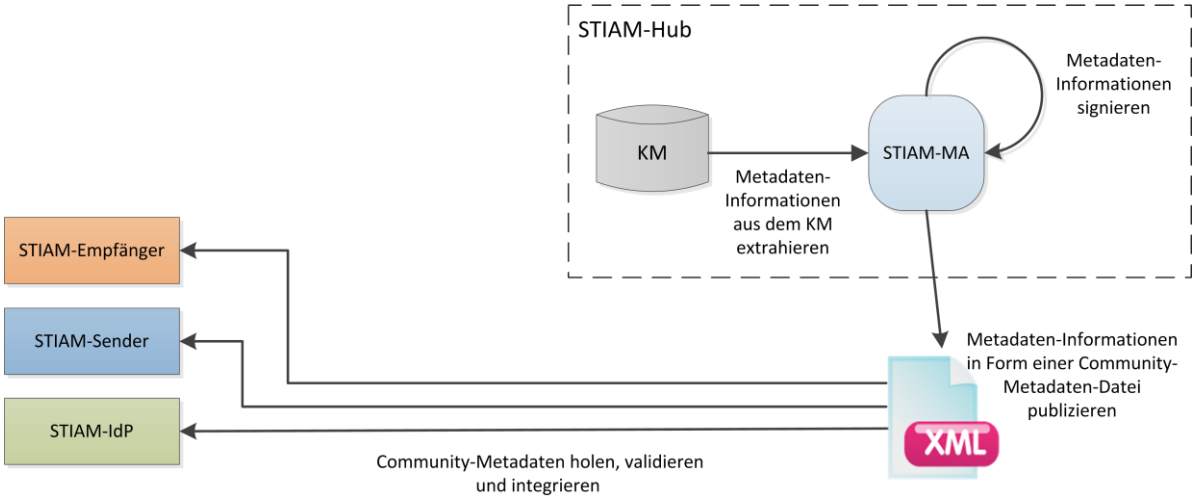

Abbildung 14: Publizierung der Community-Metadaten

Auf diese Weise verfügen die STIAM-Community-Mitglieder über alle notwendigen Informationen, um mit dem STIAM-Hub über SAML 2.0 kommunizieren zu können.

5.1 Community-Metadaten

Die Community-Metadaten enthalten alle notwendigen Informationen über den STIAM-Hub und die STIAM-IdPs. Sie werden periodisch vom STIAM-MA neu erstellt, signiert und publiziert.

Listing 1 zeigt ein Beispiel einer vom STIAM-MA publizierten Community-Metadaten-Datei.

Eine Community-Metadaten Datei enthält ein `<md:EntitiesDescriptor>` Element, welches die `<md:EntityDescriptor>` des Hubs (SSO und ACS) sowie die SSO aller STIAM-IdPs enthält. Das `validUntil` Attribut gibt die Gültigkeitszeit der Community-Metadaten-Datei und das `cacheDuration` Attribut die maximale Länge der Zeit an, während der STIAM-Mitglieder die Community-Metadaten-Datei speichern sollten.

[23] Der STIAM-MA Dienst publiziert nur die Informationen des STIAM-Hubs und der STIAM-IdPs.

```
<md:EntitiesDescriptor
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    validUntil="2017-02-20T23:00:00Z"
    cacheDuration="PT24H"
    ID="csxy-3wwa-qy01-ewda-e1df-xydg">

    <ds:Signature>...</ds:Signature>

    <!-- STIAM-Hub -->
    <md:EntityDescriptor
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    ID="eenl-4ftv-7uqa-lmg3-q123-vsaq"
    entityID="https://hub.gov.ch">
     ...
    </md:EntityDescriptor>

    <!-- STIAM-IdP -->
    <md:EntityDescriptor
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    ID="eenl-2rxp-7uqa-q123-tecm"
    entityID="https://idp.gov.ch">
     ...
    </md:EntityDescriptor>
</md:EntitiesDescriptor>
```

Listing 1: Community-Metadaten-Datei

5.2 SAML-Metadaten Richtlinien

In der Folge werden die zu publizierenden STIAM-Hub und STIAM-IdP Metadaten näher beschrieben. Sie enthalten unter anderem Informationen über:

- Die Adresse und den Namen der Entität (STIAM-Komponente).
- Die Endpunktkonfigurationen der Entität (URL).
- Die Public-Key-Zertifikate zur Prüfung signierter SAML-Nachrichten.
- Die Public-Key-Zertifikate zur Prüfung signierter SAML-Assertions.
- SAML-Attribute, die von der Entität konsumiert/erzeugt werden können.

5.2.1 Allgemeine Vorgaben zu `<md:EntityDescriptor>` Elementen

- Die STIAM-Hub Metadaten Definition MUSS mit einem `<md:EntityDescriptor>` Element beginnen.
- Das `<md:EntityDescriptor>` Element MUSS ein `entityID` Attribut haben. Dessen Wert MUSS eine URI sein, welche in der STIAM-Community eindeutig ist und als Identifikator verwendet wird.
- Im `<md:EntityDescriptor>` Element MUSS ein `<Extensions>` Element mit den vom STIAM-Hub oder STIAM-IdP unterstützten *Authentication Assurance Level* gemäss *SAMLV2.0 Identity Assurance Profiles Version 1.0* [12] angeben werden.
- Das `<md:EntityDescriptor>` Element KANN ein oder mehrere Elemente vom Typ `<md:IDPSSODescriptor>`, `<md:SPSSODescriptor>` oder `<md:AttributeAuthorityDescriptor>` enthalten.
- Das `<md:EntityDescriptor>` Element KANN ein `<md:Organization>` Element enthalten, welches wiederum ein `<md:OrganizationName>` und eine `<md:OrganizationURL>` aufweist.

- Ein `<md:OrganizationDisplayName>` Element und ein `<md:ContactPerson>` Element sind OPTIONAL.

5.2.2 Vorgaben zu STIAM-Hub Metadaten

IDPSSODescriptor:

Der IDPSSODescriptor beschreibt die SSO-Metainformationen des STIAM-Hubs.

- Das STIAM-Hub `<md:EntityDescriptor>` Element MUSS ein Element vom Typ `<md:IDPSSODescriptor>` und ein Element vom Typ `<md:SPSSODescriptor>` enthalten.
- Das `<md:IDPSSODescriptor>` Element MUSS ein `protocolSupportEnumeration` Attribut haben, dessen Wert MUSS `urn:oasis:names:tc:SAML:2.0:protocol` sein.
- Das `WantAuthenticationRequestsSigned` Attribut des `<md:IDPSSODescriptor>` Elementes MUSS auf `true` gesetzt werden. Dies bedeutet, dass die STIAM-Empfänger den Authentication Request (`<samlp:AuthnRequest>`) signieren MÜSSEN, sonst wird dieser vom STIAM-Hub nicht akzeptiert.
- Das `<md:KeyDescriptor>` Element für Signatur MUSS im `<md:IDPSSODescriptor>` vorhanden sein. Dessen `<ds:KeyInfo>` MUSS ein `<ds:X509Data>` Element enthalten, und dieses wiederum ein `<ds:X509Certificate>`.
- Das `<md:IDPSSODescriptor>` Element MUSS ein oder mehrere `<md:SingleSignOnService>` Elemente enthalten.
- Das `<md:IDPSSODescriptor>` Element MUSS ein oder mehrere `<md:SingleLogOutService>` Elemente enthalten.
- Das `<md:IDPSSODescriptor>` Element MUSS mehrere `<md:NameIDFormat>` Elemente enthalten. Es MÜSSEN `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` und `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` unterstützt werden [11].

SPSSODescriptor:

Der SPSSODescriptor beschreibt die Service Provider Metainformationen des STIAM-Hubs.

- Das STIAM-Hub `<md:SPSSODescriptor>` Element MUSS ein `protocolSupportEnumeration` Attribut haben, dessen Wert MUSS `urn:oasis:names:tc:SAML:2.0:protocol` sein.
- Das `WantAssertionsSigned` Attribut des `<md:SPSSODescriptor>` Elementes MUSS auf `true` gesetzt werden.
- Das `AuthnRequestsSigned` Attribut des `<md:SPSSODescriptor>` Elementes MUSS auf `true` gesetzt sein.
- Das `<md:KeyDescriptor>` Element für Signatur MUSS im `<md:SPSSODescriptor>` vorhanden sein. Dessen `<ds:KeyInfo>` MUSS ein `<ds:X509Data>` Element enthalten, und dieses wiederum ein `<ds:X509Certificate>`.
- Das `<md:SPSSODescriptor>` Element MUSS ein oder mehrere `<md:AssertionConsumerService>` Elemente enthalten. Das Protocol Binding `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST` MUSS unterstützt werden. Das `index` Attribut MUSS in jedem `<md:AssertionConsumerService>` Element vorhanden sein.

- Das `<md:SPSSODescriptor>` Element MUSS mindestens ein `<md:NameIDFormat>` Element enthalten. Dessen Wert DARF NICHT `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` sein.[24]

[24] Das `<md:SPSSODescriptor>` Element beschreibt den Protokoll-Endpunkt des STIAM-Hubs, der die Antworten auf Authentication Requests und Attribute Queries. Sowohl STIAM-IdPs als auch STIAM-Sender müssen eindeutige, nicht-transiente IDs verwenden, um das Identity-Linking zu ermöglichen.

Beispiel für eine STIAM-Hub Metadaten Definition:

```
<md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
ID="eenl-4ftv-7uqa-lmg3-q123-vsaq"
entityID="https://hub.gov.ch">

    <Extensions>
        <attr:EntityAttributes>
             <saml:Attribute
              NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
              Name="urn:oasis:names:tc:SAML:attribute:assurance-certification">
                    <saml:AttributeValue>
                         https://stiam.gov.ch/authenticationassurancelevel(1-4)
                    </saml:AttributeValue>
             </saml:Attribute>
        </attr:EntityAttributes>
    </Extensions>

    <md:IDPSSODescriptor
        WantAuthnRequestsSigned="true"
        protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

        <md:KeyDescriptor  use="signing">
            <ds:KeyInfo>
                <ds:X509Data>
                    <ds:X509Certificate>
                        ...
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </md:KeyDescriptor>
        <md:SingleLogoutService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
            Location=" https://hub.gov.ch/SAML/SLO/SOAP"/>
        <md:SingleLogoutService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"

            Location="https://hub.gov.ch/SAML/SLO/Browser"
            ResponseLocation=" https://hub.gov.ch/SAML/SLO/Response"/>
        <md:NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:transient
        </md:NameIDFormat>
        <md:NameIDFormat>
            urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
        </md:NameIDFormat>
        <md:SingleSignOnService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
            Location="https://hub.gov.ch/SAML/SSO/Browser"/>
        <md:SingleSignOnService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
            Location="https://hub.gov.ch/SAML/SSO/Browser"/>
</md:IDPSSODescriptor>
```

```
<md:SPSSODescriptor
        AuthnRequestsSigned="true"  WantAssertionsSigned="true"
        protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <md:KeyDescriptor use="signing">
            <ds:KeyInfo>
                <ds:X509Data>
                    <ds:X509Certificate>
                        ...
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </md:KeyDescriptor>

      <md:NameIDFormat>
                urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
     </md:NameIDFormat>

     <md:AssertionConsumerService  isDefault="true"  index="1"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                Location="https://sp.example.ch/SAML/ACS/POST"/>
     <md:AttributeConsumingService  index="1" isDefault="true">
          <md:ServiceName xml:lang="en">hub.gov.ch</md:ServiceName>
    </md:AttributeConsumingService>
</md:SPSSODescriptor>

  <md:Organization>
        <md:OrganizationName xml:lang="en">
            STIAM-Hub Provider
        </md:OrganizationName>
        <md:OrganizationDisplayName xml:lang="en">
            SuisseTrust-IAM Service Provider Corp.
        </md:OrganizationDisplayName>
        <md:OrganizationURL xml:lang="en">
            https://stiam.gov.ch
        </md:OrganizationURL>
  </md:Organization>
  <md:ContactPerson contactType="administrative">
        <md:GivenName>Hans</md:GivenName>
        <md:SurName>Muster</md:SurName>
        <md:EmailAddress>hansm@gov.ch</md:EmailAddress>
  </md:ContactPerson>

</md:EntityDescriptor>
```

Listing 2: Beispiel eines STIAM-Hub Entity Descriptors

5.2.3 Vorgaben zu STIAM-IdP Metadaten

Im `<md:EntitiesDescriptor>` Element können nebst dem Hub weitere Definitionen für STIAM-IdPs integriert werden. Diese `<md:EntityDescriptor>` Elemente weisen nur ein `<md:IDPSSODescriptor>` Element mit den folgenden Angaben auf (vgl. dazu Listing 3):

- Das `<md:IDPSSODescriptor>` Element MUSS ein `protocolSupportEnumeration` Attribut haben, dessen Wert MUSS `urn:oasis:names:tc:SAML:2.0:protocol` sein.
- Das `WantAuthenticationRequestsSigned` Attribut des `<md:IDPSSODescriptor>` Elementes MUSS auf `true` gesetzt werden.
- Das `<md:KeyDescriptor>` Element für Signatur MUSS im `<md:IDPSSODescriptor>` vorhanden sein. Dessen `<ds:KeyInfo>` MUSS ein `<ds:X509Data>` Element enthalten, und dieses wiederum ein `<ds:X509Certificate>`.

- Das `<md:IDPSSODescriptor>` Element MUSS ein oder mehrere `<md:SingleSignOnService>` Elemente enthalten.
  Das HTTP-POST Protocol Binding MUSS unterstützt sein.
- Das `<md:IDPSSODescriptor>` Element MUSS ein `<md:NameIDFormat>` Element enthalten. Dessen Wert DARF NICHT `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` sein.

```
<md:EntityDescriptor>
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    ID="eenl-2rxp-7uqa-q123-tecm"
    entityID="https://idp.gov.ch">

    <Extensions>
        <attr:EntityAttributes>
             <saml:Attribute
              NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
              Name="urn:oasis:names:tc:SAML:attribute:assurance-certification">
                    <saml:AttributeValue>
                         urn:stiam.gov.ch/authenticationassurancelevel(1-4)
                    </saml:AttributeValue>
             </saml:Attribute>
        </attr:EntityAttributes>
    </Extensions>

   <md:IDPSSODescriptor
           WantAuthnRequestsSigned="true"
           protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

           <md:KeyDescriptor use="signing">
               <ds:KeyInfo>
                   <ds:X509Data>
                       <ds:X509Certificate>
                           ...
                       </ds:X509Certificate>
                   </ds:X509Data>
               </ds:KeyInfo>
           </md:KeyDescriptor>

           <md:NameIDFormat>
               urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
           </md:NameIDFormat>

           <md:SingleSignOnService
               Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
               Location=" https://idp.example.ch/SAML/SSO/Browser"/>
      </md:IDPSSODescriptor>
 </md:EntityDescriptor>
```

Listing 3: Beispiel eines STIAM-IdP EntityDescriptor

6 Messages

Festzulegen, wie die STIAM-Community-Mitglieder miteinander kommunizieren, ist für eine bessere Gewährleistung der Sicherheit und Interoperabilität des STIAMs sehr wichtig. Dieses Kapitel befasst sich mit den verschiedenen Messages, die zwischen den STIAM-Community-Mitgliedern ausgetauscht werden.

6.1 Messages Richtlinien

In diesem Abschnitt werden die Richtlinien für die Erstellung der verschiedenen Messages zwischen den STIAM-Komponenten festgelegt.

6.1.1 Richtlinien für alle Messages

- Das `ID` Attribut im Wurzelelement MUSS in jeder Message vorhanden sein. Der Wert MUSS in der Message eindeutig sein. Er wird als Referenz bei der Signatur der Message verwendet.
- Das `Destination` Attribut im Wurzelelement jeder Message MUSS immer vorhanden sein. Dessen Wert MUSS die URL sein, an die die Message gesendet wird.
- Das `IssueInstant` Attribut im Wurzelelement jeder Message MUSS immer vorhanden sein. Dessen Wert gibt den Zeitpunkt der Erstellung der Message an. Dieser MUSS in UTC codiert werden.
- Das `Version` Attribut im Wurzelelement jeder Message MUSS immer vorhanden sein. Dessen Wert MUSS `2.0` sein.
- Der Wert des `<saml:Issuer>` Elements MUSS in jeder Message mit dem `EntityID` Attribut aus den Metadaten derjenigen Entität übereinstimmen, welche die Message erstellt hat.
- Alle Messages MÜSSEN mit einem in der STIAM-Community anerkannten Zertifikat (STIAM-Applikationszertifikat) digital signiert sein (`<ds:Signature>` Element enthalten).

6.1.2 Richtlinien für Authentication Requests

- Das `<samlp:AuthnRequest>` Element MUSS die Wurzel des Authentication Requests sein.
- Das `<samlp:AuthnRequest>` Element MUSS die Attribute `AssertionConsumerServiceURL` und `ProtocolBinding` enthalten.

  Der Wert von `AssertionConsumerServiceURL` MUSS mit dem `AssertionConsumerService` Element aus den Metadaten derjenigen Entität übereinstimmen, welche den Authentication Request erstellt hat.

  Der Wert des `ProtocolBinding` Attributes MUSS `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST` sein.
- Das `<samlp:AuthnRequest>` Element KANN ein `ForceAuthn` Attribut enthalten.

- Das `<samlp:AuthnRequest>` Element KANN ein `<samlp:NameIDPolicy>` Element enthalten. Der Wert dessen `Format` Attribut MUSS `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` oder `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` sein. Falls er persistent ist, MUSS im `<samlp:NameIDPolicy>` Element das `AllowCreate` Attribut vorhanden sein, und dessen Wert MUSS auf `true` gesetzt werden.

  Wenn das `<samlp:NameIDPolicy>` Element weggelassen wird, gilt das `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` Format als Default.
- Wenn der Authentication Request vom STIAM-Empfänger erstellt wird, SOLLTE dieser das `AttributeConsumingServiceIndex` Attribut enthalten. Falls dieses nicht vorhanden ist, gilt der Default[25].
- Der Authentication Request KANN ein `<saml:Subject>` Element mit einem `nameID` Element für den Fall eines Session-Refreshing oder einer Step-Up-Authentifizierung enthalten.
- Der Authentication Request KANN ein `<samlp:RequestedAuthnContext>` Element mit einem `<saml:AuthnContextClassRef>` Element enthalten, wenn eine Authentisierung mit einem höheren Level erwünscht wird als in den Metadaten bzw. im KM des STIAM-Hubs gefordert wird. Der Wert des `<saml:AuthnContextClassRef>` Elements MUSS einer der folgenden STIAM Qualitätslevels nach eCH-0170 [7] sein, z.B:
  - urn:stiam.gov.ch/authenticationassurance/level1
  - urn:stiam.gov.ch/authenticationassurance/level2
  - urn:stiam.gov.ch/authenticationassurance/level3
  - urn:stiam.gov.ch/authenticationassurance/level4
- Der Authentication Request DARF KEINE weiteren Elemente (z.B. Conditions, Scope, …) enthalten.

Beispiel eines AuthnRequest (STIAM-Empfänger zu STIAM-Hub)

Das `AssertionConsumerServiceURL` Attribut gibt die URL an, an welche die vom STIAM-Hub erstellte Response gesendet werden muss.

Der Wert des `AttributeConsumingServiceIndex` Attributes stimmt mit dem vom `index` Attribut eines `<md:AttributeConsumingService>` Elements in den Metadaten des

---

[25] Das `AttributeConsumingServiceIndex` Attribut entspricht der Ressource ID (RES_ID) aus dem KM des STIAM-Hubs. Wenn ein STIAM-Empfänger zwischen verschieden Ressourcen unterscheiden möchte, müssen diese im KM konfiguriert werden. Eine der Ressourcen muss als Default markiert werden. Die Ressourcen-Konfiguration kann optional auch in den Metadaten des STIAM-Empfängers vorgenommen und in das KM importiert werden. Die Ressourcen werden in den Metadaten als `<md:AttributeConsumingService>` Elemente angegeben.

STIAM-Empfängers überein. Dieser referenziert indirekt die Attribute, welche im Authentication Request abgefragt werden sollen.

Das `<samlp:NameIDPolicy>` Element wurde im Authentication Request weggelassen, somit gilt hier das `transient` Format als Default.

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    ID="ewda-e1df-xydg-xwsq"
    Version="2.0"
    IssueInstant="2013-12-05T09:21:59Z"
    Destination="https://hub.gov.ch/SAML/SSO/Browser"
    AssertionConsumerServiceURL="https://sp.example.ch/SAML/ACS/POST"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    AttributeConsumingServiceIndex="1">

    <saml:Issuer>https://sp.example.ch</saml:Issuer>

    <ds:Signature>...</ds:Signature>

</samlp:AuthnRequest>
```

Listing 4: AuthnRequest vom STIAM-Empfänger zum STIAM-Hub

Beispiel eines AuthnRequest (STIAM-Hub zu STIAM-IdP)

Der Authentication Request vom STIAM-Hub zu einem STIAM-IdP ist ähnlich wie der vom STIAM-Empfänger zum STIAM-Hub, ausser dass ersterer kein `AttributeConsumingServiceIndex` Attribut enthält.

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    ID="mkqs-ezew-qplo-snrt"
    Version="2.0"
    IssueInstant="2013-12-05T09:22:30Z"
    Destination="https://idp.example.ch/SAML/SSO/Browser"
    AssertionConsumerServiceURL="https://hub.gov.ch/SAML/ACS/Browser"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST">

    <saml:Issuer>https://hub.gov.ch</saml:Issuer>

    <ds:Signature>...</ds:Signature>

    <samlp:NameIDPolicy
        AllowCreate="true"
        Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" />

</samlp:AuthnRequest>
```

Listing 5: AuthnRequest (STIAM-Hub zu STIAM-IdP)

6.1.3 Richtlinien für Attribute Queries

- Das `<samlp:AttributeQuery>` Element MUSS die Wurzel des Attribute Query Requests sein.
- Das `<samlp:AttributeQuery>` Element MUSS ein `<saml:Subject>` Element enthalten. Dieses MUSS ein `<saml:NameID>` enthalten. Dessen Format DARF NICHT `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` sein.
- Das `<samlp:AttributeQuery>` Element MUSS ein oder mehrere `<Attribute>` Elemente enthalten.
- Das `<samlp:AttributeQuery>` DARF NICHT zwei oder mehrere `<Attribute>` Elemente enthalten, die dasselbe `Name` und `NameFormat` Attribut haben.
- Das `<samlp:AttributeQuery>` Element KANN ein `<samlp:Extensions>` Element enthalten. Das `<samlp:Extensions>` Element MUSS ein `<samlp:Assertion>` Element enthalten.

Beispiel für eine Standard Attribute Query

Im Listing 6 wird das Attribut *E-Mail Adresse* mit *Qualitätslevel 3*[26] des Benutzers abgefragt, welches im `<saml:NameID>` Element identifiziert wird.

[26] Die Angabe der Qualitätslevel der Attribute nach eCH-0171 [8] sollte standardisiert werden.

```
<samlp:AttributeQuery
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:ech-0174="http://www.stiam.ch/1.0"
    ID="aafe-we23-enzz-d3et"
    Version="2.0"
    IssueInstant="2013-12-05T09:26:05Z"
    Destination="https://aa.example.ch/SAML/AS/POST">

    <saml:Issuer>https://hub.gov.ch</saml:Issuer>

    <ds:Signature>...</ds:Signature>

    <saml:Subject>
        <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:
            nameid-format:unspecified">
            2347-0987-4few-cf643-jtf12swe
        </saml:NameID>
    </saml:Subject>

    <saml:Attribute
     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
     Name=
     "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
     ech-0174:AQAA-Level="3">
    </saml:Attribute>

</samlp:AttributeQuery>
```

Listing 6: Standard Attribute Query (STIAM-Hub zu STIAM-Sender)

Beispiel für eine Extended Attribute Query

```
<samlp:AttributeQuery
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:ech-0174="http://www.stiam.ch/1.0"
    ID="aawd-wedz-wczg-pmdr"
    Version="2.0"
    IssueInstant="2013-12-05T09:28:05Z"
    Destination="https://aa.example.ch/SAML/AS/POST">

    <saml:Issuer>https://hub.gov.ch</saml:Issuer>
    <ds:Signature>...</ds:Signature>

    <samlp:Extensions>
      <saml:Assertion
        ID="ojqx-mlhj-xydg-xdew"
        Version="2.0"
        IssueInstant="2013-12-05T09:23:59Z">
        <saml:Issuer>https://idp.example.ch</saml:Issuer>
        <ds:Signature>...</ds:Signature>
        <saml:Subject>
          <saml:NameID Format=
          "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
                3f7b3dcf-1674-4ecd-92c8-1544f346baf8
           </saml:NameID>
           <saml:SubjectConfirmation
                Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <saml:SubjectConfirmationData
                    NotOnOrAfter="2013-12-05T09:33:59Z "
                    Recipient="https://hub.gov.ch/SAML/ACS/Browser "
                    InResponseTo="mkqs-ezew-qplo-snrt"/>
            </saml:SubjectConfirmation>
        </saml:Subject>
        <saml:Conditions
            NotBefore="2013-12-05T09:23:59Z"
            NotOnOrAfter="2013-12-05T09:33:59Z">
            <saml:AudienceRestriction>
                <saml:Audience>https://hub.gov.ch</saml:Audience>
            </saml:AudienceRestriction>
        </saml:Conditions>
        <saml:AuthnStatement
            AuthnInstant="2013-12-05T09:23:50Z"
            SessionIndex="234122">
            <saml:AuthnContext>
                <saml:AuthnContextClassRef>
                    urn:stiam.gov.ch/authenticationassurancelevel3
                </saml:AuthnContextClassRef>
            </saml:AuthnContext>
        </saml:AuthnStatement>
      </saml:Assertion>
     </samlp:Extensions>
```

```
    <saml:Subject>
        <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:
            nameid-format:unspecified">
            3f7b3dcf-1674-4ecd-92c8-1544f346baf8
        </saml:NameID>
    </saml:Subject>
    <saml:Attribute
     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
     Name=
     "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
     ech-0174:AQAA-Level="3">
    </saml:Attribute>

</samlp:AttributeQuery>
```

Listing 7: Extended Attribute Query (STIAM-Hub zu STIAM-Sender)

6.1.4 Richtlinien für Responses

- Das `<samlp:Response>` Element MUSS die Wurzel der Response Message sein.
- Das `<saml:Response>` Element MUSS mit einem vom STIAM-Verbund anerkannten Zertifikat digital signiert sein (`<ds:Signature>` Element enthalten).
- Das `<Response>` Element MUSS ein `InResponseTo` Attribut enthalten.

  Das `InResponseTo` Attribut MUSS mit der `ID` der Abfrage, für welche die Response erstellt wurde, übereinstimmen.
- Das `<samlp:Response>` Element MUSS ein `<samlp:Status>` Element enthalten und dieses wiederum ein `<samlp:StatusCode>`.
- Beim erfolgreichen Authentication Request MUSS das `<samlp:Response>` Element ein `<saml:Assertion>` Element enthalten.
- Ist der Authentication Request nicht erfolgreich, MUSS im `<samlp:StatusCode>` Element eine Fehlermeldung nach SAML2.0 [11] vorhanden sein und die Assertion fällt weg.

Beispiel einer Response (STIAM-IdP zu STIAM-Hub)

Das `InResponseTo` Attribut enthält den Wert des `ID` Attributes des Authentication Requests, für den die Response erstellt wurde. Falls dieser Wert nicht mit dem des `ID` Attributes des Authentication Requests übereinstimmt, DARF der STIAM-Hub die Response NICHT akzeptieren.
Das `<saml:Status>` Element spezifiziert den Status des zu dieser Response gehörenden Authentication Requests. So wird im `Value` Attribut des `<saml:StatusCode>` Elementes genaue Information über den Status des Authentication Requests angegeben. Im untenstehenden Listing 8 wurde dieser erfolgreich durchgeführt.

Das `<saml:Assertion>` Element wird im folgenden Abschnitt beschrieben.

```
<samlp:Response
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    ID="lnqw-xqap-xydg-kxsr"
    InResponseTo="mkqs-ezew-qplo-snrt"
    Version="2.0"
    IssueInstant="2013-12-05T09:23:59Z"
    Destination="https://hub.gov.ch/SAML/ACS/Browser">

    <saml:Issuer>https://idp.example.ch</saml:Issuer>
    <ds:Signature>...</ds:Signature>

    <samlp:Status>
      <samlp:StatusCode Value=
    "urn:oasis:names:tc:SAML:2.0:status:Success" />
    </samlp:Status>

    <saml:Assertion>...</saml:Assertion>

  </samlp:Response>
```

Listing 8: Response (STIAM-IdP zu STIAM-Hub)

6.1.5 Richtlinien für Assertions

- Das `<saml:Assertion>` Element MUSS ein `ID` und `IssueInstant` Attribut enthalten.
- Das `<saml:Assertion>` Element MUSS ein `<saml:Issuer>` Element enthalten. Dessen Wert MUSS mit dem vom `EntityID` Attribut aus den Metadaten derjenigen Entität übereinstimmen, welche die Assertion erstellt hat.
- Das `<saml:Assertion>` Element MUSS mit einem bei der STIAM anerkannten Zertifikat digital signiert sein (`<ds:Signature>` Element enthalten).
- Das `<saml:Assertion>` Element MUSS ein `<saml:Subject>` Element enthalten. Dieses MUSS ein `<saml:NameID>` und `<saml:SubjectConfirmation>` Element enthalten.

  Das `<saml:SubjectConfirmation>` Element MUSS ein `Method` Attribut haben. Dessen Wert MUSS `urn:oasis:names:tc:SAML:2.0:cm:bearer` [9] sein.

  Das `<saml:SubjectConfirmation>` Element MUSS ein `<saml:SubjectConfirmationData>` Element haben. Dieses MUSS ein `InResponseTo`, `Recipient` and `NotOnOrAfter` Attribut haben.
- Das `<saml:Assertion>` Element MUSS ein `<saml:Conditions>` Element enthalten. Dieses MUSS ein `NotBefore` und `NotOnOrAfter` Attribut haben. Ausserdem MUSS dieses ein `<saml:AudienceRestriction>` Element enthalten, welches ein `<saml:Audience>` Element hat. Dessen Wert MUSS mit dem `Enti-`

tyID Attribut aus den Metadaten der Entität übereinstimmen, für welche die Assertion erstellt wurde[27].

- Das <Assertion> Element MUSS genau ein <saml:AuthnStatement> Element enthalten. Dieses MUSS ein AuthnInstant und ein SessionIndex Attribut, sowie ein <saml:AuthnContext> Element enthalten.

  Das <saml:AuthnContext> Element MUSS ein <saml:AuthnContextClassRef> Element enthalten. Dessen Wert MUSS einer der folgenden STIAM-Qualitätslevels nach eCH-0170 [7] sein:

  - urn:stiam.gov.ch/authenticationassurance/level1
  - urn:stiam.gov.ch/authenticationassurance/level2
  - urn:stiam.gov.ch/authenticationassurance/level3
  - urn:stiam.gov.ch/authenticationassurance/level4

- Das <Assertion> Element KANN ein <saml:AttributeStatement> Element enthalten.

  Das <saml:AttributeStatement> Element MUSS ein oder mehrere <saml:Attribute> Elemente enthalten und diese KÖNNEN ein oder mehrere <saml:AttributeValue> Elemente haben.

Beispiel einer Assertion (STIAM-IdP zu STIAM-Hub)

Der authentifizierte Benutzer, für welchen die Äusserung im <saml:AuthnStatement> Element der Assertion gilt, wird im <saml:Subject> Element identifiziert.

Im <saml:NameID> Element wird ein persistenter Name Identifier definiert, um die Identität des Benutzers zu verbergen.

Das InResponseTo Attribut des <saml:SubjectConfirmationData> Elements enthält den Wert des ID Attributes des Authentication Requests, für den die Assertion erstellt wurde. Falls dieser Wert nicht mit dem des ID Attributes des Authentication Requests übereinstimmt, DARF der STIAM-Hub die Assertion NICHT akzeptieren. Das Recipient Attribut enthält die Assertion Consumer URL des STIAM-Hubs – die URL, an die die Assertion gesendet wurde.

Das <saml:Conditions> Element gibt an, ab und bis wann die Assertion gültig ist. Der Wert im <saml:Audience> Element beschreibt, für welches STIAM-Mitglied die Assertion erstellt wurde. In untenstehendem Listing 9 wurde die Assertion für den STIAM-Hub erstellt.

Im <saml:AuthnStatement> Element wird beschrieben, mit welchem QAA-Level nach eCH-0170 [7] sich der Benutzer bei dem STIAM-IdP authentifiziert hat. Das AuthnInstant Attribut gibt die Zeit der Authentifizierung an und das SessionIndex Attribut identifiziert die

[27] In der Assertion der Extended Attribute Query stimmt der Wert des <saml:Audience> Elements nicht mit dem EntityID Attribut aus den Metadaten eines STIAM-Senders überein, da die Assertion für den STIAM-Hub erstellt wurde und dieser sie an den STIAM-Sender weiterleitet.

Session, die im STIAM-IdP für den Benutzer nach der Authentifizierung erstellt wurde.

```
<saml:Assertion
      ID="ojqx-mlhj-xydg-xdew"
      Version="2.0"
      IssueInstant="2013-12-05T09:23:59Z">
      <saml:Issuer>https://idp.example.ch</saml:Issuer>

      <ds:Signature>...</ds:Signature>

      <saml:Subject>
        <saml:NameID Format=
        "urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
              3f7b3dcf-1674-4ecd-92c8-1544f346baf8
         </saml:NameID>
     <saml:SubjectConfirmation
             Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
             <saml:SubjectConfirmationData
                 NotOnOrAfter="2013-12-05T09:33:59Z "
                 Recipient="https://hub.gov.ch/SAML/ACS/Browser "
                 InResponseTo="mkqs-ezew-qplo-snrt"/>
         </saml:SubjectConfirmation>
      </saml:Subject>

      <saml:Conditions
          NotBefore="2013-12-05T09:23:59Z"
          NotOnOrAfter="2013-12-05T09:33:59Z">
          <saml:AudienceRestriction>
              <saml:Audience>https://hub.gov.ch</saml:Audience>
          </saml:AudienceRestriction>
      </saml:Conditions>

      <saml:AuthnStatement
          AuthnInstant="2013-12-05T09:23:50Z"
          SessionIndex="234122">
          <saml:AuthnContext>
              <saml:AuthnContextClassRef>
                  https://stiam.gov.ch/authenticationassurance/level3
              </saml:AuthnContextClassRef>
          </saml:AuthnContext>
      </saml:AuthnStatement>

  </saml:Assertion>
```

Listing 9: Assertion vom STIAM-IdP zum STIAM-Hub

Beispiel einer Assertion (STIAM-Sender zu STIAM-Hub)

Die Assertion vom STIAM-Sender zum STIAM-Hub ist ähnlich wie die vom STIAM-IdP zum STIAM-Hub. Sie enthält ein `<saml:AttributeStatement>` Element, aber kein `<saml:AuthnStatement>`.

Im `<saml:AttributeStatement>` Element werden Attribute zu dem vorher beim STIAM-IdP authentifizierten Benutzer[28] geliefert.

[28] Der Benutzer, welcher im `<saml:NameID>` Element referenziert ist.

Das `<saml:AttributeValue>` Element enthält neben dem Typ und dem Wert des Attributes auch die Attribut-Qualität (AQAA-Level) nach eCH-0171 [8].

```
  <saml:Assertion

xmlns:xs="http://schemas.xmlsoap.org/ws/2005/05/identity/clai
ms"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ech-0174="http://www.stiam.ch/1.0"
        ID="mnbv-mlhj-acmu-pzsg"
        Version="2.0"
        IssueInstant="2013-12-05T09:25:05Z">

        <saml:Issuer>https://aa.example.ch</saml:Issuer>

        <ds:Signature>...</ds:Signature>

        <saml:Subject>
            <saml:NameID Format=
            "urn:oasis:names:tc:SAML:2.0:nameid-
format:unspecified">
            2347-0987-4few-cf643-jtf12swe
            </saml:NameID>
            <saml:SubjectConfirmation
                Meth-
od="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <saml:SubjectConfirmationData
                     NotOnOrAfter="2013-12-05T09:35:05Z"
                     Recipi-
ent="https://hub.gov.ch/SAML/ACS/Browser"
                     InResponseTo="aafe-we23-enzz-d3et"/>
                </saml:SubjectConfirmation>
        </saml:Subject>

        <saml:Conditions
            NotBefore="2013-12-05T09:25:05Z"
            NotOnOrAfter="2013-12-05T09:35:05Z">
            <saml:AudienceRestriction>
        <saml:Audience>https://hub.gov.ch</saml:Audience>
            </saml:AudienceRestriction>
        </saml:Conditions>

        <saml:AttributeStatement>
            <saml:Attribute
```

Listing 10: Assertion vom STIAM-Sender zum STIAM-Hub

Beispiel einer Assertion (STIAM-Hub zu STIAM-Empfänger)

Die Assertion vom STIAM-Hub zum STIAM-Empfänger enthält unter anderem ein `<saml:AuthnStatement>` und ein `<saml:AttributeStatement>` Element.

Das `<saml:AttributeValue>` beinhaltet das neue `OriginalIssuer` Attribut. Dieses Attribut enthält den Namen[29] des STIAM-Senders, welcher das Attribut geliefert hat.

[29] Der STIAM-Hub kennt den Namen des STIAM-Senders, welcher das Attribut geliefert hat, anhand der in der KM gespeicherten Informationen.

```
<saml:Assertion
      xmlns:xs="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:ech-0174="http://www.stiam.ch/1.0"
      ID="we34-bhou-pyaq-gbhf"
      Version="2.0"
      IssueInstant="2013-12-05T09:27:05Z">

      <saml:Issuer>https://hub.gov.ch</saml:Issuer>
      <ds:Signature>...</ds:Signature>

      <saml:Subject>
          <saml:NameID Format=
          "urn:oasis:names:tc:SAML:2.0:nameidformat:transient">
           wdrt-6gre-wcbp-ubwq-234gz
          </saml:NameID>
          <saml:SubjectConfirmation
              Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
             <saml:SubjectConfirmationData
                  NotOnOrAfter="2013-12-05T09:37:05Z"
                  Recipient="https://sp.example.ch/SAML/ACS/POST"
                  InResponseTo="ewda-e1df-xydg-xwsq"/>
             </saml:SubjectConfirmation>
      </saml:Subject>

      <saml:Conditions
          NotBefore="2013-12-05T09:27:05Z"
          NotOnOrAfter="2013-12-05T09:37:05Z">
          <saml:AudienceRestriction>
      <saml:Audience>https://sp.example.ch</saml:Audience>
          </saml:AudienceRestriction>
      </saml:Conditions>

  <saml:AuthnStatement
          AuthnInstant="2013-12-05T09:23:50Z"
          SessionIndex="234122">
          <saml:AuthnContext>
              <saml:AuthnContextClassRef>
                  urn:stiam.gov.ch/authenticationassurancelevel3
              </saml:AuthnContextClassRef>
          </saml:AuthnContext>
      </saml:AuthnStatement>

      <saml:AttributeStatement>
          <saml:Attribute
       Name=
   "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml:AttributeValue  xsi:type="xs:StringMaxLength255MinLenght1"
          ech-0174:AQAA-Level="3" OriginalIssuer="STIAM-Sender-X">
          hans@stiam.ch
      </saml:AttributeValue>
         </saml:Attribute>
      </saml:AttributeStatement>
```

Listing 11: Assertion vom STIAM-Hub zum STIAM-Empfänger

6.1.6 Single Logout Request

- Das `<samlp:LogoutRequest>` Element MUSS die Wurzel des Single Logout Requests sein.
- Das `<samlp:LogoutRequest>` Element MUSS ein `<saml:NameID>` und mindestens ein `<samlp:SessionIndex>` Element enthalten.

Beispiel LogoutRequest vom STIAM-Empfänger

Im `<saml:NameID>` Element wird der Benutzer identifiziert, welcher ausgeloggt werden soll.

Die `<samlp:SessionIndex>` Elemente identifizieren die Sessions, die im STIAM-Hub für den Benutzer erstellt wurden und beendet werden sollen.

Falls der Benutzer weitere Sessions bei anderen STIAM-Empfängern hat, würde der STIAM-Hub jedem STIAM-Empfänger einen Logout Request mit den betroffenen Sessions senden, um ein globales Logout zu erreichen (siehe auch Kapitel 4.1.1).

```
<samlp:LogoutRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="d2b7-c388c-ec36-fa7c3"
    Destination="https://hub.gov.ch/SAML/SLO/Browser"
    IssueInstant="2013-12-05T09:50:05Z "
    Version="2.0">

    <saml:Issuer>https://sp.example.ch</saml:Issuer>

    <saml:NameID  Format="urn:oasis:names:tc:SAML:2.0:nameid-
        format:transient">
        wdrt-6gre-wcbp-ubwq-234gz
    </saml:NameID>

    <samlp:SessionIndex>234122</samlp:SessionIndex>

</samlp:LogoutRequest>
```

Listing 12: LogoutRequest

6.1.7 Logout Response

- Das `<samlp:LogoutResponse>` Element MUSS die Wurzel der Logout Response sein.
- Das `<samlp:LogoutResponse>` Element MUSS ein `InResponseTo` Attribut enthalten.

  Das `InResponseTo` Attribut MUSS mit der `ID` des Logout Requests, für den die Logout Response erstellt wurde, übereinstimmen.
- Das `<samlp:Response>` Element MUSS ein `<samlp:Status>` Element enthalten und dieses wiederum ein `<samlp:StatusCode>` Element.

Beispiel LogoutResponse

Das `<samlp:Status>` Element spezifiziert den Status des zu dieser Logout Response gehörenden Single Logout Requests. So werden im `Value` Attribut des `<samlp:StatusCode>` Elementes genaue Informationen über den Status des Single Logout Requests angegeben. Im untenstehenden Listing 13 wurde dieser erfolgreich durchgeführt.

```
<samlp:LogoutResponse
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="k073-kd21-b628-p10d"
    Destination="https://sp.example.ch/SAML/SLO/Response"
    InResponseTo="d2b7-c388c-ec36-fa7c3"
    IssueInstant="2013-12-05T09:51:05Z " Version="2.0">

    <Issuer>https://hub.gov.ch</Issuer>

    <samlp:Status>
        <samlp:StatusCode
            Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>

</samlp:LogoutResponse>
```

Listing 13: LogoutResponse

7 Erweiterungen und Spezialfälle

Dieses Dokument beschreibt die Funktionen und die Kommunikation zwischen den verschiedenen Komponenten im STIAM-Verbund, basierend auf SAML 2.0. Einerseits wird in diesem Dokument davon ausgegangen, dass die Funktionen dieser Komponenten strikt zwischen Konsumenten, Vermittler, Authentifizierungsserver und Attributlieferanten getrennt werden und andererseits, dass eine dieser Komponenten Teil des STIAM-Verbunds sein muss. In diesem Kapitel sollen einige Erweiterungen und Spezialfälle aufgezeigt werden, welche ebenfalls mit einem Hub-'n'-Spoke Modell abgebildet werden können, aber nicht direkt ersichtlich sind.

7.1 Anbindung externer SAML-IdPs

Nicht nur STIAM-IdPs können vom STIAM-Hub zwecks Authentifizierung des Benutzers verwendet werden, sondern auch externe Identity Provider sofern sie SAML 2.0 unterstützen. Solche IdPs müssen zentral im KM registriert werden. Dazu müssen eventuell Anpassungen im Protokoll auf Seiten STIAM-Hub vorgenommen werden. Zum Beispiel kann der von SwissSign betriebene Extended SuisseID IdP eingebunden werden, womit ein weiterer Authentifizierungsserver und Informationslieferant zur Verfügung stehen würde.

7.2 Erweiterter Identity Hub

Es ist durchaus denkbar, dass der STIAM-Hub nicht nur als Authentifizierungsproxy fungiert, sondern selbst als Identity Provider bzw. als Informationslieferant. An den in diesem Dokument beschriebenen Funktionen der Komponenten ändert sich nichts, nur dass die Kommunikationswege kürzer und für den Benutzer einfacher werden. Der Hub kann als Teil des Account Managements weitere Verfahren unterstützen und entsprechende Credentials (PKI-based, OTP, usw.) für einen Benutzer hinterlegen.

7.3 Eingegrenzte Benutzerauthentisierung

Wie in Kapitel 2.3 unter der Funktionsbeschreibung E-DZ-06 beschrieben, kann für eine bestimmte Ressource ein Identity Provider vorgeben werden. Der Hub muss dabei wie in Kapitel 2 Schritt 5 beschrieben den Benutzer direkt zum entsprechenden IdP leiten. Damit entfällt ein weiterer Schritt zur Auswahl eines IdPs durch den Benutzer und er kann sich in Abhängigkeit einer bestimmten Ressource direkt beim richtigen Authentisierungsdienst einloggen.

7.4 Holder-of-Key Profil

In diesem Dokument wird das SAML 2.0 Web Browser SSO-Profile als Standard verwendet. In diesem Profil wird die Assertion in der Authentication Response vom Hub über den Browser an die anfragende RP übermittelt. Obschon die Kommunikationskanäle mit TLS abgesichert sind, kann diese Form der Übertragung ein Sicherheitsrisiko darstellen, da der Browser dazwischen als nicht vertrauenswürdig angesehen werden muss. Die Assertion kann also jederzeit gestohlen und missbraucht werden.

Um diese Schwachstelle zu beseitigen, kennt der SAML 2.0 Standard mit Holder-of-Key einen speziellen Typ ‚SubjectConfirmation'. Hierbei muss der Überbringer der SAML 2.0 As-

sertion sich durch den Besitz eines bestimmten sicheren Schlüssels gegenüber der RP zusätzlich ausweisen.

Grundsätzlich kann die in diesem Dokument vorgestellte Infrastruktur auch mit einem Holder-of-Key Profile umgehen. Dies würde Erweiterungen auf Seiten STIAM-Empfänger und STIAM-Hub bedingen, die im Folgenden beschrieben sind.

7.4.1 Authentifizierung mit dem SAML HoK Profile

Im Folgenden wird das Protokoll zur Authentisierung basierend SAML V2.0 Holder-of-Key Web Browser SSO Profile [13] beschrieben. Grundsätzlich entscheidet der STIAM-Empfänger für welche seiner Ressourcen das HoK-Profil erforderlich ist und legt diese Information im Komponentenmanagement ab. Der STIAM-Hub berücksichtigt diese Information bei der Auswahl der STIAM-IdPs (Discovery-Service) und verwendet seinen HoK Assersion Comsumer Services (ACS Profile) für die Authentication Response vom STIAM-IdP.

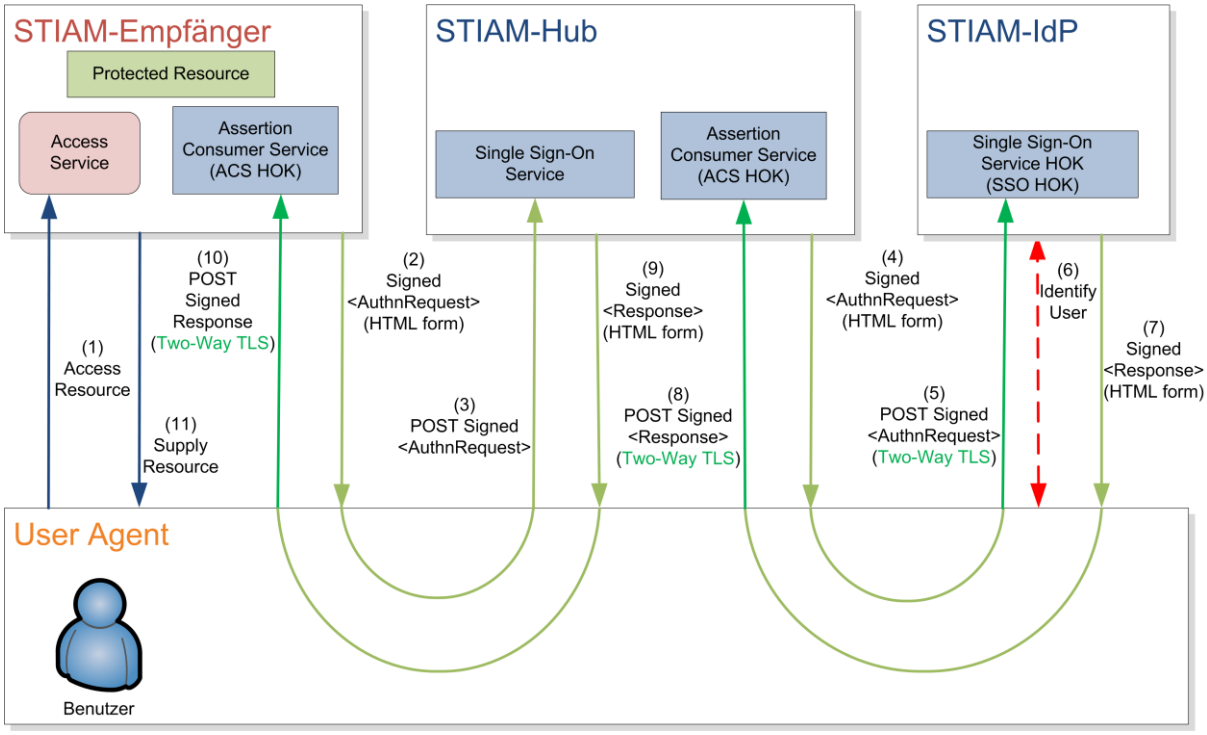

Abbildung 15: Authentifizierungs-Protokoll mit dem SAML HoK-Profile

Schritt	Bemerkung
1	• Der Benutzer möchte auf eine geschützte Ressource zugreifen.
2	• Der STIAM-Empfänger (RP) erstellt einen digital signierten `<samlp:AuthnRequest>` mit Angaben zur Ressource, auf die der Benutzer zugreifen will, und sendet diesen in einer HTML Form zurück an den Browser (User Agent) des Benutzers.
3	• Der Browser sendet den `<samlp:AuthnRequest>` zum SSO-Service des STIAM-Hubs als HTTP POST-Nachricht.
4	• Der Hub validiert den `<samlp:AuthnRequest>` (Issuer, AssertionConsumerServiceURL, Signatur, etc.).

Schritt	Bemerkung
	• Der STIAM-Hub erstellt einen `<samlp:AuthnRequest>` anhand des ausgewählten Identity Providers (hier nicht dargestellt, siehe Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**, S chritte 4-6) und sendet diesen in einer HTML Form zurück zum Browser des Benutzers.
5	• Der Browser sendet eine HTTP POST-Nachricht mit dem `<samlp:AuthnRequest>` zum SSO-Service HOK des IdPs über TLS. In dem TLS Handshake präsentiert der Browser dem STIAM-IdP[30] ein X.509 Zertifikat. • Der IdP validiert den `<samlp:AuthnRequest>` (Issuer, AssertionConsumerServiceURL, Signatur, etc).
6	• Falls der Benutzer nicht durch das präsentierte X.509-Zertifikat authentifiziert wird, fordert der STIAM-IdP diesen zur Authentisierung auf (Username/Password, 2FA, etc.).
7	• Der STIAM-IdP authentifiziert den Benutzer und sendet eine digital signierte `<samlp:Response>` in einer HTML Form zurück an den Browser des Benutzers. • Die `<samlp:Response>` enthält eine signierte `<saml:Assertion>`, welche unter anderen ein `<saml:AuthnStatement>` und ein `<saml:SubjectConfirmation>` Element mit dem Attribut `Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key"` enthält. Ausserdem hat das `<saml:SubjectConfirmation>` ein `<ds:KeyInfo>` Element, welches ein `<ds:X509Data>` Element hat und dieses wiederum ein `<ds:X509Certificate>` Element. Das `<ds:X509Certificate>` enthält das bei dem TLS Handshake vom Browser erhaltene X.509 Zertifikat.
8	• Der Browser leitet die `<samlp:Response>` des IdPs in einer HTTP POST Nachricht zum ACS HOK des STIAM-Hubs über TLS weiter. In dem TLS Handshake präsentiert der Browser dem STIAM-Hub ein X.509 Zertifikat (dasselbe Zertifikat, das dieser dem IdP präsentiert hat). • Der Hub validiert die Response, bzw. die Assertion (Issuer, Signatur, Conditions, etc.). Bei der Validierung MUSS der Hub auch das bei der TLS Handshake erhaltene X.509 Zertifikat mit dem X.509 Zertifikat vergleichen, welches sich im `<ds:X509Certificate>` Element des `<saml:SubjectConfirmation>` Elements der

[30] Das verwendete Zertifikat MUSS KEIN vertrauenswürdiges Zertifikat sein. Allerdings MUSS es im TLS Handshake präsentiert werden. Dies beweist, dass der Browser im Besitz des entsprechenden privaten Schlüssels ist.
Gemäß dem TLS-Protokoll, ist die Validierung des Client-Zertifikats optional. Ebenso ist bei dem Holder-of-Key Web Browser SSO Profile TLS-Client-Authentifizierung OPTIONAL. Deswegen MUSS der TLS-Server so konfiguriert werden, dass die TLS Handshake Verbindung nicht abgebrochen wird, wenn das Client-Zertifikat nicht vertrauenswürdig ist.

Schritt	Bemerkung
	Assertion befindet. Nur wenn diese gleich sind und die anderen Validierungsschritte ebenfalls erfolgreich waren, KANN der Hub der Assertion vertrauen.
9	• Der Hub sendet eine digital signierte `<samlp:Response>` mit einen entsprechenden `<saml:SubjectConfirmation>` Element (wie in Schritt 7) in einer HTML Form zurück an den Browser des Benutzers.
10	• Analog zu Schritt 8.
11	• Der STIAM-Empfänger (RP) verwehrt oder gewährt den Zugriff auf die Ressource.

Bemerkung: In dem oben beschriebenen Anwendungsfall gibt es keine Attributabfrage an einen STIAM-Sender. Da Attributabfragen auf dem SAML Assertion Query/Request Profile [9] mit SOAP Binding basieren, werden diese durch die Verwendung des SAML HoK Profile für die Authentifizierung nicht verändert.

7.4.2 Notwendige Ergänzungen in den Metadaten

Metadaten des STIAM-IdPs

Wenn ein STIAM-IdP das SAML HoK-Profil unterstützen will, MUSS er im `<md:IDPSSODescriptor>` einen zusätzlichen `<md:SingleSignOnService>` definieren (siehe Listing 14).

```
<md:SingleSignOnService
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:hoksso=
      "urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
    hoksso:ProtocolBinding=
      "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Binding=
      "urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
    Location="https://idp.example.ch/SAML/SSOHOK/Browser" />
```

Listing 14: HoK SSO Service am STIAM-IdP

Metadaten des STIAM-Hubs

Damit der STIAM-Hub das SAML HoK-Profil unterstützt, MUSS er in seinem `<md:SPSSODescriptor>` einen zusätzlichen `<md:AssertionConsumerService>` definieren (siehe Listing 15).

```
<md:AssertionConsumerService
    index="2" isDefault="false"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:hoksso=
       "urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
    hoksso:ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Binding="urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
    Location="https://hub.gov.ch/SAML/ACSHOK/POST"/>
```

Listing 15: HoK ACS Service am STIAM-Hub

Metadaten des STIAM-Empfängers

Im `<md:SPSSODescriptor>` des STIAM-Empfängers MUSS ein entsprechender `<md:AssertionConsumerService>` definiert werden (siehe Listing 16). Dieser muss auch als `AssertionsConsumerServiceURL` im Authentication Request an den STIAM-Hub angeben werden.

```
<md:AssertionConsumerService
    index="2" isDefault="false"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns:hoksso=
      "urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
    hoksso:ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Binding="urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
    Location="https://sp.example.ch/SAML/ACSHOK/POST"/>
```

Listing 16: HoK ACS Service am STIAM-Empfänger

7.4.3 HoK-Assertions

Im Folgenden ist ein Beispiel einer Assertion gezeigt, die als Teil einer Authentication Response vom STIAM-IdP zum STIAM-Hub gesendet wird.

```
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        ID="ojqx-mlhj-xydg-xdew"
        Version="2.0"
        IssueInstant="2013-12-05T09:23:59Z">
   <saml2:Issuer>https://idp.example.ch</saml2:Issuer>
   <!-- Signature of the IdP -->
   <ds:Signature>...</ds:Signature>
   <saml2:Subject>
      <saml2:NameID
             Format="urn:oasis:names:tc:SAML:2.0:nameidformat:persistent">
             3f7b3dcf-1674-4ecd-92c8-1544f346baf8
       </saml2:NameID>
        <!-- Subject Confirmation Method: holder-of-key -->
        <!-- The Assertion was created for the Hub -->
        <saml2:SubjectConfirmation
             Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
           <saml2:SubjectConfirmationData
                 xsi:type="saml2:KeyInfoConfirmationDataType"
               NotOnOrAfter="2013-12-05T09:33:59Z"
               Recipient="https://hub.gov.ch/SAML/ACSHOK/POST"
               InResponseTo="mkqs-ezew-qplo-snrt">
              <ds:KeyInfo>
                <ds:X509Data>
                    <!--  X509 Certificate from TLS Handshake -->
                  <ds:X509Certificate>...</ds:X509Certificate>
                </ds:X509Data>
              </ds:KeyInfo>
             </saml2:SubjectConfirmationData>
        </saml2:SubjectConfirmation>
      </saml2:Subject>
      <saml2:Conditions
            NotBefore="2013-12-05T09:23:59Z"
            NotOnOrAfter="2013-12-05T09:33:59Z">
          <saml2:AudienceRestriction>
                          <saml2:Audience>https://hub.gov.ch</saml2:Audience>
          </saml2:AudienceRestriction>
       </saml2:Conditions>
       <saml2:AuthnStatement
             AuthnInstant="2013-12-05T09:23:50Z"
             SessionIndex="234122">
           <saml2:AuthnContext>
                 <saml2:AuthnContextClassRef>
                     https://stiam.gov.ch/authenticationassurancelevel3
                 </saml2:AuthnContextClassRef>
           </saml2:AuthnContext>
       </saml2:AuthnStatement>
</saml2:Assertion>
```

Listing 17: HoK Assertion

8 Haftungsausschluss/Hinweise auf Rechte Dritter

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellt, oder welche **eCH** referenziert, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

9 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

eCH-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

Anhang A – Referenzen und Bibliographie

[1] IETF. (1997) www.ietf.org. [Online]. http://www.ietf.org/rfc/rfc2119.txt

[2] eCH. (2013, Dez.) eCH-0107 IAM-Gestaltungsprinzipien. [Online]. http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0107&documentVersion=2.0

[3] eCH. (2014, Jun) eCH-0167 SuisseTrustIAM-Rahmenkonzept. [Online]. http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0167&documentVersion=1.0

[4] eCH, "eCH-0168 SuisseTrustIAM technische Architektur und Prozesse," eCH, 2014.

[5] OASIS. SAML. [Online]. http://saml.xml.org/saml-specifications

[6] (2014, Sep.) eCH-0169: SuisseTrustIAM-Geschäftsarchitektur. [Online]. http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0169

[7] eCH. (2014, Jun) eCH-0170 eID Qualitätsmodell. [Online]. http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0170&documentVersion=1.0

[8] eCH, "eCH-0171 Qualitätsmodell der Attributwertbestätigung zur eID,".

[9] OASIS. (2005, March) Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0. [Online]. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[10] OASIS. (2005, March) Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. [Online]. http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf

[11] OASIS. (2005, March) Saml-Core-2.0 for the OASIS Security Assertion Markup Language (SAML) V2.0. [Online]. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[12] (2010) SAML V2.0 Identity Assurance Profiles Version 1.0. [Online]. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile.pdf

[13] OASIS. (2010, August) SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0. [Online]. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.pdf

Anhang B – Mitarbeit & Überprüfung

Laube-Rosenpflanzer Annett	Berner Fachhochschule, eCH Fachgruppe IAM
Hassenstein Gerhard	Berner Fachhochschule
Yandy Piedra Guerra	Berner Fachhochschule
Sabine Zumstein	Berner Fachhochschule
Adrian Berger	Ergon Informatik AG, eCH Fachgruppe IAM
Torsten Grouner	ISB
Martin Topfel	Berner Fachhochschule
Hans Burger	Adnovum

Anhang C – Abkürzungen

IdP	Identity Provider
AA	Attribute Authority
RP	Relying Party
SSO	Single-Sign-On
SLO	Single Logout
SAML	Security Assertion Markup Language
HoK	Holder-of-Key
IAM	Identity und Access Management
KM	Komponenten-Management
OID	Object Identifier
XSD	XML Schema Definition
URL	Uniform Resource Locator
URI	Uniform Resource Indicator
UIR	User Identifier Repository
STIAM	SuisseTrustIAM
SysAdmin	Systemadministrator
OrgSysAdmin	Systemadministrator einer Organisation
OV	Organisationsverantwortlicher
CSP	Certification Service Provider
CA	Certification Authority
RLM	Reporting, Logging, Monitoring
MA	Metadata-Aggregator
MDR	Metadata Registry
TLS	Transport Layer Security

Dieses Dokument verwendet grundsätzlich die Begriffsdefinitionen aus eCH-0107 [2] und eCH-0168 [4].

Anhang E – Abbildungsverzeichnis

Anhang F – Verzeichnis der Listings

Anhang G – Tabellenverzeichnis