eCH-0225 – Vermittlerbasierte Identity Federations – Implementierung mit OIDC

Name	Vermittlerbasierte Identity Federations – Implementierung mit OIDC
eCH-Nummer	eCH-0225
Kategorie	**Standard**
Reifegrad	Definiert
Version	1.0
Status	Genehmigt
Beschluss am	2020-06-04
Ausgabedatum	2020-06-05
Ersetzt Version	-
Voraussetzungen	eCH-0107, eCH-0219, eCH-0224, eCH-0170
Beilagen	keine
Sprachen	Deutsch (Original), Französisch (Übersetzung)
Autoren	Fachgruppe IAM Bojan Leimer, BFH TI, bojan.leimer@bfh.ch Gerhard Hassenstein, BFH TI, gerhard.hassenstein@bfh.ch Annett Laube-Rosenpflanzer, BFH TI, annett.laube@bfh.ch Marc Kunz, BFH TI, marc.kunz@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

Dieser Standard beschreibt die Umsetzung der vermittlerbasierten Identity Federation Modelle aus dem Standard eCH-0224 mit Hilfe von OpenID Connect (OIDC). Ziel ist die Sicherstellung von Interoperabilität vor allem für Relying Parties in den Szenarien G2G, G2B und G2C. Dazu werden die notwendigen Schnittstellen und Protokolle definiert und besonders auf die notwendigen Anpassungen von OIDC hingewiesen. Der Standard richtet sich damit vorwiegend an IT-Architekten und Entwickler von Komponenten in der Identity Federation.

Inhaltsverzeichnis

1 Status 6
2 Einleitung 6
2.1 Ziel des Dokumentes 6
2.2 Einordnung 6
2.3 Zielgruppe 7
2.4 Anwendungsgebiet 7
2.5 Abgrenzung 8
2.6 Normativer Charakter der Kapitel 8
3 Identity Federation basierend auf OpenID Connect 9
3.1 Zur Verfügung stehende Authentication Flows 9
3.2 Tokens 9
3.2.1 ID Token 9
3.2.2 Access Token 11
3.3 Claims 12
3.3.1 Normal Claims 12
3.3.2 Aggregated Claims 12
3.3.3 Distributed Claims 12
3.4 OIDC Endpoints 12
3.5 Identifikatoren 13
3.6 Single Sign-on und Single Logout (deskriptiv) 13
4 Allgemeine Richtlinien 15
5 Schnittstellen der Vermittlermodelle 18
5.1 Authentifizierung 18
5.2 Authentifizierung mit Attributübermittlung 18
5.2.1 Vermittler Double-Blinding 19
5.2.2 Vermittler Offene Quellen 20
5.2.3 Vermittler Vertraulichkeitsschutz 20
6 Schnittstellen für die Relying Party (RP) 22
6.1 Authentifizierung 22
6.1.1 Authentifizierungsanfrage an den Vermittler 23
6.1.2 Authentifizierungsbestätigung vom Vermittler 24
6.2 Authentifizierung mit Attributübergabe 26
6.2.1 Attributanfrage an Vermittler 26
6.2.2 Attributbestätigung als Normal Claim 27
6.2.3 Attributbestätigung als Aggregated Claim 27

6.2.4 Attributbestätigung als Distributed Claim 28
7 Schnittstellen für Vermittler 30
7.1 Authentifizierung 31
7.1.1 Authentifizierungsanfrage der Relying Party (RP) 31
7.1.2 Authentifizierungsanfrage an Identity Provider (IdP) 32
7.1.3 Authentifizierungsbestätigung vom Identity Provider (IdP) 32
7.1.4 Authentifizierungsbestätigung an Relying Party (RP) 34
7.2 Authentifizierung mit Attributübergabe 35
7.2.1 Attributanfrage der Relying Party (RP) 36
7.2.2 Attributanfragen des Vermittlers 36
7.2.3 Attributbestätigungen an den Vermittler 38
7.2.4 Attributbestätigungen an RP 39
8 Schnittstellen für Identity Provider (IdP) 43
8.1 Authentifizierungsanfrage von Vermittler 43
8.2 Authentifizierungsbestätigung an Vermittler 44
9 Schnittstellen für Attribut Provider (AP und IdP/AP) 46
9.1 Attributanfrage von Vermittler 46
9.1.1 Attributanfrage an einen IdP/AP 47
9.1.2 Attributanfrage an einen AP 47
9.2 Attributanfrage von Relying Party (RP) 48
9.2.1 Anfrage mit Access Token 48
9.2.2 Anfrage mit JWT Access Token 49
9.3 Attributbestätigungen 49
9.3.1 Attributbestätigung an Vermittler (Normal Claim) 50
9.3.2 Attributbestätigung an Vermittler (Aggregated Claim) 50
9.3.3 Attributbestätigung an RP 51
10 Message Richtlinien 52
10.1 Authorization Code Flow 52
10.1.1 Authentication Request 52
10.1.2 Authentication Response 52
10.1.3 Token Request 52
10.1.4 Token Response 53
10.1.5 UserInfo Request 53
10.1.6 UserInfo Response 53
10.2 Hybrid Flow 53
10.2.1 Authentication Request 53

10.2.2 Authentication Response 54
10.2.3 Weitere Hybrid Flow Messages 54
10.3 Introspection 54
10.3.1 Introspection Request 54
10.3.2 Introspection Response 54
10.4 Weitere Messages (deskriptiv) 54
10.5 Fehlermeldungen 55
11 Metadaten Richtlinien 56
11.1 Identity Provider 56
11.2 Relying Party 57
11.3 Vermittler 57
11.4 Attribute Provider 58
12 Haftungsausschluss/Hinweise auf Rechte Dritter 59
13 Urheberrechte 59
Anhang A – Referenzen & Bibliographie 60
Anhang B – Mitarbeit & Überprüfung 61
Anhang C – Abkürzungen und Glossar 61
Anhang E – Abbildungsverzeichnis 62
Anhang F – Listingsverzeichnis 63
Anhang G – Tabellenverzeichnis 65

Hinweis

Aus Gründen der besseren Lesbarkeit und Verständlichkeit wird im vorliegenden Dokument bei der Bezeichnung von Personen ausschliesslich die maskuline Form verwendet. Diese Formulierung schliesst Frauen in ihrer jeweiligen Funktion ausdrücklich mit ein.

1 Status

Genehmigt: Das Dokument wurde vom Expertenausschuss genehmigt. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

2 Einleitung

Eine Identity Federation im E-Government ermöglicht es Behörden ihre Leistungen online den Mitarbeitern anderer Behörden und den Bürgern sowie Organisationen und Unternehmen ihres Landes zur Verfügung zu stellen (siehe eCH-0219 IAM Glossar [1] Kapitel 2.62). Die Behörden delegieren dazu die Authentifizierung und die Bestätigung von Attributen an verschiedene IAM-Dienstanbieter. Grundlage der Delegation ist gegenseitiges Vertrauen, das durch rechtliche, organisatorische/architektonische und technischen Vereinbarungen abgestützt wird.

Im Standard eCH-0224 [2] wurde anhand von geschärften Anforderungen aus dem E-Government zusammen mit Anforderungen zum Schutz der Privatsphäre der Subjekte, der Raum der möglichen Identity Federation Systeme auf den folgenden drei vermittlerbasierten Modellen eingeschränkt:

- Vermittler Double Blinding,
- Vermittler Offene Quellen,
- Vermittler Vertraulichkeitsschutz.

Die Umsetzung dieser drei Modelle mit Hilfe von OpenID Connect (OIDC) [3] wird in diesem Standard beschrieben.

2.1 Ziel des Dokumentes

Ziel ist die Sicherstellung von Interoperabilität zwischen den verschiedenen Komponenten vermittlerbasierten Identity Federations, vor allem für Relying Parties, in den Szenarien G2G, G2B und G2C. Dazu werden die notwendigen Schnittstellen und Protokolle defniert und besonders auf die notwendigen Anpassungen bzw. Erweiterungen von OIDC hingewiesen.

2.2 Einordnung

Unter dem Standard eCH-0107 [4] positionieren sich Konzepte für föderierte IAM-Lösungen und ergänzende Hilfsmittel. Die Konzepte sind konkrete Beschreibungen, wie ein IAM-Lösungsvorschlag aussieht, und beinhalten Teilkonzepte und Architekturen, die für die Umsetzung berücksichtigt werden müssen. Den Konzepten werden Hilfsmittel zur Seite gestellt, die ergänzende Informationen zur Verfügung stellen. Diese können aber für mehr als ein Konzept relevant sein. Die in diesem Dokument dargestellten Qualitäts- und Maturitätsmodelle sind Beispiele für solche Hilfsmittel. Die Liste dieser Hilfsmittel ist nicht abschliessend.

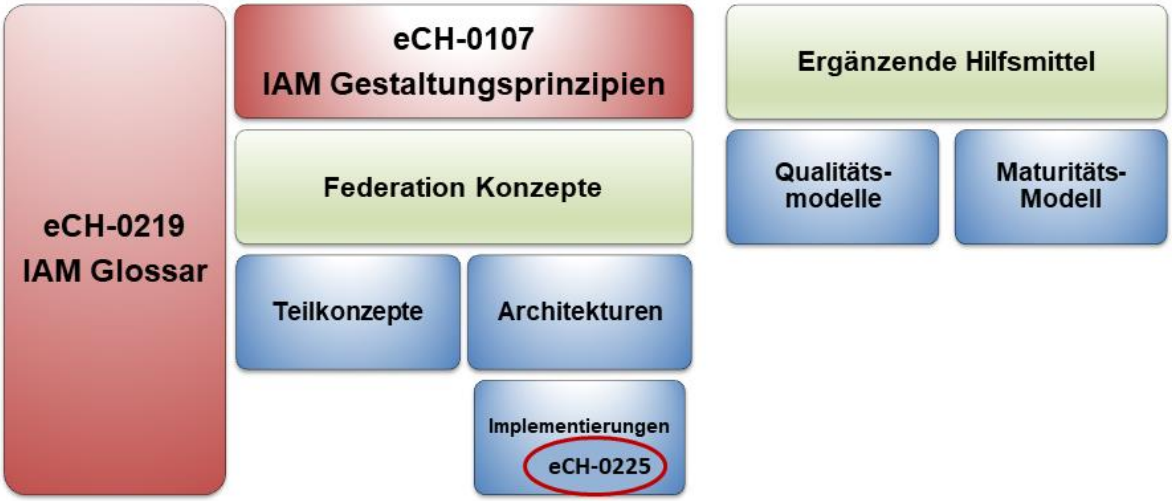

Abbildung 1: Einordnung des Standards eCH-0225

Der vorliegende Standard eCH-0225 beschreibt die Umsetzung der in Standard eCH-0224 [2] beschriebenen Architekturmodelle auf der technologischen Basis von OIDC [3].

2.3 Zielgruppe

Der Standard richtet sich vorwiegend an IT-Architekten und Entwickler von Identity Federations oder einzelnen Komponenten (RP, Vermittler, IdP, AP) von Identity Federation Systemen im E-Government. Er setzt Kenntnis der Vermittlermodelle aus eCH-Standard eCH-0224 [2] sowie fundiertes Wissen über den OIDC-Standard [3] voraus.

Die Kapitel 3 bis 4 definieren die Grundlagen eines Identity Federation Systems mit OIDC und sollten von IT-Architekten und Entwicklern studiert werden.

Kapitel 5 definiert die Umsetzung der Vermittlermodelle. Dieses Kapitel ist für IT-Architekten sowie Entwickler gleichermassen wichtig. Anhand des zu verwendeten Vermittlermodells wird entsprechend pro Komponente auf die zu implementierenden Schnittstellen verwiesen.

Die Kapitel 6, 7, 8 und 9 definieren die Schnittstellen der Komponenten und richtet sich an die Entwickler, die die entsprechenden Schnittstellen implementieren. IT-Architekten sollten diese Kapitel dennoch studieren, um das Zusammenspiel der Komponenten zu verstehen.

Kapitel 10 und 11 liefen den Entwicklern weitere Informationen zu den Implementierungen der Schnittstellen.

2.4 Anwendungsgebiet

Der Standard beschreibt die Umsetzung der vermittlerbasierten Identity Federation Modelle aus eCH-0224 [2]. Er ist eine Richtlinie zur Implementierung der notwendigen Schnittstellen und Protokolle für die einzelnen Identity Federation Komponenten, um die Interoperabilität zwischen verschiedenen Anbietern sicher zu stellen.

2.5 Abgrenzung

Dieser Standard beschreibt die Umsetzung der in eCH-0224 [2] definierten Anforderungen mit den heute technischen Mitteln auf der Basis von OIDC [3]. Es werden ausschliesslich die vermittlerbasierten Identity Federation Modelle aus eCH-0224 [2] berücksichtigt. Andere Modelle oder Varianten von OIDC [3] oder OAuth [5] werden nicht adressiert.

Die Umsetzung des Vertraulichkeitsschutzes (gegenüber dem Vermittler) betrifft nur die Attributbestätigungen, d.h. es werden keine verschlüsselten Authentifizierungsbestätigungen (ID Token) zwischen IdP und RP übertragen.

Falls ein IAM-System gemäss BGEID [6] umgesetzt wird, müssen die gesetzlichen Bestimmungen, sowie die dazugehörigen Verordnungen und Ausführungsbestimmungen eingehalten werden, bevor die in den eCH-Standards definierten Richtlinien angewendet werden können. Bei Konflikten hat das BGEID [6] Vorrang.

2.6 Normativer Charakter der Kapitel

Die Kapitel des vorliegenden Standards sind von normativem oder auch deskriptivem Charakter. Folgende Tabelle definiert die Einordnung der Kapitel.

Kapitel	Beschreibung
2 Einleitung	**Deskriptiv**
3 Identity Federation basierend auf OpenID Connect	**Normativ**
3.6 States, Single SignOn und Single Logout	**Deskriptiv**
4 Allgemeine Richtlinien	**Normativ**
5 Schnittstellen der Vermittlermodelle	**Normativ**
6 Schnittstellen für die Relying Party (RP)	**Normativ**
7 Schnittstellen Vermittler	**Normativ**
8 Schnittstellen für Identity Provider (IdP)	**Normativ**
9 Schnittstellen für Attribut Provider (AP und IdP/AP)	**Normativ**
10 Message Richtlinien	**Normativ**
10.4 Weitere Messages	**Deskriptiv**
11 Metadaten Richtlinien	**Normativ**

Anhang A und Anhang C sind ebenfalls normativ. Alle anderen Anhänge dieses Standards sind deskriptiv.

3 Identity Federation basierend auf OpenID Connect

Dieser Standard baut auf den Standards von OpenID Connect [3] und OAuth 2.0 [5] auf. Das Protokoll von OpenID Connect ist sehr flexibel und lässt für eine Implementation viel Spielraum. Deshalb werden in diesem Standard, die Möglichkeiten von OpenID Connect aufs wesentliche eingeschränkt, sodass die Anforderungen und Modelle vom eCH-0224 [2] implementiert werden können. Die folgenden Unterkapitel beschreiben die Elemente von OpenID Connect, die in diesem Standard zum Einsatz kommen.

3.1 Zur Verfügung stehende Authentication Flows

OpenID Connect unterstützt drei unterschiedliche Authentication Flows mit jeweils anderen Eigenschaften und Verwendungszwecken (siehe Beschreibung der Flows in Kapitel 3 der OpenID Connect Spezifikation [3]). In Tabelle 1 werden die Flows für die Protokollabschnitte 1 und 2 (Die Protokollabschnitte sind in Kapitel 8.1.1 vom eCH-0224 [2] definiert) aufgelistet. Dabei werden die Flows ausgeschlossen, die die Anforderungen aus eCH-0107 [4] und eCH-0224 [2] nicht erfüllen.

Nr.	*response_type*	Flow	Anforderungskonflikt
1.	code	Authorization Code Flow	-
2.	id_token	Implicit Flow	224-LE-5 [2], IAM-11 [4] und LB-15 [4]
3.	id_token token	Implicit Flow	IAM-11 [4] und LB-15 [4]
4.	code id_token	Hybrid Flow	IAM-11 [4] und LB-15 [4]
5.	code token	Hybrid Flow	-
6.	code id_token token	Hybrid Flow	IAM-11 [4] und LB-15 [4]

Tabelle 1: Zur Verfügung stehende Authentication Flows

Flows, bei denen in ‘response_type’ ‘id_token’ enthalten ist, DÜRFEN NICHT verwendet werden. Bei diesen Flows wird ein ID Token an die RP gesendet, bevor die RP authentisiert wurde. Dies widerspricht einigen Anforderungen (siehe Tabelle 1).

Es MÜSSEN gemäss Tabelle 1 entweder der Authorization Code Flow (Nr. 1) oder Hybrid Flow (Nr. 5) verwendet werden. Alle anderen Flows sind ausgeschlossen und werden in diesem Dokument nicht weiter behandelt.

3.2 Tokens

In den folgenden Abschnitten werden die Formate von ID Token und Access Token definiert.

3.2.1 ID Token

Ein ID Token ist eine Authentifizierungsbestätigung (im JWT-Format) und MUSS vom Aussteller signiert werden. Optional KANN das ID-Token für den Empfänger verschlüsselt sein. Wird das ID Token signiert und verschlüsselt, ist das ID Token erst zu signieren und dann zu verschlüsseln, wobei dabei ein verschachteltes JWT entsteht (nested JWT).

Nachfolgend sind die im ID Token enthaltenen Parameter definiert. In Listing 1 ist der Inhalt

eines ID Tokens als Beispiel dargestellt. Die Anpassungen bei einer Verwendung eines Hybrid Flows sind in Kapitel 3.2.1.1 beschrieben.

Parameter	Beschreibung	Referenz
iss	MUSS die URL des Ausstellers (Issuer) des Tokens aufweisen.	OIDC
sub	MUSS einen eindeutigen Subjekt-Identifier enthalten. Um die Privatsphäre des Subjektes zu schützen SOLLTE für jede RP ein neuer Identifier verwendet werden (siehe auch Kapitel 3.5).	OIDC
aud	MUSS die *client_id* der RP (Audience) enthalten, die das ID Token angefordert hat. Beim Wert dieses Parameters handelt es sich um ein Array, das mehrere *client_id*s enthalten KANN.	OIDC
exp	MUSS die Gültigkeitsdauer des Tokens definieren.	OIDC
iat	MUSS die Erstellungszeit des JWTs enthalten.	OIDC
acr	MUSS die Vertrauensstufe enthalten, mit der das Subjekt beim Issuer (*iss*) authentifiziert wurde.	Richtlinie 4[1]
nonce	KANN als zusätzliche Sicherheit vor Replay Attacken verwendet werden und MUSS – falls enthalten – identisch mit dem *nonce* Parameter des Authentication Requests sein.	OIDC
source_iss	KANN die URL der effektiv ausstellenden Instanz des ID Tokens enthalten. Beim Vermittlerszenario mit "Offene Quellen", MUSS dieser Parameter den ausstellenden IdP enthalten.	Erweiterung zu OIDC
auth_time	KANN gemäss der OIDC Spezifikation [3] verwendet werden.	OIDC
amr	KANN gemäss der OIDC Spezifikation [3] verwendet werden.	OIDC
azp	DARF NICHT verwendet werden.	Einschränkung zu OIDC

Tabelle 2: ID Token - Parameter

Das ID Token DARF KEINE Attribute enthalten.

```
{
 "iss": "https://oidc-idp.example.com",
 "sub": "789456123",
 "aud": "123456789",
 "nonce": "n-0S6_WzA2Mj",
 "exp": 1511430048,
 "iat": 1511430348,
 "acr": "ech0170.vs1"
}
```

Listing 1: Beispiel eines generellen ID Tokens

3.2.1.1 Anpassungen bei Verwendung des Hybrid Flows

Bei Verwendung des Hybrid Flows MÜSSEN die nachfolgenden Parameter zum ID Token hinzugefügt werden.

[1] Siehe Kapitel 4: Qualitätsmodell für die Authentifizierung

Parameter	Beschreibung	Referenz
nonce	MUSS enthalten sein.	OIDC
at_hash	MUSS angegeben werden, wenn ein Access Token retourniert wird (siehe Kapitel 3.3.2.11 der OpenID Connect Spezifikation [3]).	OIDC
c_hash	Gemäss Kapitel 3.3.2.11 der OpenID Connect Spezifikation [3] ist dieser Parameter erforderlich, wenn bei der Authentifizierungsanfrage 'code' im Parameter *response_type* enthalten ist. Daher MUSS dieser Parameter ebenfalls im ID Token enthalten sein.	OIDC

Tabelle 3: ID Token - Parameter im Hybrid Flow

3.2.2 Access Token

Das Access Token ermöglicht das Abfragen von Attributen (Claims) bei UserInfo Endpoints (z.B. bei einem AP). Die RP (bzw. der Vermittler) erhält, nach erfolgreicher Authentifizierung des Subjektes, ein Access Token. Das Access Token DARF NICHT an unbefugte Dritte weitergegeben werden.

Es gibt zwei Typen von Access Token:

- Ersteres besteht aus einer Zeichenkette gemäss OAuth Spezifikation [7] Kapitel 1.4.
- Der zweite Typ ist ein JWT Access Token und ist im Kapitel 3.2.2.1 definiert.

3.2.2.1 JWT Access Token

Ein JWT Access Token enthält weitere Informationen zum Subjekt. Die Zeichenkette MUSS aus Daten und Signatur bestehen (siehe Appendix A.12 in RFC6749 [7]). Das JWT Access Token MUSS vom Aussteller (Issuer) signiert werden, damit der Empfänger (Audience) die Integrität und Authentizität des Tokens prüfen kann. Die Parameter desJWT Access Token sind wie folgt definiert. In Listing 2 ist ein Beispiel eines JWT Access Token dargestellt.

Parameter	Beschreibung
iss	MUSS den Identifier des Ausstellers (z. B. URL des Issuers) des Access Tokens enthalten.
aud	MUSS den Identifier der Audience (z. B. URL des APs) beinhalten, für den das Access Token ausgestellt wurde.
sub	MUSS einen eindeutigen Subjekt-Identifier enthalten.
iat	MUSS einen Integer Timestamp der Ausstellungszeit enthalten.
exp	MUSS einen Integer Timestamp enthalten. Damit wird die Gültigkeitsdauer des JWT Access Tokens angegeben.
scope	MUSS die von der Audience abgefragten Scopes beinhalten. Bei mehreren Scope-Werten sind diese durch einen Leerschlag getrennt aufzuführen. Mögliche Scope-Werte sind in den Kapiteln 3.1.2.1 und 5.4 der OpenID Connect Spezifikation [3] zu finden.

Tabelle 4: JWT Access Token - Parameter

```
{
 "alg": "RS256",
 "typ": "JWT"
}
{
 "iss": "https://vermittler.example.com",
 "aud": "https://oidc-ap.example.com/attribute-src",
 "sub": "897248261001",
 "iat": 1516440389,
 "exp": 1516440419,
 "scope": "email attribute"
}
```

Listing 2: Beispiel eines JWT Access Tokens

3.3 Claims

Claims stehen in der OpenID Connect Spezifikation [3] für Attributbestätigungen. Die drei Claims-Typen: *Normal Claims*, *Aggregated Claims* und *Distributed Claims* sind in Kapitel 5.6 der OpenID Connect Spezifikation [3] definiert.

3.3.1 Normal Claims

Bei Normal Claims stammen alle Attribute aus einer Quelle. Die Attribute des Subjekts werden in ein JSON Objekt verpackt.

3.3.2 Aggregated Claims

Aggregated Claims KÖNNEN verwendet werden, um Attribute von der ursprünglichen Quelle zu signieren (sind somit verifizierbar) und/oder Attribute von mehreren Quellen zusammenzufassen und diese als eine kombinierte Attributbestätigung zu retournieren.

3.3.3 Distributed Claims

Distributed Claims KÖNNEN verwendet werden, um Attribute vom Empfänger direkt oder Attribute von mehreren Quellen abzufragen. Anstelle der Attributwerte, erhält der Empfänger Referenz-URL(s).

3.4 OIDC Endpoints

Tabelle 5 fasst die aus Kapitel 5 des eCH-0224 Standards [2], den RFC 6749 [7] und RFC 7662 [8] definierten Endpoints zusammen. Dabei wird aufgezeigt, welche Komponente den jeweiligen Endpoint implementiert und welche Endpoints für die Komponenten optional sind.

Die Definitionen der OpenID Connect Endpoints sind den folgenden Spezifikationen zu entnehmen:

- **Autorization Endpoint**: RFC 6749 [7] Kapitel 3.1.
- **Token Endpoint**: RFC 6749 [7] Kapitel 3.2.
- **UserInfo Endpoint**: OpenID Connect Spezifikation [9] Kapitel 5.3.
- **Introspection Endpoint**: RFC 7662 [8] Kapitel 2.

		Anwendungsfälle nach eCH-0224 [2]	
		Reine Authentifizierung	Authentifizierung mit Attributübergabe
OpenID Connect Endpoints	Autorization Endpoint	Vermittler, IdP	
	Token Endpoint	Vermittler, IdP	
	UserInfo Endpoint	-	Vermittler, IdP/AP, AP
	Introspection Endpoint (opt)	-	Vermittler

Tabelle 5: OIDC Endpoints zu Anwendungsfällen

3.5 Identifikatoren

Die OpenID Connect Spezifikation definiert zwei Arten von Identifikatoren, den *public identificator* und den *pairwise identificator*, wobei ersterer einem verteilten Identifikator (Distributed ID) nach Kapitel 5.1.3.1 des eCH-0224 [2] und letzterer einem Relying Party gebundenen Identifikator (Persistent ID) nach Kapitel 5.1.3.2 des eCH-0224 Standards [2] entspricht. Der *public* Identifikator DARF aufgrund des fehlenden Privatsphärenschutzes (siehe 224-LB-4 'Unverkettbarkeit' aus eCH-0224 [2]) NICHT verwendet werden.

Zum Schutz der Privatsphäre von Subjekten, werden in diesem Standard ausschliesslich *Pairwise Pseudonymous Identifier* (PPID) als Identifikatoren verwendet. Mit dem PPID erhält jede RP für jedes Subjekt einen für sie eindeutigen Identifier, der sich nicht mit den Identifier anderer RPs korrelieren lässt. Um die unterschiedlichen Identifikatoren (aus Kapitel 5.1.3 des eCH-0224 [2]) zu unterstützen, ist eine erweiterte Definition der Identifikatortypen notwendig und wird in Tabelle 6 beschrieben.

Subject Identifier Type	eCH-0224 Bezeichnung	Beschreibung
public	Distributed ID	Verteilter Identifikator. DARF **NICHT** verwendet werden.
pairwise	Persistent ID	Einmalig berechneter Identifier pro RP und Subjekt.
transient	Transient ID	Pro Authentifizierung (Session) eines Subjektes wird ein neuer Identifier berechnet.

Tabelle 6: Identifikatoren für Subjekte

3.6 Single Sign-on und Single Logout (deskriptiv)

Für Single Sign-On und Single Logout mit OIDC gibt es bisher kaum Erfahrungen und noch keine etablierten Standards, schon gar nicht in Kombination mit vermittlerbasierten Identity Federations. Da das Thema «Session Handling» sehr komplex ist und eine vollständige Beschreibung sehr umfangreich wäre, wird hier nur kurz darauf eingegangen. Das Thema

«Session Handling, inkl. Single Sign-On und Single Logout» sollte – wenn mehr Erfahrungen mit dem Thema gesammelt wurden - in einem eigenen Standard ausführlich behandelt werden.

Single Logout

Für Single Logout bei OIDC ist der Standard *Session Management Single Logout* [10] im Moment nur als Draft verfügbar. Dieser erweitert die OIDC Core Spezifikation [3], um das durchgängige Monitoring des Authentifizierungsstatus der Endbenutzer durch einen IdP. Damit wird einer RP ermöglicht einen Benutzer abzumelden, sobald sich dieser beim IdP abgemeldet hat.

In einer Identity Federation übernimmt der Vermittler die Rolle des IdPs gegenüber der RP und sollte daher diesen Standard implementieren.

Ergänzt wird der Draft *Session Management Single Logout* [10] durch zwei weitere Spezifikationen, die ebenfalls nur einen Draft-Status haben:

OpenID Connect Front-Channel Logout [11]
OpenID Connect Back-Channel Logout [12]

Single Sign-on

Für Single Sign-on gibt es derzeit keinen OIDC Standard oder Draft, der das Session Handling eines IdP für Single Sign On (siehe auch eCH-0224 Standards [2], Kap. 3.3.1) spezifiziert. Das im Draft *Session Management Single Logout* [10] beschriebene Session Handling könnte erweitert und angepasst werden, um die notwendigen Funktionalitäten zu ermöglichen.

In einer Identity Federation übernimmt der Vermittler die Rolle des IdPs gegenüber der RP und müsste dann diese Anpassungen implementieren.

4 Allgemeine Richtlinien

Basierend auf den Anforderungen für eine Identity Federation im E-Government nach eCH-0224 [2] und eCH-0107 [4] MÜSSEN für diesen Standard zusätzlich die folgenden Richtlinien eingehalten werden.

Aus diesen Richtlinien resultieren Einschränkungen bzw. Erweiterungen des OIDC Standards [3]. Ausserdem wird definiert, welche Anforderung mit welcher technischen Massnahme erfüllt wird.

Richtlinie 1 – Authentifizierung der anfragenden Instanz: Die angefragte Instanz MUSS die anfragende Komponente vor dem Retournieren von Identitäts-, Attribut- oder Authentifizierungsinformationen eines Subjektes, authentifizieren.

Die Authentifizierung MUSS gemäss einer aus Kapitel 9 des OpenID Connect Standards [3] definierten Methode erfolgen (siehe auch Kapitel 10.1.3). Die Methode *none* DARF NICHT verwendet werden.

Erfüllte Anforderungen: 224-LE-5, IAM-11 und LB-15.

Richtlinie 2 – Authentizität und Integrität der Antwort: Alle retournierten ID Tokens und Claims MÜSSEN von der ausstellenden Instanz signiert werden.

Ist die abnehmende Instanz der Vermittler, MUSS dieser die Signatur prüfen. Die RP SOLLTE eine Prüfung der Signatur vornehmen.

In allen beteiligten Instanzen SOLLTEN empfohlene kryptographische Algorithmen und Schlüssellängen angewendet werden.[2]

Erfüllte Anforderung: IAM-9

Richtlinie 3 – Vertraulichkeitsschutz der Attributinhalte: Zum Schutz der Privatsphäre des Subjekts SOLLTE die ausstellende Instanz (Quelle) die Attribute so ausstellen, dass nur die berechtigte Instanz die Attribute einsehen kann.

Falls die Kommunikation über Back-Channel läuft, MUSS der Kanal sicher und authentisiert sein.

In allen beteiligten Instanzen SOLLTEN empfohlene kryptographische Algorithmen und Schlüssellängen angewendet werden.[2]

Erfüllte Anforderung: 224-LB-4

Richtlinie 4 – Qualitätsmodell für die Authentifizierung: Für die Qualität der Authentifizierung der Subjekte MÜSSEN die Vertrauensstufen nach eCH-0170 [13] verwenden werden:

- *ech0170.vs1*: Vertrauensstufe 1 (Kein oder minimales Vertrauen),
- *ech0170.vs2*: Vertrauensstufe 2 (Geringes Vertrauen),
- *ech0170.vs3*: Vertrauensstufe 3 (Beträchtliches Vertrauen),
- *ech0170.vs4*: Vertrauensstufe 4 (Hohes Vertrauen).

[2] Für die Wahl von adäquaten Parametern wird auf diese Quellen verwiesen: www.keylength.org

Die RP MUSS die geforderte Vertrauensstufe bei der Registrierung in ihren Metadaten (siehe Kapitel 11.2) festlegen und KANN (falls sie unterschiedliche Vertrauensstufen benötigt) diese in der Authentifizierungsanfrage mitsenden.

Der Vermittler KANN in der Authentifizierungsanfrage die geforderte Vertrauensstufe mitsenden.

Der IdP MUSS die verfügbaren Vertrauensstufen in den Metadaten definieren. Falls mehrere Vertrauensstufen möglich sind, MUSS der IdP die verwendete Vertrauensstufe in der Authentifizierungsbestätigung mitsenden.

Der Vermittler MUSS in jedem Fall die Vertrauensstufe in der Authentifizierungsbestätigung mitsenden.

Erfüllte Anforderungen: 224-IAM-4, 224-IAM-4.1

Richtlinie 5 – Qualitätsmodell für die Attributbestätigung: Für die Qualität der Attributbestätigung der Subjekte MUSS das folgende oder ein gleichwertiges Vertrauensstufenmodell verwendet werden.

- *ech0224.aq1*: Attributqualität 1 (nicht bestätigte Attribute),
- *ech0224.aq2*: Attributqualität 2 (bestätigte Attribute),
- *ech0224.aq3*: Attributqualität 3 (staatlich bestätigte Attribute).

Die RP MUSS bei der Registrierung der Ressource beim Vermittler mitteilen, welche Attribute in welcher Qualität sie verlangt.

Der Vermittler MUSS in der Attributbestätigung die Qualität der Attribute an die RP mitsenden. Die Qualität der Attribute erhält der Vermittler entweder zur Laufzeit von der Attributbestätigung der AP oder aus den zur Definitionszeit definierten Metadaten der AP.

Falls der Vermittler Attribute bei der AP anfordert, KANN die AP bei der Attributbestätigung die Qualität der Attribute an den Vermittler mitsenden.

Falls eine RP Attribute bei der AP direkt anfordert (Distributed Claim), SOLLTE die AP bei der Attributbestätigung die Qualität der Attribute an die RP mitsenden.

Erfüllte Anforderung: 224-IAM-5

Richtlinie 6 – Überprüfen der Authentifizierungsanfrage: Die Authentifizierungsanfrage MUSS gemäss Kapitel 3.1.2.2 der OpenID Connect Spezifikation [3] von einem Empfänger überprüft werden. Falls die Validierung fehlschlägt, wird eine Fehlermeldung gemäss Kapitel 10.5 zurückgeschickt.

Erfüllte Anforderungen: 224-LB-5, 224-LB-8

Richtlinie 7 – Vorgaben an die Attributbestätigung: In der Attributbestätigung (UserInfo Response) MÜSSEN die Parameter *sub, iss* und *aud* identische Werte wie im zugehörigen ID Token haben.

Erfüllte Anforderung: 224-LE-3.1

Richtlinie 8 – Freigabe der Attribute (User Consent): Der Vermittler MUSS das Einverständnis des Subjekts einholen, dass die ausgewählten Attribute an die RP übermittelt werden dürfen.

Erfüllte Anforderung: 224-LB-6

5 Schnittstellen der Vermittlermodelle

In diesem Kapitel wird definiert, wie die drei Vermittlermodelle aus eCH-0224 [2] mit den entsprechenden Schnittstellenbeschreibungen umgesetzt werden. Es werden hier nur die Schnittstellen zwischen den Akteuren für die Laufzeitprozesse spezifiziert. Weitere Tätigkeiten der Akteure befinden sich im eCH Standard eCH-0224 [2] ab Kapitel 8.

Im Kapitel 5.1 befindet sich die Übersicht der Schnittstellen für die Authentifizierung und es wird auf die entsprechende Schnittstellendefinition der Akteure verwiesen.

Im Kapitel 5.2 befindet sich die Übersicht der Schnittstellen für die Authentifizierung mit Attributübergabe. Da die Vermittlermodelle sich bei der Attributübergabe unterscheiden, werden sie einzeln spezifiziert.

5.1 Authentifizierung

Bei einem Vermittlermodell erfolgt die reine Authentifizierung in beiden Protokoll-Abschnitten gleich. Abbildung 2 zeigt eine Übersicht der Authentifizierung.

Abbildung 2: Übersicht Schnittstellen Authentifizierung

Die RP sendet eine Authentifizierungsanfrage an den Vermittler (Kapitel 6.1.1). Der Vermittler MUSS die Anfrage validieren. Nachdem ein IdP gewählt wurde (Discovery) MUSS der Vermittler eine Authentifizierungsanfrage an den IdP senden (Kapitel 7.1.2). Das Subjekt MUSS vom IdP authentifiziert werden. Der IdP MUSS bei erfolgreicher Authentifizierung eine Authentifizierungsbestätigung an den Vermittler senden (Kapitel 8.2). Der Vermittler MUSS die Authentifizierungsbestätigung transformieren und diese an die RP senden (Kapitel 7.1.4).

5.2 Authentifizierung mit Attributübermittlung

Tabelle 7 stellt eine Übersicht dar, wie die Authentifizierung mit Attributübermittlung in den in eCH-0224 definierten Vermittlermodellen umgesetzt wird. In den darauffolgenden Kapiteln wird dies dann genauer beschrieben.

	Vermittlermodelle		
	Double-Blinding	Offene Quellen	Vertraulichkeitsschutz
RP	Erhält die Attribute vom Vermittler als Normal Claim. (Kapitel 6.2.2)	Sichtbar: Normal Claim mit Quelle als Attribut. (Kapitel 6.2.2)	Erhält vom Vermittler ein Distributed Claim. (Kapitel 6.2.4)
		Verifizierbar: Aggregated Claim signiert von AP (IdP/AP). (Kapitel 6.2.3)	Die Attribute werden direkt beim AP abgefragt. (Kapitel 6.2.4.2)

Vermittler	Übermittelt die Attribute als Normal Claim an die RP. (Kapitel 7.2.4.1)	Sichtbar: Vermittler macht die AP über ein Attribut sichtbar. (Kapitel 7.2.4.1)	Übermittelt die Attribute als Distributed Claim mit JWT Access Token
		Verifizierbar: Der Vermittler übernimmt das signierte Aggregated Claim vom AP und sendet dieses weiter an die RP. (Kapitel 6.2.3)	Access Token mit Introspection (Kapitel 7.2.4.3)
AP	Übermittelt die Attribute als Normal Claim an den Vermittler. (Kapitel 9.3.1)	Sichtbar: Übermittelt die Attribute als Normal Claim an den Vermittler. (Kapitel 9.3.1)	Sendet die Attribute direkt zur RP. (Kapitel 9.3.3)
		Verifizierbar: Aggregated Claim, bei welchem der 'sub' Parameter ausserhalb des Aggregated Claims ist. (Kapitel 9.3.2)	

Tabelle 7: Übersicht Authentifizierung mit Attributübermittlung

5.2.1 Vermittler Double-Blinding

In der Abbildung 3 ist eine Übersicht der Schnittstellen ersichtlich. Die detaillierte Definition des Vermittlermodells Double-Blinding befindet sich im eCH-0224 [2] im Kapitel 8.2.

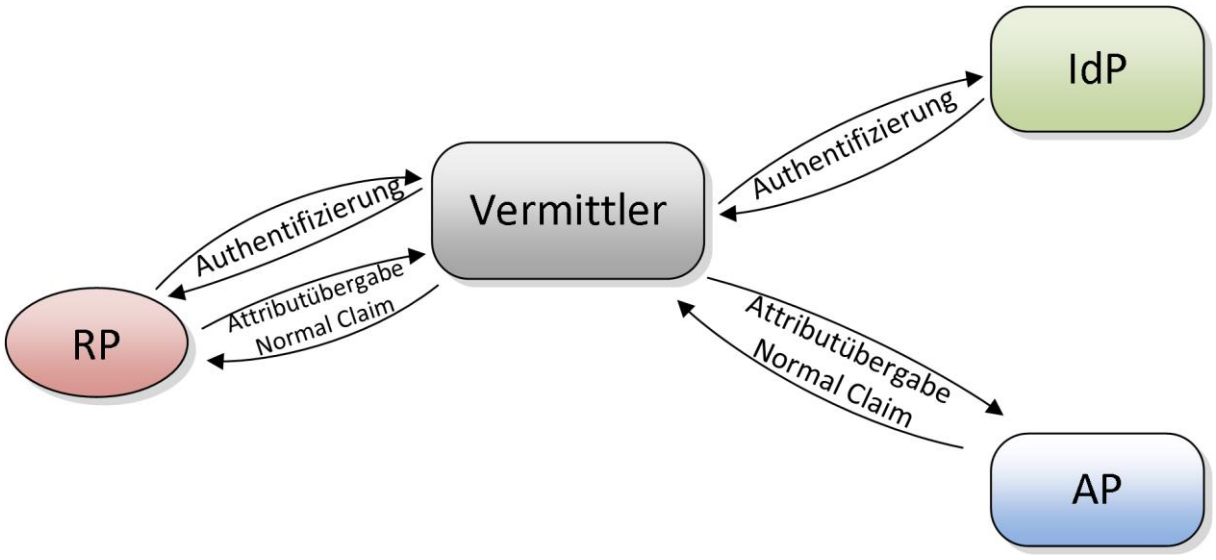

Abbildung 3: Übersicht Schnittstellen - Vermittlermodell Double-Blinding

Der Vermittler MUSS der RP die Attribute als Normal Claim übergeben (siehe Kapitel 3.3.1).

Die AP übergibt dem Vermittler die Attribute. Dies SOLLTE mit einem Normal Claim geschehen, wobei auch Aggregated Claim oder Distributed Claim möglich sind. Wichtig ist, dass der Vermittler die Authentifizierungs- und Attributbestätigungen so transformieren MUSS, sodass

die RP nicht auf den IdP bzw. AP rückschliessen kann.

5.2.2 Vermittler Offene Quellen

Abbildung 4 zeigt eine Übersicht der Schnittstellen, bei der die Attributquellen verifizierbar sind. Die detaillierte Definition des Vermittlermodells Offene Quellen befindet sich im eCH-0224 [2] im Kapitel 8.3.

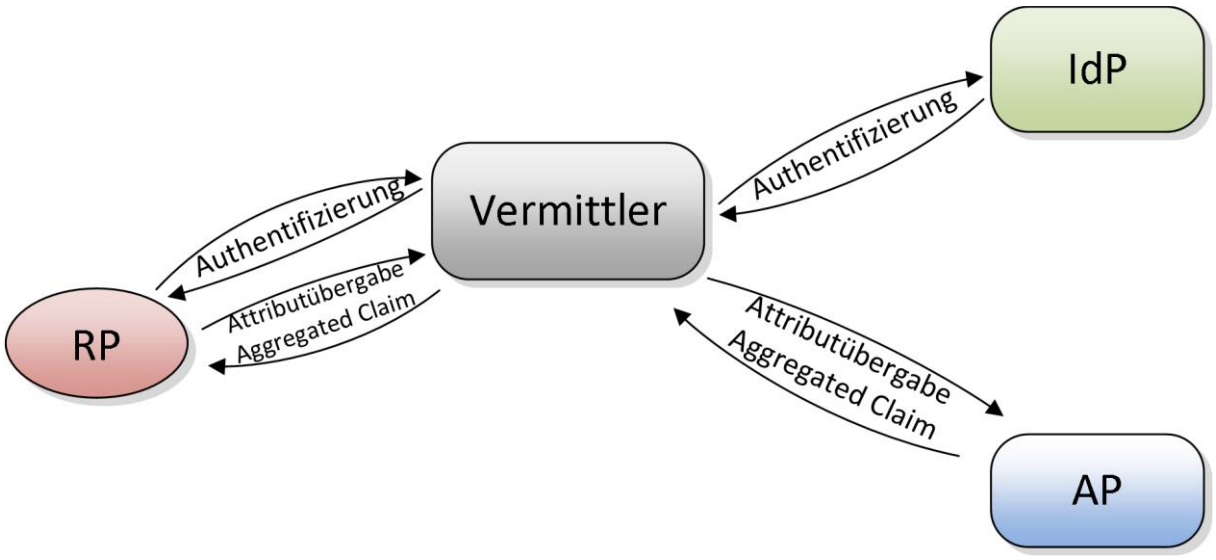

Abbildung 4: Übersicht Schnittstellen - Vermittlermodell Offene Quellen

Falls die Quellen nur sichtbar gemacht werden müssen, KANN der Vermittler analog zu Kapitel 5.2.1 verfahren, MUSS aber zusätzlich die Quelle mit einem weiteren Attribut in einem Normal Claim offenlegen.

Für die Verifizierbarkeit von Attributquellen für die RP MÜSSEN Aggregated Claims verwendet werden. Dabei MUSS der AP einen Aggregated Claim an den Vermittler senden. Der Vermittler MUSS den Aggregated Claim an die RP weiterleiten. Der AP DARF NICHT wissen, an wen die Attribute gesendet wurden. Umgekehrt KANN die RP die Quelle der Attribute verifizieren.

5.2.3 Vermittler Vertraulichkeitsschutz

In der Abbildung 5 ist eine Übersicht der Schnittstellen für einen Vermittler mit Vertraulichkeitsschutz ersichtlich. Die detaillierte Definition des Vermittlermodells Vertraulichkeitsschutz befindet sich im eCH-0224 [2] im Kapitel 8.4.

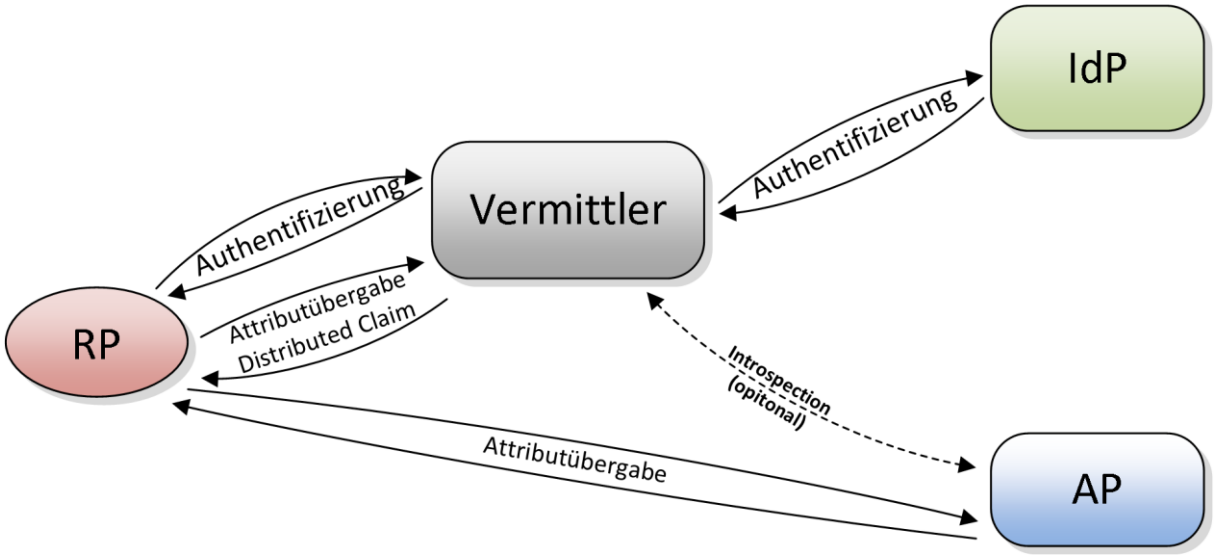

Abbildung 5: Übersicht Schnittstellen - Vermittlermodell Vertraulichkeitsschutz

Damit die Vertraulichkeit der Attribute gewahrt wird, MUSS der Vermittler ein Distributed Claim mit entsprechendem Access Token an die RP senden. Die RP KANN mit diesem die Attribute direkt beim AP abholen. Der Vermittler DARF KEINE Einsicht in die Attributwerte erhalten. Siehe Richtlinie 3 (Vertraulichkeitsschutz der Attributinhalte).

Der Vermittler MUSS der RP ein (oder mehrere) JWT Access Token (siehe 3.2.2.1) oder Access Token (siehe 3.2.2) im Distributed Claim übertragen.

Falls der Vermittler JWT Access Token verwendet, MUSS das JWT Access Token vom Vermittler signiert und für die AP verschlüsselt sein. So wird sichergestellt, dass die RP keine Einsicht in Informationen bezüglich Subjekt erhält. Falls das JWT Access Token verwendet wird, kann die AP nachvollziehen, für welches Subjekt welche Attribute übertragen werden sollen. Da das JWT Access Token signiert ist, KANN die AP es auf seine Authentizität überprüfen.

Falls der Vermittler Access Token im Distributed Claim verwendet, MUSS der Vermittler einen Introspection Endpoint für die APs anbieten. Falls die RP mit dem Access Token Attribute abfragt, MUSS der AP über den Introspection Endpoint weitere Informationen (Subjekt, Scope) abfragen, um der RP die entsprechenden Attribute zu liefern.

6 Schnittstellen für die Relying Party (RP)

In diesem Kapitel werden die Schnittstellen zur RP spezifiziert. Der Ablauf der Authentifizierung über eine Vermittlerinfrastruktur ist in Kapitel 6.1 und eine Authentifizierung mit Attributübergabe ist in Kapitel 6.2 definiert. Ein spezieller Anwendungsfall, in welchem Attribute aus mehreren Quellen (zusätzlichen APs) abgefragt werden, ist in Kapitel 6.2.4 definiert.

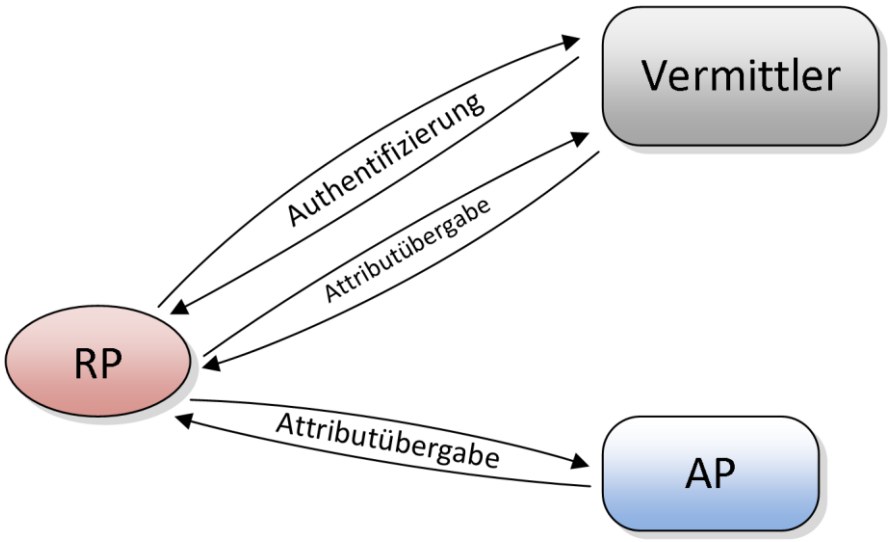

Abbildung 6: Übersicht Schnittstellen für RP

Die folgende Tabelle zeigt, in welchem Kapitel die entsprechende Anfrage (bzw. Antwort) der RP spezifiziert werden.

Authentifizierung	Authentifizierungsanfrage	Kapitel 6.1.1
	Authentifizierungsbestätigung	Kapitel 6.1.2
Authentifizierung mit Attributübergabe	Attributanfrage	Kapitel 6.2.1
	Attributbestätigung	Als Normal Claim (Attribute von Vermittler): - Kapitel 6.2.2 Als Aggregated Claim (Attribute von Vermittler): - Kapitel 6.2.3 Als Distributed Claim (Attributreferenz von Vermittler, Attribute von AP): - Kapitel 6.2.4

Tabelle 8: Übersicht Spezifikation Schnittstellen RP

6.1 Authentifizierung

Die Authentifizierung MUSS immer über den Vermittler laufen. Das Resultat einer erfolgreichen Authentifizierung MUSS ein ID Token (Kapitel 3.2.1) sein.

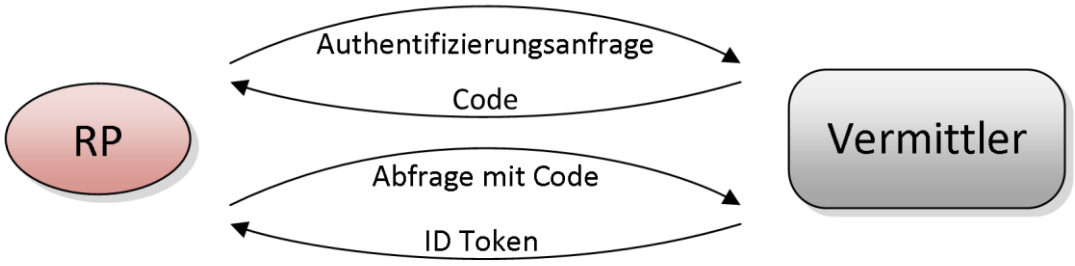

Abbildung 7: Schnittstellen der RP für Authentifizierung

6.1.1 Authentifizierungsanfrage an den Vermittler

Der Ablauf der Authentifizierungsanfrage ist gemäss Kapitel 3.1.2 der OpenID Connect Spezifikation [3] definiert. Die RP MUSS das Subjekt wird über die Client Plattform an den Vermittler weiterleiten. Weitere Informationen zur Authentifizierungsanfrage sind im Kapitel 10.1.1 ersichtlich.[3]

```
HTTP/1.1 302 Found
  Location: https://vermittler.example.com/authorize?
    response_type=code
    &scope=openid
    &client_id=rp_client_id
    &redirect_uri=https%3A%2F%2Foidc-rp.example.com%2Foidccallback
    &state=bjl4cV3m78K12
```

Listing 3: HTTP Redirect Authentication Request Response

Die Client Plattform MUSS aufgrund der Weiterleitung einen Authentication Request mittels HTTP-GET an den Vermittler senden.

```
GET /authorize?
    response_type=code
    &scope=openid
    &client_id=rp_client_id
    &state=bjl4cV3m78K12
    &redirect_uri=https%3A%2F%2Foidc-rp.example.com%2Foidccallback
HTTP/1.1
  Host: https://vermittler.example.com
```

Listing 4: HTTP-GET Authentication Request

Sobald das Subjekt erfolgreich authentifiziert wurde, MUSS der Vermittler einen Authorization Code über die Client Plattform des Subjekts an die RP senden. Richtlinie 1 (Authentifizierung der anfragenden Instanz) MUSS eingehalten werden.

[3] Der Parameter *acr_value* KANN in dieser Anfrage auch verwendet werden, falls die RP mehrere Vertrauensstufen unterstützt.

```
HTTP/1.1 302 Found
  Location: https://oidc-rp.example.com/oidccallback
    code=Bpl12OBeZQIAQYbYS6WxSb
    &state=bjl4cV3m78K12
```

Listing 5: Successful Authentication Response

Die RP MUSS die Antwort gemäss Kapitel 3.1.2.7 der OpenID Connect Spezifikation [3] überprüfen.

6.1.2 Authentifizierungsbestätigung vom Vermittler

Der Erhalt der Authentifizierungsbestätigung (ID Token) ist gemäss Kapitel 3.1.3 der OpenID Connect Spezifikation [3] definiert.

Die RP MUSS mittels erhaltenem Authorization Code das ID Token beim Token Endpoint des Vermittlers anfordern. Nachstehend der Token Request mit *private_key_jwt* als Authentisierungsmethode. Alternativ KANN *client_secret_basic* als Authentifizierungsmethode verwendet werden (siehe Kapitel 10.1.3).

```
POST /token HTTP/1.1
  Host: vermittler.example.com
  Content-Type: application/x-www-form-urlencoded

  grant_type=authorization_code
   &code=Bpl12OBeZQIAQYbYS6WxSb
   &client_id=rp_client_id
   &redirect_uri=https%3A%2F%2Foidc-rp.example.com%2Foidccallback
   &client_assertion_type= urn%3Aietf%3Aparams%3Aoauth%3Aclient-asser-
tion-type%3Ajwt-bearer
   &client_assertion=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ
ycF9jbGllbnRfaWQiLCJzdWIiOiJycF9jbGllbnRfaWQiLCJhdWQiOiJodHRwczovL3Zl
cm1pdHRsZXIuZXhhbXBsZS5jb20vdG9rZW4iLCJqdGkiOiI2NmViN2EyNGQzNTI2Y2EwY
zgzYTM4MTFhODE1YWQ5NzNkOWE0OTU5M2NiYTRmZGY4M2Q4ZjNkMDc5MDM2Mzk3IiwiZX
hwIjoxNTI1MjcwNjUyLCJpYXQiOjE1MjUyNzA1MDB9.sVjt03q20U_yS10ZPdndwHkc3F
QjBhUtVv_jGgYZjhQTqsiBGw3DuZXkOQrQaCOL1Cy_1RhzCVvlbFbBcpaChgZSBRieIKj
XcriSerhTfCliNpEyjiPw7A0O0KhnCWuECRFgpNSlKuIrsXssYoAu1VgkYVrALQNLePHZ
m17-4TG02LyZEPELU0LEvEMu2aiXV3zJwZOX7TcdC3gL3l2fahPKrDFPyo7YUJyrefr-M
D_N9zMCu5-zgphRLA2y3bUe0sww8aXSTwoXC7Ve2T1RmW1iSi0LFzy9AmVIxhqmpBYrQb
lxF-pKCB2aUE-weswpLHGX1lkcKyCCqq5R9Ms4Qg
```

Listing 6: HTTP-POST Token Request

In Listing 7 ist der Inhalt des signierten JWTs – aus dem Parameter *client_assertion* – von Listing 6 ersichtlich. Die Parameter des JWTs sind in Kapitel 9 der OpenID Connect Spezifikation [3] definiert.

```
{
  "iss": "rp_client_id",
  "sub": "rp_client_id",
  "aud": "https://vermittler.example.com/token",
  "jti":
"66eb7a24d3526ca0c83a3811a815ad973d9a49593cba4fdf83d8f3d079036397",
  "exp": 1525270652,
  "iat": 1525270500
}
```

Listing 7: Inhalt des private_key_jwts – Authentifizierungsbestätigung RP

Der Vermittler MUSS die Anfrage gemäss Kapitel 3.1.3.2 der OpenID Connect Spezifikation [3] überprüfen. Schlägt die Validierung fehl, MUSS eine Fehlermeldung gemäss Kapitel 10.5 retourniert werden.

Darauffolgend (in Listing 8) ist die Authentifizierungsbestätigung (Token Response) des Vermittlers. Um die OAuth 2.0 Spezifikation [7] einzuhalten, MUSS das Access Token ebenfalls Teil der Response sein. Das Access Token wird in diesem Anwendungsfall nicht benötigt und DARF daher KEINE Gültigkeit aufweisen.

Refresh Tokens (OpenID Connect Spezifikation [3] Kapitel 12) KÖNNEN verwendet werden. Die Verantwortung liegt dabei bei der Gemeinschaft (IAM-Führung).

```
HTTP/1.1 200 OK
  Content-Type: application/json
  Cache-Control: no-store
  Pragma: no-cache

{
  "access_token": "WE34ixoiEr",
  "token_type": "Bearer",
  "expires_in": 1,
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwc
zovL3Zlcm1pdHRsZXIuZXhhbXBsZS5jb20iLCJzdWIiOiI2MTAwMjQ4NDU5NzEiLCJhdW
QiOiJycF9jbGllbnRfaWQiLCJhY3IiOiJlY2gwMTcwLnZzMSIsImV4cCI6MTUyNTI3MDk
yMiwiaWF0IjoxNTI1MjcwNjIyfQ.xQbs2NTdvEDKp6cnGZezYCSPkGFPfZqbuHnWor6bi
cQHssiVgoLUpHRz5gvEML94K2_vbpmPF8j8DMGt8nqV8dtq3LHHvWdfYGKxxT5w0YWBBB
pDIe-UnR6tbByX_vQJ_c7RotAqx7BhslEIfN2gHEUzKD7VM8hg8wFkOxSzNxsdr25ukG2
kegJlnHZz_5QoHN9o5FhvVpC3flN9efUeQi8bpKDBDYlQuzeFG0ks6GzFz054ayAEJP_6
RGwxUZd_Xe744PNiBDS3znVPYb9ZlifQJXeBa0z_Kj-I2aJg7SUMhK4aKl1PHext70YNJ
qpfCPHySrqpanZQHWY_z0ME0A"
}
```

Listing 8: Successful Token Response

Die RP MUSS das ID Token gemäss Kapitel 3.1.3.7 der OpenID Connect Spezifikation [3] überprüfen.

Folgend der JWT Payload des ID Tokens aus Listing 8. Das ID Token wurde vom Vermittler für die RP ausgestellt.

```
{
  "iss": "https://vermittler.example.com",
  "sub": "610024845971",
  "aud": "rp_client_id",
  "acr": "ech0170.vs1",
  "exp": 1525270922,
  "iat": 1525270622
}
```

Listing 9: JWT Payload des ID Tokens

6.2 Authentifizierung mit Attributübergabe

Bevor Attribute an die RP übertragen werden, MUSS das Subjekt erfolgreich authentifiziert worden sein. Abbildung 8 zeigt eine Übersicht der Kommunikation zwischen Vermittler und RP. Die RP MUSS mit der Authentifizierungsbestätigung (ID Token) ein Access Token erhalten. Mithilfe des Access Tokens KANN die RP Attribute beim Vermittler abfragen.

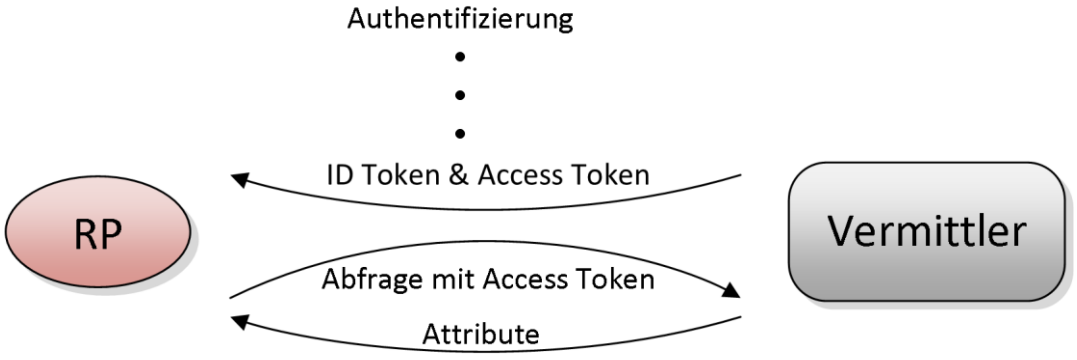

Abbildung 8: Schnittstellen der RP für Authentifizierung mit Attributübergabe

6.2.1 Attributanfrage an Vermittler

Im Unterschied zu Kapitel 6.1.1 MUSS die RP beim Authentication Request den *scope* mit den gewünschten Attributen anfordern.

```
GET /authorize?
    response_type=code
    &scope=openid%20profile%20email
    &client_id=rp_client_id
    &state=bjl4cV3m78K12
    &redirect_uri=https%3A%2F%2Foidc-rp.example.com%2Foidccallback
HTTP/1.1
  Host: https://vermittler.example.com
```

Listing 10: HTTP-GET Authentication Request

Das Subjekt MUSS analog zu Kapitel 6.1.1 über dessen Client Plattform an den Vermittler weitergeleitet werden.

Die RP MUSS mittels Token Request und Authorization Code, das ID Token und Access Token beim Token Endpoint des Vermittlers anfordern. Der Token Request ist identisch zu dem aus Kapitel 6.1.1.

Die Response vom Vermittler MUSS nebst ID Token – welches gemäss Richtlinie 2 (Authentizität und Integrität der Antwort) mindestens signiert ist – auch ein gültiges Access Token enthalten (siehe Kapitel 4). Die Gültigkeit des Access Token MUSS zeitlich beschränkt sein. Im Listing 11 ist die Gültigkeit des Access Tokens auf eine Stunde (3600 Sekunden) beschränkt.

```
[…]
  "access_token": "SlAV32hkKG",
  "token_type": "Bearer",
  "expires_in": 3600,
  "id_token": […]
```

Listing 11: Successful Token Response mit Access Token

Der Vermittler MUSS das ID Token gemäss Kapitel 3.1.3.7 und das Access Token gemäss Kapitel 3.1.3.8 der OpenID Connect Spezifikation [3] überprüfen.

Anschliessend KANN die RP mit dem Access Token die Attribute beim UserInfo-Endpoint des Vermittlers abfragen.

```
GET /userinfo HTTP/1.1
  Host: vermittler.example.com
  Authorization: Bearer SlAV32hkKG
```

Listing 12: Attributabfrage an Vermittler

Die Attributbestätigung kann je nach Vermittlermodell unterschiedlich aufgebaut sein (siehe Tabelle 7).

6.2.2 Attributbestätigung als Normal Claim

In einem Normal Claim MÜSSEN die Attribute im JWT Format vom Vermittler signiert an die RP gesendet werden (siehe Kapitel 3.1.2 der OpenID Connect Spezifikation [3]). Die Richtlinie 7 (Vorgaben an die Attributbestätigung) MUSS erfüllt sein.

Nachfolgend der JWT Inhalt mit den Attributen der UserInfo Response.

```
{
  "iss": "https://vermittler.example.com»,
  "aud": "rp_client_id",
  "sub": "610024845971",
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "preferred_username": "j.doe",
  "email": "janedoe@example.com"
}
```

Listing 13: Attributbestätigung (von Vermittler an RP)

6.2.3 Attributbestätigung als Aggregated Claim

In einem Aggregated Claim, MÜSSEN die Attribute im JWT Format vom Vermittler gemäss Kapitel 5.6.2 der OpenID Connect Spezifikation [3] an die RP gesendet werden. Die Richtlinie 7 (Vorgaben an die Attributbestätigung) MUSS erfüllt sein.

Nachfolgend ist ein mögliches Beispiel eines JWTs.


```
{
  "iss": "https://vermittler.example.com",
  "aud": "rp_client_id",
  "sub": "610024845971",
  "_claim_names": {
    "name": "idp-ap_example_src",
    "given_name": "idp-ap_example_src ",
    "family_name": "idp-ap_example_src ",
    "preferred_username": "idp-ap_example_src ",
    "email": "idp-ap_example_src "
  },
  "_claim_sources": {
    "idp-ap_example_src": {"JWT": "eyJ0eXAiOiJqd3QiLCJhbGciOiJSUzI1Ni
J9.eyJpc3MiOiJodHRwczovL29pZGMtaWRwLWFwLmV4YW1wbGUuY29tIiwibmFtZSI6Ik
phbmUgRG9lIiwiZ2l2ZW5fbmFtZSI6IkphbmUiLCJmYW1pbHlfbmFtZSI6IkRvZSIsInB
yZWZlcnJlZF91c2VybmFtZSI6ImouZG9lIiwiZW1haWwiOiJqYW5lZG9lQGV4YW1wbGUu
Y29tIn0.WJu0mc23fReDsZAc2VRHgSzcw5eGSTJydhnI9BGkyG8v79iy7qkLd9-94Bq43
mBv16zY2DlsfUkFDbiut5mq4WOVzPyFQf3-TCYJdItwPckPQc9gbA4RFfw8jN96j2vM2P
ur7J-_w7Di1Jr59Si96M1r4K_ETM7TDDjJH6Ppwq-bNYe1jG5eLl30rQlKn8363zdxAEy
8XJrYFBokE12sQDX_3SljexOsMiac18BJdo0xewvxquXlRYcLOlS3JP3R1sQJoDTnlC3x
FH1-JYqiSOxDnw_-yZknbTdVxkZ96ie5K8DK4FFCq8okUTqvkm0m7T1m1cWwEXcU5clhU
Cy_g"}
  }
}
```


Listing 14: JWT Payload der UserInfo Response mit Aggregated Claim

6.2.4 Attributbestätigung als Distributed Claim

Falls die Attribute bei einem weiteren AP abgefragt werden müssen, MUSS die RP anstatt eines Access Tokens eine Referenz in Form eines Distributed Claims vom Vermittler erhalten. Der Distributed Claim ist im Kapitel 5.6.2 der OpenID Connect Spezifikation [3] definiert. Mit Hilfe dieser Referenz KANN die RP die Attribute beim AP direkt abfragen.

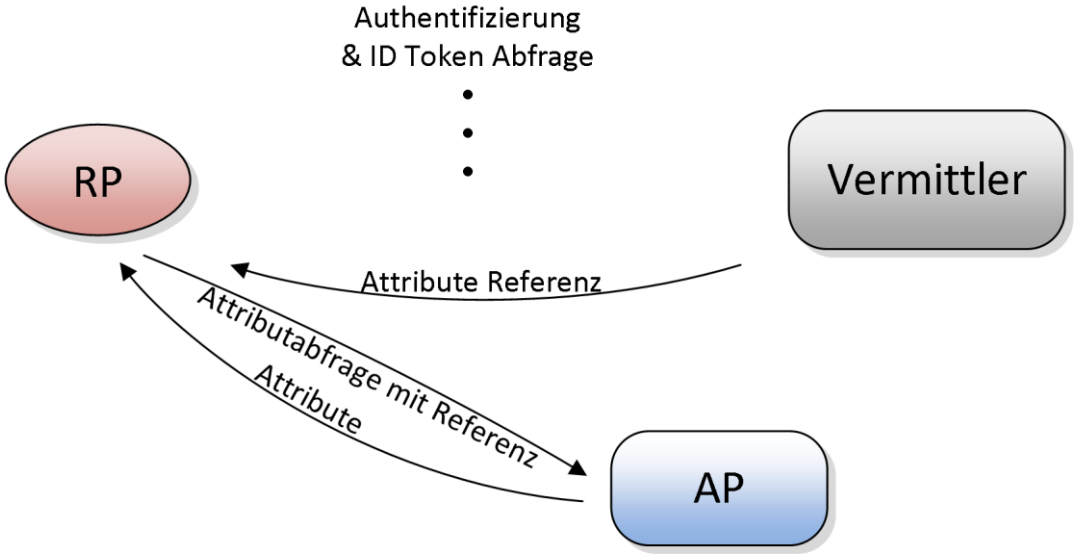

Abbildung 9: Schnittstellen RP - Attributbestätigung als Distributed Claim

Die Antwort der Attributabfrage beim Vermittler MUSS ein Distributed Claim sein, welcher als JWT vom Vermittler signiert an die RP versendet wird. Nachstehend der JWT Inhalt, der retournierten UserInfo Response, als JSON. Die Richtlinie 7 (Vorgaben an die Attributbestätigung) MUSS erfüllt sein.

```
{
  "iss": "https://vermittler.example.com»,
  "aud": "rp_client_id",
  "sub": "610024845971",
  "_claim_names": {
    "name": "ap1_src",
    "given_name": "ap1_src",
    "family_name": "ap1_src",
    "preferred_username": "ap1_src",
    "email": "ap1_src"
  },
  "_claim_sources": {
   " ap1_src": { "endpoint": "https://oidc-ap.example.com/attribut-
src",
      "access_token": "09a312B578Z654" }
  }
```

Listing 15: JWT Payload der UserInfo Response mit Distributed Claim

Die RP KANN anhand der erhaltenen URL (im *endpoint* von *ap1_src*) und des Access Tokens einen UserInfo Request direkt an den AP senden. Im Listing 15 wird als Beispiel eine zufällige Zeichenkette als Access Token verwendet. Es KANN auch ein verschlüsseltes und signiertes JWT verwendet werden (siehe Kapitel 3.2.2).

6.2.4.1 Attributanfrage an Attribut Provider (AP) mit Referenz

Für die Attributabfrage MUSS die in der Response vom Vermittler gelieferten Information verwendet werden.

```
GET /attribute-src HTTP/1.1
  Host: oidc-ap.example.com
  Authorization: Bearer 09a312B578Z654
```

Listing 16: Attributabfrage beim AP

6.2.4.2 Attributbestätigung von Attribut Provider (AP)

Die Attributbestätigung MUSS in Form eines JWTs vom AP geliefert werden. Das JWT KANN signiert sein. Falls das JWT signiert ist, MUSS die Signatur vom Empfänger überprüft werden, bevor die Attribute verwendet werden dürfen.

Nachfolgend sind die Attribute im JSON Format ersichtlich.

```
{
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "preferred_username": "j.doe",
  "email": "janedoe@example.com"
}
```

Listing 17: Attributwerte im JSON Format

Die Parameter *sub*, *iss* und *aud* sind explizit nicht vorhanden.

7 Schnittstellen für Vermittler

Der Vermittler ist der Protokoll-Endpunkt von RP, IdP und AP. Grundsätzlich sieht die RP den Vermittler als IdP/AP. IdP und AP sehen den Vermittler als RP.

Der Vermittler ist Bindeglied zwischen den Identitätslieferanten und -konsumenten. Die Authentifizierung MUSS immer über den Vermittler laufen. Je nach gewähltem Vermittlermodell werden die Attribute vom Vermittler oder von der AP an die RP übermittelt. In der Abbildung 10 ist eine Übersicht der Schnittstellen vom und zum Vermittler ersichtlich.

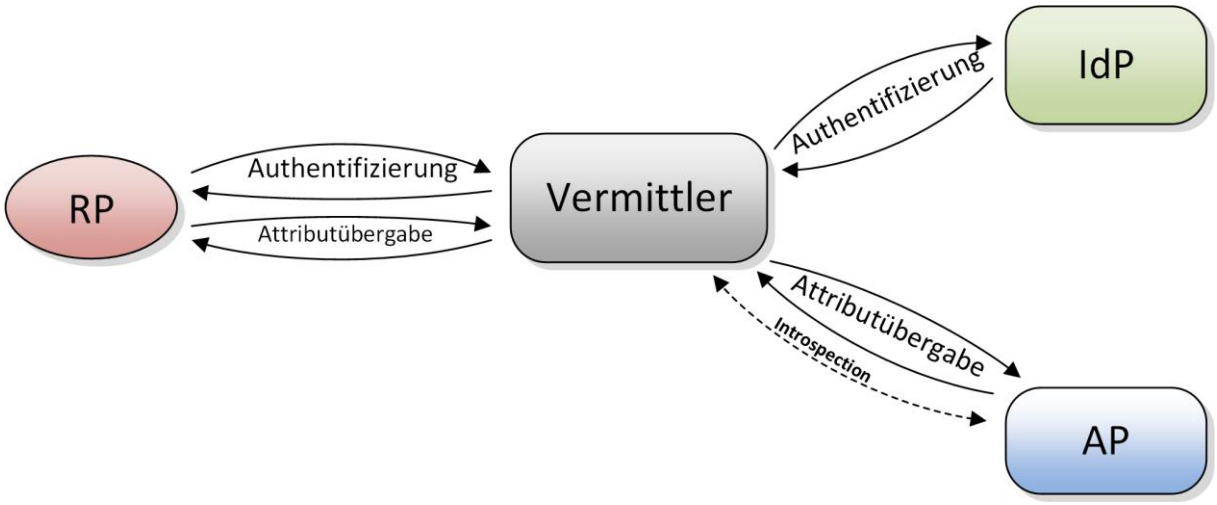

Abbildung 10: Schnittstellen Übersicht Vermittler

In den folgenden Kapiteln werden die Schnittstellen spezifiziert. Dabei werden pro Anwendungsfall die ausgetauschten Nachrichten aus Sicht des Vermittlers definiert.

Authentifizierung	Authentifizierungsanfrage von RP	Kapitel 7.1.1
	Authentifizierungsbestätigung an RP	Kapitel 7.1.4
	Authentifizierungsanfrage an IdP	Kapitel 7.1.2
	Authentifizierungsbestätigung von IdP	Kapitel 7.1.3
Authentifizierung mit Attributübermittlung	Attributanfrage von RP	Kapitel 7.2.1
	Attributbestätigung an RP	Als Normal Claim: - Kapitel 7.2.4.1 Als Aggregated Claim: - Kapitel 7.2.4.2 Als Distributed Claim (und Introspection): - Kapitel 7.2.4.3

	Attributanfrage von Vermittler	Attributanfrage an einen IdP/AP: - Kapitel 7.2.2.1 Attributanfragen an ein AP: - Kapitel 7.2.2.2
	Attributbestätigung an Vermittler	Als Normal Claim: - Kapitel 7.2.3.1 Als Aggregated Claim: - Kapitel 7.2.3.2

Tabelle 9: Übersicht Spezifikation Schnittstellen Vermittler

7.1 Authentifizierung

Der Vermittler erwartet eine Authentifizierungsanfrage von der RP (Kapitel 7.1.1). Nachdem ein IdP für die Authentifizierung gewählt wurde, MUSS der Vermittler eine Authentifizierungsanfrage an den IdP senden (Kapitel 7.1.2).

Nach erfolgreichen Authentifizierung MUSS der IdP einen Code an den Vermittler senden. Der Vermittler fragt die Authentifizierungsbestätigung (ID Token) mit diesem Code ab (Kapitel 7.1.3).

Anschliessend MUSS der Vermittler einen von ihm generierten Code an die RP senden. Die RP fragt danach die Authentifizierungsbestätigung mit diesem Code ab (Kapitel 7.1.4).

Abbildung 11: Authentifizierung aus Sicht vom Vermittler

7.1.1 Authentifizierungsanfrage der Relying Party (RP)

Der Vermittler erhält eine Authentifizierungsanfrage von der RP. Weitere Informationen zur Authentifizierungsanfrage sind im Kapitel 10.1.1.

```
HTTP/1.1 302 Found
  Location: https://vermittler.example.com/authorize?
    response_type=code
    &scope=openid
    &client_id=rp_client_id
    &redirect_uri=https%3A%2F%2Foidc-rp.example.com%2Foidccallback
    &state=bjl4cV3m78K12
```

Listing 18: HTTP Redirect Authentication Request Response

Der Vermittler MUSS die Authentifizierungsanfrage gemäss Richtlinie 6 (Überprüfen der Authentifizierungsanfrage) überprüfen.

7.1.2 Authentifizierungsanfrage an Identity Provider (IdP)

Der Vermittler sendet eine Authentifizierungsanfrage mittels HTTP-GET an den IdP. Weitere Informationen zur Authentifizierungsanfrage sind im Kapitel 10.1.1 ersichtlich.

```
HTTP/1.1 302 Found
  Location: https://oidc-rp.example.com/authorize?
    response_type=code
    &scope=openid
    &client_id=vermittler_client_id
    &state=af0ifjsldkj
    &redirect_uri=https%3A%2F%2Fvermittler.example.com%2Foidccallback
    &acr_values=ech0170.vs1
```

Listing 19: HTTP Redirect Authentication Request Response

Der Parameter *acr_values* KANN verwendet werden, um dem IdP mitzuteilen, mit welcher Qualität das Subjekt authentifiziert werden sollte.

Das Subjekt wird über die Client Plattform zum IdP weitergeleitet.

```
GET /authorize?
    response_type=code
    &scope=openid
    &client_id=vermittler_client_id
    &state=af0ifjsldkj
    &redirect_uri=https%3A%2F%2Fvermittler.example.com%2Foidccallback
    &acr_values=ech0170.vs1
HTTP/1.1
  Host: https://oidc-rp.example.com
```

Listing 20: HTTP-GET Authentication Request

Der IdP MUSS die Authentifizierungsanfrage gemäss Richtlinie 6 (Überprüfen der Authentifizierungsanfrage) überprüfen.

7.1.3 Authentifizierungsbestätigung vom Identity Provider (IdP)

Das Subjekt wird vom IdP authentifiziert. Bei erfolgreicher Authentifizierung MUSS ein Authorization Code über die Client Plattform des Subjekts an den Vermittler retourniert werden. Bei Abbruch des Login-Vorgangs von Seiten des Subjektes, MUSS eine Fehlermeldung gemäss Kapitel 10.5 anstelle des Authorization Codes zurückgegeben werden.

```
HTTP/1.1 302 Found
  Location: https://vermittler.example.com/oidccallback
    code=SplxlOBeZQQYbYS6WxSbIA
    &state=af0ifjsldkj
```

Listing 21: Successful Authentication Response

Der Vermittler MUSS die Antwort gemäss Kapitel 3.1.2.7 der OpenID Connect Spezifikation [3] überprüfen.

Der Vermittler fordert mit dem Authorization Code das ID Token beim Token Endpoint des IdPs an. Als Authentifizierungsmethode MUSS *private_key_jwt* vom Vermittler verwendet werden. Die Definition der Parameter ist in Kapitel 9 der OpenID Connect Spezifikation [3] zu

finden. Es ist an dieser Stelle auch Richtlinie 1 (Authentifizierung der anfragenden Instanz) einzuhalten.

```
POST /token HTTP/1.1
  Host: oidc-rp.example.com
  Content-Type: application/x-www-form-urlencoded

  grant_type=authorization_code
   &code=SplxlOBeZQQYbYS6WxSbIA
   &client_id=vermittler_client_id
   &redirect_uri=https%3A%2F%2Fvermittler.example.com%2Foidccallback
   &client_assertion_type= urn%3Aietf%3Aparams%3Aoauth%3Aclient-asser-
tion-type%3Ajwt-bearer
   &client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ
2ZXJtaXR0bGVyX2NsaWVudF9pZCIsInN1YiI6InZlcm1pdHRsZXJfY2xpZW50X2lkIiwi
YXVkIjoiaHR0cHM6Ly93d3cub2lkYy1pZHAuY2gvdG9rZW4iLCJqdGkiOiJiYmQ0ZmRmO
DNkOGYzZDA3OTAzNjM5NzY2ZWI3YTI0ZDM1MjNjYTBjODNhMzgxMmE4MTVhZDk3M2Q5YT
Q5NTkzIiwiZXhwIjoxNTI1MjcwNjUyLCJpYXQiOjE1MjUyNzA1MDB9.npAZ2Psmd4MIi9
CCpHAR-W5YJhmdE_wiYkEFCsY8ovnr2SzD1oIoQJAdTEP9iAux1aMyy3426guSE5lDB8O
ra4WpmhTO9E0vtjxS2VHiFz1pYQCwDKeWKxpyQ7KN7_Nu1AFA0uMW4mVILR-JWQy9lgta
qzN-2NTtZOq0bYcBEX4bjoa4TjcnuOHxRUwvom0Uo4GXBG99P_mzx2WXmndA6zOsoV0VF
FQwYWYZiQr-__ECNRL7dF-a3pR0Wojfrmm2CesPoH6zvqsHSSDCQLgb_WqKrOsePIDBGP
ln7xVYaV8_XflnNXlAFY8ttxnw0VZkdaxd5GEL16TCEpFR8PYQ3g
```

Listing 22: HTTP-POST Token Request

In Listing 23 ist der Inhalt des signierten JWTs – aus dem Parameter *client_assertion* – von Listing 22 ersichtlich. Die Parameter des JWTs sind in Kapitel 9 der OpenID Connect Spezifikation [3] definiert.

```
{
  "iss": "vermittler_client_id",
  "sub": "vermittler_client_id",
  "aud": "https://oidc-rp.example.com/token",
  "jti":
"bbd4fdf83d8f3d07903639766eb7a24d3523ca0c83a3812a815ad973d9a49593",
  "acr": "ech0170.vs1",
  "exp": 1525270652,
  "iat": 1525270500
}
```

Listing 23: JSON von der Authentifizierungsbestätigung

Der IdP MUSS die Anfrage gemäss Kapitel 3.1.3.2 der OpenID Connect Spezifikation [3] überprüfen. Schlägt die Validierung fehl, MUSS eine Fehlermeldung gemäss Kapitel 10.5 retourniert werden.

In Listing 24 ist die Token Response des IdPs ersichtlich. Um die OAuth 2.0 Spezifikation [7] einzuhalten, MUSS das Access Token ebenfalls Teil der Response sein. Das Access Token wird in diesem Anwendungsfall nicht benötigt und weisst daher keine Gültigkeit auf.

```
HTTP/1.1 200 OK
  Content-Type: application/json
  Cache-Control: no-store
  Pragma: no-cache

{
  "access_token": "SlAV32hkKG",
  "token_type": "Bearer",
  "expires_in": 1,
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwc
zovL29pZGMtaWRwLmV4YW1wbGUuY29tIiwic3ViIjoiMjQ4Mjg5NzYxMDAxIiwiYXVkIj
oidmVybWl0dGxlcl9jbGllbnRfaWQiLCJhY3IiOiJlY2gwMTcwLnZzMSIsImV4cCI6MTU
yNTI3MDkyMCwiaWF0IjoxNTI1MjcwNjIwfQ.VTwfvkFzxRCfc8iuU9cWgH_13UD2jxjmT
sNtmFA_MO4pSqxdnXbIWCeobhcyAXyxylzNXN7kiVqc2Qg84x0f5r8o374b6e_oaSysf2
t30P3pIVl_2Vg8MG2oMne2SltWCv0Oo9q-zJ8_6Zg8Km_OjRuIFp9qynAjWNKFNuSSMHX
6pQvov66TnN3lstF17LdUnAuMnxzQ5ns3c1AaQBW4Fk9h9FmK9PT6eiBc49r8oo1_exx4
vsCnLPQxSLOalPbYY1iVh8TFlbd2JQJxUj3csYQZRAQOqLppt30Cr2JM6psAD3o6VRK1Z
hcTU-ksTkCQA9U6JtNCDK2p0-fk_Ui7bw"
}
```

Listing 24: Successful Token Response

Der Vermittler MUSS das ID Token gemäss Kapitel 3.1.3.7 OpenID Connect Spezifikation [3] prüfen.

Im nachfolgenden Listing 25 ist der Inhalt des ID Tokens aus Listing 24.

```
{
  "iss": "https://oidc-idp.example.com",
  "sub": "248289761001",
  "aud": "vermittler_client_id",
  "acr": "ech0170.vs1",
  "exp": 1525270920,
  "iat": 1525270620
}
```

Listing 25: JWT Payload des ID Tokens

7.1.4 Authentifizierungsbestätigung an Relying Party (RP)

Im Listing 26 ist der Inhalt einer Authentifizierungsbestätigung angegeben. Um das Double-Blinding zu wahren, MÜSSEN die Parameter *iss*, *sub* und *aud* gegenüber der Authentifizierungsbestätigung (Listing 25) angepasst werden, bevor diese an die RP versendet werden.

```
{
  "iss": "https://vermittler.example.com",
  "sub": "610024845971",
  "aud": "rp_client_id",
  "acr": "ech0170.vs1",
  "exp": 1525270922,
  "iat": 1525270622
}
```

Listing 26: ID Token vom Vermittler an die RP

Die RP erhält einen Code. Der Parameter *state* MUSS derselbe sein wie in Listing 18.

```
HTTP/1.1 302 Found
  Location: https://oidc-rp.example.com/oidccallback
    code=Bpl12OBeZQIAQYbYS6WxSb
    &state=bjl4cV3m78K12
```

Listing 27: Successful Authentication Response

Anschliessend fragt die RP mit Hilfe des Codes das ID Token ab.

```
POST /token HTTP/1.1
  Host: vermittler.example.com
  Content-Type: application/x-www-form-urlencoded

  grant_type=authorization_code
   &code=Bpl12OBeZQIAQYbYS6WxSb
   &client_id=rp_client_id
   &redirect_uri=https%3A%2F%2Foidc-rp.example.com%2Foidccallback
   &client_assertion_type= urn%3Aietf%3Aparams%3Aoauth%3Aclient-as-
sertion-type%3Ajwt-bearer
   &client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ
ycF9jbGllbnRfaWQiLCJzdWIiOiJycF9jbGllbnRfaWQiLCJhdWQiOiJodHRwczovL3Zl
cm1pdHRsZXIuZXhhbXBsZS5jb20vdG9rZW4iLCJqdGkiOiI2NmViN2EyNGQzNTI2Y2EwY
zgzYTM4MTFhODE1YWQ5NzNkOWE0OTU5M2NiYTRmZGY4M2Q4ZjNkMDc5MDM2Mzk3IiwiZX
hwIjoxNTI1MjcwNjAwLCJpYXQiOjE1MjUyNzA2MDB9.vSS8-spNJfloz55PFZr2KMq7q2
tGxqemJW3bA6B1ipkVShi2fw-0YjJd6nDBv62iUUKVqEL8f2R_buK8hm2Ni7fWXQ4YdZZ
_0YOUzdyVUyCguKHPAw-4TjE0ikz7UxKahpqf3lJkHRS8rqqjpqRtJOe4q_9Tiw9NHTnL
tXKESnzOIMgukPv1KUwkl_UI82rjuDoV0gUVHpYvJthcmhsEw4k9Q7LaOKV7v1pHjwp4B
bgko4VhzJkW7IReZPzx3fZ4hQExlabTu2yWuNWls477uEWqY7A5A4QW2JQgaVV8Px4GAF
G4VhWDl4CdKmejzN0Qz6yRk7ofUGD8tfXYLE-q3g
```

Listing 28: Token Anfrage RP an Vermittler

Der Vermittler MUSS die Anfrage gemäss Kapitel 3.1.3.2 der OpenID Connect Spezifikation [3] überprüfen. Schlägt die Validierung fehl, MUSS eine Fehlermeldung gemäss Kapitel 10.5 retourniert werden.

7.2 Authentifizierung mit Attributübergabe

Der Vermittler erwartet eine Attributanfrage von der RP (Kapitel 7.2.1). Anschliessend fragt der Vermittler beim entsprechenden AP (oder IdP/AP) die Attribute ab (Kapitel 7.2.2).

Der AP sendet eine Attributbestätigung an den Vermittler (Kapitel 7.2.3). Der Vermittler transformiert die Attributbestätigung und sendet diese an die RP (Kapitel 7.2.4).

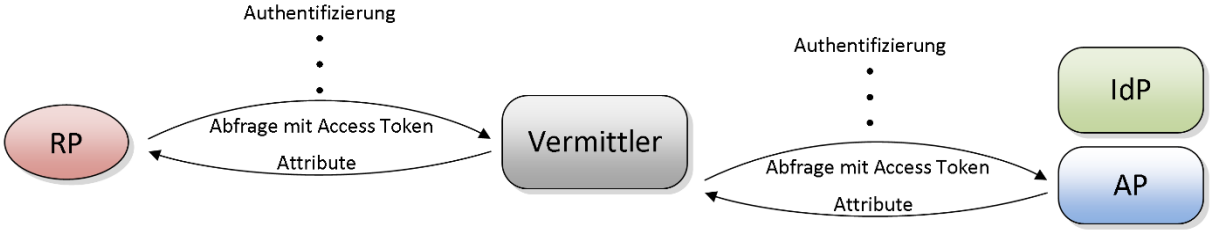

Abbildung 12: Attributübergabe aus Sicht des Vermittlers

7.2.1 Attributanfrage der Relying Party (RP)

Mit Hilfe des Parameters *scope* bei der Authentifizierungsanfrage MUSS angegeben werden, welche Attribute die RP verlangt.

```
[…] &scope=openid%20profile%20email […]
```

Listing 29: Scope Parameter der Authentifizierungsanfragen der RP

Die Response vom IdP/AP MUSS nebst einem ID Token – welches gemäss Richtlinie 2 (Authentizität und Integrität der Antwort) mindestens signiert ist – auch ein gültiges Access Token (siehe Kapitel 3.2.2) enthalten. Die Gültigkeit des Access Token MUSS zeitlich beschränkt sein. Im Listing 30 ist die Gültigkeit des Access Tokens auf eine Stunde (3600 Sekunden) beschränkt.

```
[…]
  "access_token": "SlAV32hkKG",
  "token_type": "Bearer",
  "expires_in": 3600,
  "id_token": […]
```

Listing 30: Authentifizierungsbestätigung mit Access Token

Der Vermittler MUSS das ID Token gemäss Kapitel 3.1.3.7 und das Access Token gemäss Kapitel 3.1.3.8 der OpenID Connect Spezifikation [3] prüfen.

Anschliessend KANN die RP mit dem Access Token die Attribute beim UserInfo-Endpoint des Vermittlers abfragen.

```
GET /userinfo HTTP/1.1
  Host: oidc-idp-ap.example.com
  Authorization: Bearer SlAV32hkKG
```

Listing 31: Attributabfrage an Vermittler

7.2.2 Attributanfragen des Vermittlers

Dieses Kapitel spezifiziert die Attributanfrage an einen IdP/AP (Kapitel 7.2.2.1) und AP (Kapitel 7.2.2.2).

7.2.2.1 Attributanfragen an IdP/AP

Während der Authentifizierungsanfrage MÜSSEN mit dem Parameter *scope* die angeforderten Attributsets angegeben werden.

```
[…] &scope=openid%20profile%20email […]
```

Die Response vom IdP/AP MUSS nebst einem ID Token – welches gemäss Richtlinie 2 (Authentizität und Integrität der Antwort) mindestens signiert ist – auch ein gültiges Access Token (siehe Kapitel 3.2.2) enthalten. Die Gültigkeit des Access Token MUSS zeitlich beschränkt sein. In Listing 32 ist die Gültigkeit des Access Tokens auf eine Stunde (3600 Sekunden) beschränkt.

```
[…]
  "access_token": "waG72XqiLqzV",
  "token_type": "Bearer",
  "expires_in": 3600,
  "id_token": […]
```

Listing 32: Authentifizierungsbestätigung mit Access Token

Der Vermittler MUSS das ID Token gemäss Kapitel 3.1.3.7 und das Access Token gemäss Kapitel 3.1.3.8 der OpenID Connect Spezifikation [3] prüfen.

```
GET /userinfo HTTP/1.1
  Host: oidc-idp-ap.example.com
  Authorization: Bearer waG72XqiLqzV
```

Listing 33: Attributabfrage an einen IdP/AP

7.2.2.2 Attributanfragen an einen AP

Aufgrund des vom IdP erhaltenen ID Tokens, stellt der Vermittler ein Access Token für den Attribut Provider – der die Attribute des Subjektes besitzt – aus. Für die Abfrage MUSS ein JWT Access Token (siehe Kapitel 3.2.2.1) verwendet werden. Die Parameter *sub* und *scope* MÜSSEN vorhanden sein, damit der AP das entsprechende Attributset zum retournieren erkennt. Falls ein Identity Mapping – auf dem Vermittler – vorhanden ist, MUSS der Vermittler den *sub* Parameter entsprechend anpassen.

```
{
  "aud": "ap_client_id",
  "iss": "https://vermittler.example.com",
  "exp": 1525270622,
  "iat": 1525270502,
  "sub": "248289761001",
  "scope": "attribute"
}
```

Listing 34: JWT Access Token für Attributanfrage bei einem AP

Der Vermittler MUSS das signierte JWT Access Token im HTTP Basic Authentication Header Parameter des UserInfo Requests an den UserInfo Endpoint des APs senden.

```
GET /userinfo HTTP/1.1
  Host: oidc-ap.example.com
  Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQi
OiJhcF9jbGllbnRfaWQiLCJpc3MiOiJodHRwczovL3Zlcm1pdHRsZXIuZXhhbXBsZS5jb
20iLCJleHAiOjE1MjUyNzA2MjIsImlhdCI6MTUyNTI3MDUwMiwic3ViIjoiMjQ4Mjg5Nz
YxMDAxIiwic2NvcGUiOiJhdHRyaWJ1dGUifQ.PUctS7SPf4UeEqgjsorYl6zBw4JL2Fiw
w9CvWoOp24U1l7H-4DuVuZBjY3rLHTrQV7kfxbTI4q2LMuEjQofLbzGoUcnNVKBz_6mxE
j1ThWHV3RT0NorTI5Is9Wjv0kS2ZYBgMUdxp2Vl-zINy0-TN2nRbzYGDW1vldGNYauRR1
898Z-9S8H9xOISt8W4PuTpJ8gHChmZPefF7xn2diTgv9ymE4O03bUVx-OhVO1U8PhpuMP
P2hQNVujEO4xf5L589-rfX0wOYkKUlgcMfSLoH-EiR8KlLvFPznOXydyUwNhyKILHOC6Q
ZNFgUBtiSFiSr3anV8sLcR6HTTkv3mUHTQ
```

Listing 35: HTTP-GET UserInfo Request)

7.2.3 Attributbestätigungen an den Vermittler

Die zu retournierenden Attribute MÜSSEN gemäss Richtlinie 2 (Authentizität und Integrität der Antwort) signiert zurückgesandt werden.

7.2.3.1 Normal Claim

Die Attributbestätigung als Normal Claim ist die einfachste Form einer Attributbestätigung.

Nachfolgend (Listing 36) ist der Inhalt eines JWTs mit den Attributen von einem IdP/AP. Die Richtlinie 7 (Vorgaben an die Attributbestätigung) MUSS erfüllt sein.

```
{
  "iss": "https://oidc-idp-ap.example.com",
  "aud": "vermittler_client_id",
  "sub": "248289761001",
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "preferred_username": "j.doe",
  "email": "janedoe@example.com"
}
```

Listing 36: Normal Claim von einem IdP/AP

In Listing 37 ist ein Beispiel, wie ein AP die Attribute an den Vermittler retourniert.

```
{
  "iss": "https://oidc-ap.example.com",
  "aud": "vermittler_client_id",
  "function": "tester",
  "canton": "bern"
}
```

Listing 37: Normal Claim von einem AP

7.2.3.2 Aggregated Claim

Nachfolgend dargestellt (Listing 38) ist die vom IdP/AP signierte und als Aggregated Claim retournierte Response. Die Richtlinie 7 (Vorgaben an die Attributbestätigung) MUSS erfüllt sein.

```
{
  "iss": "https://oidc-idp-ap.example.com",
  "aud": "vermittler_client_id",
  "sub": "248289761001",
  "_claim_names": {
    "name": "src1",
    "given_name": "src1",
    "family_name": "src1",
    "preferred_username": "src1",
    "email": "src1"
  },
  "_claim_sources": {
    "src1": {"JWT": "eyJ0eXAiOiJqd3QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOi
JodHRwczovL29pZGMtaWRwLWFwLmV4YW1wbGUuY29tIiwibmFtZSI6IkphbmUgRG9lIi
wiZ2l2ZW5fbmFtZSI6IkphbmUiLCJmYW1pbHlfbmFtZSI6IkRvZSIsInByZWZlcnJlZF
91c2VybmFtZSI6ImouZG9lIiwiZW1haWwiOiJqYW5lZG9lQGV4YW1wbGUuY29tIn0.WJ
u0mc23fReDsZAc2VRHgSzcw5eGSTJydhnI9BGkyG8v79iy7qkLd9-94Bq43mBv16zY2D
lsfUkFDbiut5mq4WOVzPyFQf3-TCYJdItwPckPQc9gbA4RFfw8jN96j2vM2Pur7J-_w7
Di1Jr59Si96M1r4K_ETM7TDDjJH6Ppwq-bNYe1jG5eLl30rQlKn8363zdxAEy8XJrYFB
okE12sQDX_3SljexOsMiac18BJdo0xewvxquXlRYcLOlS3JP3R1sQJoDTnlC3xFH1-JY
qiSOxDnw_-yZknbTdVxkZ96ie5K8DK4FFCq8okUTqvkm0m7T1m1cWwEXcU5clhBUCy_g"
    }
  }
}
```


Listing 38: Aggregated Claim von IdP/AP

7.2.4 Attributbestätigungen an RP

Es gibt verschiedene Möglichkeiten, wie die Attribute vom Vermittler an die RP übergeben werden können. Welche Art verwendet wird, MUSS von der Gemeinschaft evaluiert und entschieden werden.

Bevor der Vermittler die Attributbestätigung an die RP übermittelt, MUSS das Subjekt die Attribute freigeben. Siehe Richtlinie 8 - Freigabe der Attribute (User Consent).

Die einfachste Art ist die Übergabe mittels **Normal Claim**. Dabei werden die Attribute in einem JWT übergeben.

Aggregated Claim oder **Distributed Claim** können verwendet werden, um die Quellen offenzulegen oder Attribute zu übergeben, die nur für die RP ersichtlich sein sollen.

Die Richtlinie 7 (Vorgaben an die Attributbestätigung) MUSS in allen Fällen erfüllt sein.

7.2.4.1 Übergabe der Attributbestätigung als Normal Claim

Nachfolgend (Listing 39) ist ein Beispiel aufgeführt, wie die Attribute als Normal Claim an die RP übergeben werden können.

```
{
  "iss": "https://vermittler.example.com»,
  "aud": "rp_client_id",
  "sub": "610024845971",
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "preferred_username": "j.doe",
  "email": "janedoe@example.com"
}
```

Listing 39: Attribute als Normal Claim für RP

7.2.4.2 Übergabe der Attributbestätigung als Aggregated Claim

Nachfolgend (Listing 40) ist ein Beispiel aufgeführt, wie die Attribute als Aggregated Claim an die RP übergeben werden können.

```
{
  "iss": "https://vermittler.example.com»,
  "aud": "rp_client_id",
  "sub": "610024845971",
  "_claim_names": {
    "name": "src1",
    "given_name": "src1",
    "family_name": "src1",
    "preferred_username": "src1",
    "email": "src1"
  },
  "_claim_sources": {
    "src1": {"JWT": "eyJ0eXAiOiJqd3QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ
odHRwczovL29pZGMtaWRwLWFwLmV4YW1wbGUuY29tIiwibmFtZSI6IkphbmUgRG9lIiwi
Z2l2ZW5fbmFtZSI6IkphbmUiLCJmYW1pbHlfbmFtZSI6IkRvZSIsInByZWZlcnJlZF91c
2VybmFtZSI6ImouZG9lIiwiZW1haWwiOiJqYW5lZG9lQGV4YW1wbGUuY29tIn0.WJu0mc
23fReDsZAc2VRHgSzcw5eGSTJydhnI9BGkyG8v79iy7qkLd9-94Bq43mBv16zY2DlsfUk
FDbiut5mq4WOVzPyFQf3-TCYJdItwPckPQc9gbA4RFfw8jN96j2vM2Pur7J-_w7Di1Jr5
9Si96M1r4K_ETM7TDDjJH6Ppwq-bNYe1jG5eLl30rQlKn8363zdxAEy8XJrYFBokE12sQ
DX_3SljexOsMiac18BJdo0xewvxquXlRYcLOlS3JP3R1sQJoDTnlC3xFH1-JYqiSOxDnw
_-yZknbTdVxkZ96ie5K8DK4FFCq8okUTqvkm0m7T1m1cWwEXcU5clhBUCy_g"
    }
  }
}
```

Listing 40: Attribute als Aggregated Claim für RP

7.2.4.3 Übergabe der Attributbestätigung als Distributed Claim

Die Abbildung 13 zeigt eine Übersicht der Attributbestätigung als Distributed Claim vom Vermittler.

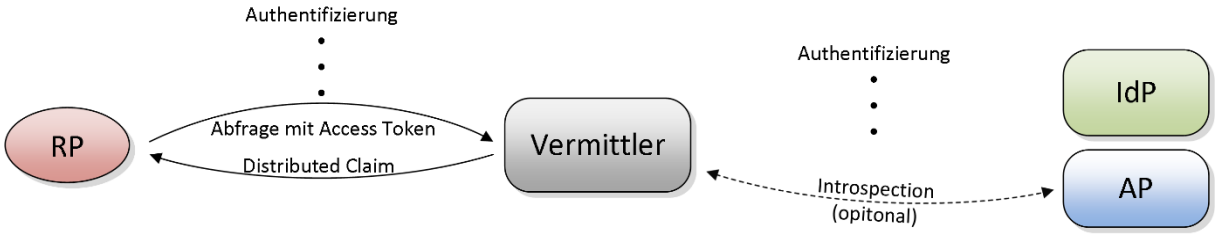

Abbildung 13: Attribute als Distributed Claim vom Vermittler

In Listing 41 ist ein Beispiel aufgeführt, wie die Attribute als Distributed Claim an die RP übergeben werden können.

Dabei MUSS entweder ein normales Access Token oder ein JWT Access Token für die Abfrage beim AP verwendet werden.

Falls ein normales Access Token verwendet wird, MUSS der Vermittler einen Introspection Endpoint zur Verfügung stellen, damit der AP abfragen kann, von welchem Subjekt die Attribute an die RP übertragen werden sollen (siehe Kapitel 7.2.4.3.1).

Bei einem JWT Access Token sind die entsprechenden Informationen für die AP bereits enthalten. Das JWT MUSS vom Vermittler signiert und für die AP verschlüsselt sein.

```
{
  "iss": "https://vermittler.example.com»,
  "aud": "rp_client_id",
  "sub": "248289761001",
  "_claim_names": {
    "preferred_username": "src1",
    "email": "src1"
  },
  "_claim_sources": {
   "src1": {"endpoint": "https://oidc-rp.example.com/userinfo",
      "access_token": "cSZmCoB…eyJ0eXA" }
  }
}
```

Listing 41: Attribute als Distributed Claim für RP

7.2.4.3.1 Introspection Protokoll

Ein Vermittler KANN einen Introspection Endpoint anbieten, bei der die APs weitere Informationen zu einem Access Token abfragen können.

```
POST /introspect HTTP/1.1
  Host: vermittler.example.com
  Content-Type: application/x-www-form-urlencoded
  Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
  token=09a312B578Z654&token_type_hint=access_token
```

Listing 42: Introspection Anfrage von AP an Vermittler

Der Vermittler überprüft anhand der Basic Authentication, ob der AP berechtigt ist, weitere Informationen zum Access Token zu erhalten. Danach MUSS der Vermittler überprüfen, ob das Access Token noch gültig ist. Bei erfolgreicher Überprüfung MUSS der Vermittler ein JSON Objekt senden. Folgend (Listing 43) ist ein Beispiel dargestellt, wie das JSON aufgebaut sein könnte.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "active": true,
  "iss": "https://vermittler.example.com»,
  "aud": "https://oidc-ap.example.com/attribute-src",
  "scope": "canton function",
  "sub": "248289761001",
  "exp": 1524057621,
  "iat": 1524057591
}
```

Listing 43: Introspection Antwort von Vermittler an AP

Falls die Überprüfung fehlschlägt, MUSS der Vermittler gemäss Kapitel 10.5 eine Fehlermeldung senden.

8 Schnittstellen für Identity Provider (IdP)

In diesem Kapitel werden die Schnittstellen für den IdP spezifiziert. Der IdP wird nur für die Authentifizierung verwendet. Daher werden nur Authentifizierungsanfrage und -bestätigung (ID Token) definiert. Die Abbildung 14 zeigt eine Übersicht.

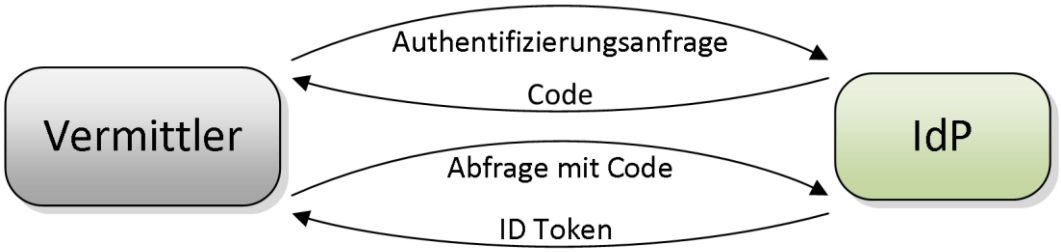

Abbildung 14: Schnittstellen vom IdP

Authentifizierung	Authentifizierungsanfrage von Vermittler	Kapitel 8.1
	Authentifizierungsbestätigung an Vermittler	Kapitel 8.2

Tabelle 10: Übersicht Spezifikation Schnittstellen IdP

8.1 Authentifizierungsanfrage von Vermittler

Der Vermittler sendet eine Authentifizierungsanfrage mittels HTTP-GET an den IdP. Weitere Informationen zur Authentifizierungsanfrage sind im Kapitel 10.1.1 ersichtlich.

```
HTTP/1.1 302 Found
  Location: https://oidc-rp.example.com/authorize?
    response_type=code
    &scope=openid
    &client_id=vermittler_client_id
    &state=af0ifjsldkj
    &redirect_uri=https%3A%2F%2Fvermittler.example.com%2Foidccallback
    &acr_values=ech0170.vs1
```

Listing 44: Authentifizierungsanfrage von Vermittler an IdP (Redirect)

Der Parameter *acr_values* KANN verwendet werden, um dem IdP mitzuteilen, mit welcher Qualität das Subjekt authentifiziert werden soll.

Das Subjekt wird über die Client Plattform zum IdP weitergeleitet.

```
GET /authorize?
    response_type=code
    &scope=openid
    &client_id=vermittler_client_id
    &state=af0ifjsldkj
    &redirect_uri=https%3A%2F%2Fvermittler.example.com%2Foidccallback
    &acr_values= ech0170.vs1
HTTP/1.1
  Host: https://oidc-rp.example.com
```

Listing 45: Authentifizierungsanfrage von Vermittler an IdP (HTTP-GET)

Der IdP MUSS die Authentifizierungsanfrage gemäss Richtlinie 6 (Überprüfen der Authentisierungsanfrage) überprüfen.

8.2 Authentifizierungsbestätigung an Vermittler

Der Vermittler fordert mit dem Authorization Code das ID Token beim Token Endpoint des IdPs an. Als Authentifizierungsmethode MUSS *private_key_jwt* vom Vermittler verwendet werden. Die Definition der Parameter ist in Kapitel 9 der OpenID Connect Spezifikation [3] zu finden. Siehe Richtlinie 1 (Authentifizierung der anfragenden Instanz).

```
POST /token HTTP/1.1
  Host: oidc-rp.example.com
  Content-Type: application/x-www-form-urlencoded

  grant_type=authorization_code
   &code=SplxlOBeZQQYbYS6WxSbIA
   &client_id=vermittler_client_id
   &redirect_uri=https%3A%2F%2Fvermittler.example.com%2Foidccallback
   &client_assertion_type= urn%3Aietf%3Aparams%3Aoauth%3Aclient-asser-
tion-type%3Ajwt-bearer
   &client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ
2ZXJtaXR0bGVyX2NsaWVudF9pZCIsInN1YiI6InZlcm1pdHRsZXJfY2xpZW50X2lkIiwi
YXVkIjoiaHR0cHM6Ly9vaWRjLWlkcC52ZXJtaXR0bGVyLmNvbS90b2tlbiIsImp0aSI6I
mJiZDRmZGY4M2Q4ZjNkMDc5MDM2Mzk3NjZlYjdhMjRkMzUyM2NhMGM4M2EzODEyYTgxNW
FkOTczZDlhNDk1OTMiLCJleHAiOjE1MjUyNzA2NTIsImlhdCI6MTUyNTI3MDUwMH0.tOt
qrhNTKe8jXSyBx3CEUtoSZiZ5AuLYXZ5j3Z19timCKHJ0b1Tfp9vBysakIPqYn_eVeRH_
Ts40e6jUb5SJZRETtQLrQaX7eFRgn6-36Rg8hBzpljmhDbyxhJlUs14Z4Txa6UFUNWczF
pqfppeGegoQUXPZ0SOdUubiEQtyXgDjHaPeUBk9lp-_y9BcrLGrUsednjMR12-DLwtQEr
ddABi0fpacn5TIPNtAxXXNrWG7w3F75jkUg94eY43y3I9wsi59sfwq1VDiSNG0piWpnSi
8mUPP6RKfko9uOrEjXF3wlUn5tYNllfnzQ_G432lnwwXXh3NRGzjm1w4W7VYKsw
```

Listing 46: Abfrage der Authentifizierungsbestätigung mittels *private_key_jwt* von Vermittler an IdP

In Listing 47 ist der Inhalt des signierten JWTs – aus dem Parameter *client_assertion* – von Listing 46 ersichtlich. Die Parameter des JWTs sind in Kapitel 9 der OpenID Connect Spezifikation [3] definiert.

```
{
  "iss": "vermittler_client_id",
  "sub": "vermittler_client_id",
  "aud": "https://oidc-rp.example.com/token",
  "jti":
"bbd4fdf83d8f3d07903639766eb7a24d3523ca0c83a3812a815ad973d9a49593",
  "exp": 1525270652,
  "iat": 1525270500
}
```

Listing 47: Inhalt des *private_key_jwts* – Authentifizierungsbestätigung Vermittler

In der Folge ist die Token Response des IdPs dargestellt. Um die OAuth 2.0 Spezifikation [7] einzuhalten, MUSS das Access Token ebenfalls Teil der Response sein. Das Access Token wird in diesem Anwendungsfall nicht benötigt und DARF daher KEINE Gültigkeit aufweisen.

```
HTTP/1.1 200 OK
  Content-Type: application/json
  Cache-Control: no-store
  Pragma: no-cache

{
  "access_token": "SlAV32hkKG",
  "token_type": "Bearer",
  "expires_in": 1,
  "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJodHRwc
zovL29pZGMtaWRwLmV4YW1wbGUuY29tIiwic3ViIjoiMjQ4Mjg5NzYxMDAxIiwiYXVkIj
oidmVybWl0dGxlcl9jbGllbnRfaWQiLCJhY3IiOiJlY2gwMTcwLnZzMSIsImV4cCI6MTU
yNTI3MDkyMCwiaWF0IjoxNTI1MjcwNjIwfQ.VTwfvkFzxRCfc8iuU9cWgH_13UD2jxjmT
sNtmFA_MO4pSqxdnXbIWCeobhcyAXyxylzNXN7kiVqc2Qg84x0f5r8o374b6e_oaSysf2
t30P3pIVl_2Vg8MG2oMne2SltWCv0Oo9q-zJ8_6Zg8Km_OjRuIFp9qynAjWNKFNuSSMHX
6pQvov66TnN3lstF17LdUnAuMnxzQ5ns3c1AaQBW4Fk9h9FmK9PT6eiBc49r8oo1_exx4
vsCnLPQxSLOalPbYY1iVh8TFlbd2JQJxUj3csYQZRAQOqLppt30Cr2JM6psAD3o6VRK1Z
hcTU-ksTkCQA9U6JtNCDK2p0-fk_Ui7bw"
}
```

Listing 48: Erfolgreiche Authentifizierungsbestätigung von IdP an Vermittler

Der Vermittler MUSS das ID Token gemäss Kapitel 3.1.3.7 OpenID Connect Spezifikation [3] prüfen.

Nachfolgend (Listing 49) der Payload des ID Tokens aus Listing 48.

```
{
  "iss": "https://oidc-idp.example.com",
  "sub": "248289761001",
  "aud": "vermittler_client_id",
  "acr": "ech0170.vs1",
  "exp": 1525270920,
  "iat": 1525270620
}
```

Listing 49: Inhalt des ID Tokens

9 Schnittstellen für Attribut Provider (AP und IdP/AP)

In diesem Kapitel werden die Schnittstellen zum AP spezifiziert. Der AP wird nur für die Attributübergabe verwendet. In Kapitel 9.1 wird der Ablauf der Attributübergabe definiert, falls der Vermittler Attribute anfragt. In Kapitel 9.2 wird die Attributübergabe definiert, falls eine RP Attribute anfragt. Die Abbildung 15 zeigt eine Übersicht der Schnittstellen.

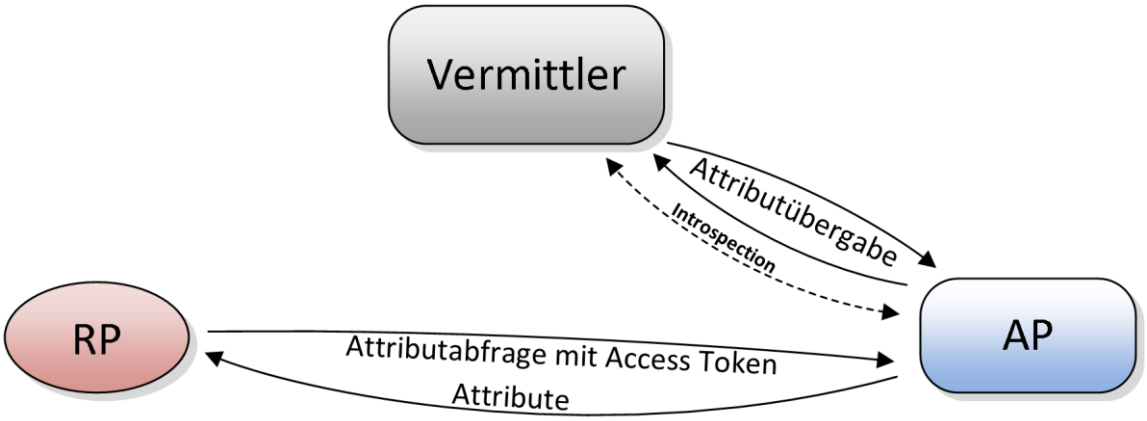

Abbildung 15: Schnittstellen aus Sicht der AP

Authentifizierung mit Attributübermittlung	Attributanfrage von Vermittler	Kapitel 9.1
	Attributbestätigung an Vermittler	Als Normal Claim: - Kapitel 9.3.1 Als Aggregated Claim: - Kapitel 9.3.2
	Attributanfrage von RP	Kapitel 9.2
	Attributbestätigung an RP	Als Normal Claim: - Kapitel 9.3.3

Tabelle 11: Übersicht Spezifikation Schnittstellen AP

9.1 Attributanfrage von Vermittler

Der Vermittler KANN bei einem IdP/AP (Kapitel 9.1.1) oder einem AP (Kapitel 9.1.2) Attribute abfragen.

Im Unterschied zu einem AP, weiss ein IdP/AP welche Access Tokens ausgestellt wurden und hat damit genug Informationen, um eine Attributanfrage zu beantworten.

Ein AP hat keine Kenntnisse über den Authentifizierungszustand des Subjekts. Somit MUSS der Vermittler ein JWT Access Token verwenden, damit der AP weiss, von welchem Subjekt Attribute angefragt werden.

9.1.1 Attributanfrage an einen IdP/AP

In Listing 50 ist eine Attributanfrage vom Vermittler zu einem IdP/AP ersichtlich.

```
GET /userinfo HTTP/1.1
  Host: oidc-idp-ap.example.com
  Authorization: Bearer waG72XqiLqzV
```

Listing 50: Attributabfrage an einen IdP/AP

9.1.2 Attributanfrage an einen AP

Aufgrund des vom IdP erhaltenen ID Tokens, stellt sich der Vermittler ein Access Token für den Attribut Provider – der die Attribute des Subjektes besitzt – aus. Der Inhalt des JWT Access Token ist in Listing 51 ersichtlich. Das JWT Access Token wird analog zu Kapitel 3.2.2.1 aufgebaut. Die Parameter *sub* und *scope* MÜSSEN vorhanden sein, damit der AP das entsprechende Attributset zum Retournieren erkennt.

```
{
  "aud": "ap_client_id",
  "iss": "https://vermittler.example.com",
  "exp": 1525270622,
  "iat": 1525270502,
  "sub": "248289761001",
  "scope": "canton function"
}
```

Listing 51: JWT Access Token für Attributanfrage bei einem unabhängigen AP

Der Vermittler sendet das signierte JWT Access Token im HTTP Basic Authentication Header Parameter des UserInfo Requests an den UserInfo Endpoint des APs.

```
GET /userinfo HTTP/1.1
  Host: oidc-ap.example.com
  Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQi
OiJhcF9jbGllbnRfaWQiLCJpc3MiOiJodHRwczovL3Zlcm1pdHRsZXIuZXhhbXBsZS5jb
20iLCJleHAiOjE1MjUyNzA2MjIsImlhdCI6MTUyNTI3MDUwMiwic3ViIjoiMjQ4Mjg5Nz
YxMDAxIiwic2NvcGUiOiJhdHRyaWJ1dGUifQ.PUctS7SPf4UeEqgjsorYl6zBw4JL2Fiw
w9CvWoOp24U1l7H-4DuVuZBjY3rLHTrQV7kfxbTI4q2LMuEjQofLbzGoUcnNVKBz_6mxE
j1ThWHV3RT0NorTI5Is9Wjv0kS2ZYBgMUdxp2Vl-zINy0-TN2nRbzYGDW1vldGNYauRR1
898Z-9S8H9xOISt8W4PuTpJ8gHChmZPefF7xn2diTgv9ymE4O03bUVx-OhVO1U8PhpuMP
P2hQNVujEO4xf5L589-rfX0wOYkKUlgcMfSLoH-EiR8KlLvFPznOXydyUwNhyKILHOC6Q
ZNFgUBtiSFiSr3anV8sLcR6HTTkv3mUHTQ
```

Listing 52: Attributanfrage an einen unabhängigen AP

Der AP MUSS die Attributanfrage anhand der Signatur des JWTs überprüfen. Ist das Access Token ungültig, MUSS eine Fehlermeldung gemäss Kapitel 10.5 retourniert werden.

Bei erfolgreicher Prüfung MÜSSEN die Attribute in einem Normal Claim und gemäss Richtlinie 2 (Authentizität und Integrität der Antwort) vom AP als signiertes JWT an den Vermittler retourniert werden.

Nachfolgend ist der Inhalt der vom AP retournierten Attributbestätigung (Userinfo Response) ersichtlich. Die Gemeinschaft definierte in diesem Beispiel den Scope Parameter «canton function» mit den beiden Attributen ***function*** und ***canton***, die in der Attributbestätigung retourniert werden.

```
{
  "iss": "https://oidc-ap.example.com",
  "aud": "vermittler_client_id",
  "function": "tester",
  "canton": "bern"
}
```

Listing 53: Inhalt der Attributbestätigung des AP

9.2 Attributanfrage von Relying Party (RP)

Eine Relying Party KANN mit einem Access Token (Kapitel 9.2.1) oder mit einem JWT Access Token (Kapitel 9.2.2) Attribute bei einem AP abfragen.

9.2.1 Anfrage mit Access Token

Die RP KANN mit einem Access Token die Attribute abfragen.

```
GET /attribute-src HTTP/1.1
  Host: oidc-ap.example.com
  Authorization: Bearer 09a312B578Z654
```

Listing 54: Attributanfrage mit Access Token

Da der AP keine Kenntnis vom Access Token hat, MUSS dieser beim Vermittler weitere Informationen über den Introspection Endpoint (siehe Kapitel 9.2.1.1) abfragen.

9.2.1.1 Introspection Endpoint

Der AP fragt beim Introspection Endpoint des Vermittlers zusätzliche Informationen zum Access Token ab.

```
POST /introspect HTTP/1.1
  Host: vermittler.example.com
  Content-Type: application/x-www-form-urlencoded
  Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
  token=09a312B578Z654&token_type_hint=access_token
```

Listing 55: Introspection - Anfrage an Vermittler

Folgend (Listing 56) ist ein Bespiel wie eine Antwort des Introspection Endpoints aussehen könnte. Anhand der Parameter *sub* und *scope* ist ersichtlich, zu welchem Subjekt die RP die entsprechenden Attribute erwartet.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "active": true,
  "iss": "https://vermittler.example.com»,
  "aud": "https://oidc-ap.example.com/attribute-src",
  "scope": "canton function",
  "sub": "248289761001",
  "exp": 1524057621,
  "iat": 1524057591
}
```

Listing 56: Introspection - Antwort vom Vermittler

9.2.2 Anfrage mit JWT Access Token

Das Access Token KANN auch ein signiertes (und optional verschlüsseltes) JWT sein. Dieses MUSS vom Vermittler signiert worden sein.

```
GET /attribute-src HTTP/1.1
  Host: oidc-ap.example.com
  Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3Mi
OiJodHRwczovL3Zlcm1pdHRsZXIuZXhhbXBsZS5jb20iLCJhdWQiOiJodHRwczovL29pZ
GMtYXAuZXhhbXBsZS5jb20vYXR0cmlidXRlLXNyYyIsInNjb3BlIjoiY2FudG9uIGZ1bm
N0aW9uIiwic3ViIjoiMjQ4Mjg5NzYxMDAxIiwiZXhwIjoxNTI0MDU3NjIxLCJpYXQiOjE
1MjQwNTc1OTF9.t3Kjw3sytROgoXzhyiskf8mQZU5sGqu74BdpEOUyfy1DqyHKQNNGGoZ
vvJsA6li6lcFihBbi9UY9nT92NqE0cAyZt7aj4m8-lrswoWLefZUFm7OVQwJbe1rnKO6t
eYIn8tkkNKz02NDnu1hkvimQ2syv4rkDLf1vkHHXvlXsap3-AFJhY4HL7YEEjr3Xv7tdO
U3Tq194th7adCBrAlnQaGEiZqTVowLYS-F65G_UUK0fzAcBolcfyt1hz-62ONiOg9KNxK
7mALvbfHXr-oNs4Pp0YdvSJvS7Jf7JYk56MqJ6lSNlrx-hNn6EA3Zvk-jhcR8fkYdpUWz
WPlCVl0JfLw
```

Listing 57: Attributanfrage mit JWT Access Token

Folgend (Listing 58) ist ein Beispiel wie der Inhalt des JWTs aussehen könnte.

```
{
 "alg": "RS256",
 "typ": "JWT"
}
{
  "iss": "https://vermittler.example.com",
  "aud": "https://oidc-ap.example.com/attribute-src",
  "scope": "canton function",
  "sub": "248289761001",
  "exp": 1524057621,
  "iat": 1524057591
}
```

Listing 58: Inhalt JWT Access Token

9.3 Attributbestätigungen

Die zu retournierenden Attribute MÜSSEN gemäss Richtlinie 2 (Authentizität und Integrität

der Antwort) signiert zurückgesandt werden. Die Attributbestätigung kann je nach Vermittlermodell unterschiedlich aufgebaut sein (siehe Tabelle 7).

9.3.1 Attributbestätigung an Vermittler (Normal Claim)

Die Attributbestätigung als Normal Claim ist die einfachste Form einer Attributbestätigung.

Nachfolgend ist (Listing 59) der Inhalt eines JWTs mit den Attributen von einem IdP/AP. Die Richtlinie 7 (Vorgaben der Attributbestätigung) MUSS erfüllt sein.

```
{
  "iss": "https://oidc-idp-ap.example.com",
  "aud": "vermittler_client_id",
  "sub": "248289761001",
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "preferred_username": "j.doe",
  "email": "janedoe@example.com"
}
```

Listing 59: Normal Claim von einem IdP/AP

Listing 60 zeigt ein Beispiel, wie die Attribute von einem AP an den Vermittler retourniert werden könnten.

```
{
  "iss": "https://oidc-ap.example.com",
  "aud": "vermittler_client_id",
  "sub": "248289761001",
  "function": "tester",
  "canton": "bern"
}
```

Listing 60: Normal Claim von einer AP

9.3.2 Attributbestätigung an Vermittler (Aggregated Claim)

Nachfolgend dargestellt (Listing 63) ist die vom IdP/AP signierte und als Aggregated Claim retournierte Response. Die Richtlinie 7 (Vorgaben der Attributbestätigung) MUSS erfüllt sein. Damit der Vermittler korrekt entsprechend dem Vermittlermodell «Offene Quellen» agieren kann, DARF der Parameter ‘sub’ NICHT im JWT von ‘_claim_sources’ enthalten sein.

```
{
  "iss": "https://oidc-idp-ap.example.com",
  "aud": "vermittler_client_id",
  "sub": "248289761001",
  "_claim_names": {
    "name": "src1",
    "given_name": "src1",
    "family_name": "src1",
    "preferred_username": "src1",
    "email": "src1"
  },
  "_claim_sources": {
    "src1": {"JWT": "eyJ0eXAiOiJqd3QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ
odHRwczovL29pZGMtaWRwLWFwLnZlcm1pdHRsZXIuY29tIiwibmFtZSI6IkphbmUgRG9l
IiwiZ2l2ZW5fbmFtZSI6IkphbmUiLCJmYW1pbHlfbmFtZSI6IkRvZSIsInByZWZlcnJlZ
F91c2VybmFtZSI6ImouZG9lIiwiZW1haWwiOiJqYW5lZG9lQGV4YW1wbGUuY29tIn0.Ms
u9zpmW1wYKmj4a2J5zyG5BN1ECqL2-2burKDNC7mVDGhxxkEHwxfPn44phy3NeE1bt2vv
YtpPUYjv-CtoMNp18ebUb3PVi45SSVqBM9BRG_4b116SNeVK95USuYqFSLfKYTBd0Sohk
ftaSNBT1HOnoOnPQL_QVqMJ4Hq0mWjBY1EcboYk7Utv4LInqXNOIvv_IQH9erQ6LcQ1Md
umfSZnAx9xHLrSuE8ud7jqKfBy8jxQI7Zaqp5BOx5XO2jqE3MRF9p-3YTriOymKP-ZpBC
PNAawBEOt9RKWZaRT5kNFV1yD-bVGmMI-H9ri7WCokzRc0NMnuBOImM6tEBHY7FA"
    }
  }
}
```


Listing 61: Aggregated Claim von IdP/AP

9.3.3 Attributbestätigung an RP

Der Attributbestätigung wird in Form eines JWT an die RP übertragen. Das JWT KANN signiert sein. Nachfolgend sind die Attribute im JSON Format ersichtlich.


```
{
  "name": "Jane Doe",
  "given_name": "Jane",
  "family_name": "Doe",
  "preferred_username": "j.doe",
  "email": "janedoe@example.com"
}
```


Listing 62: Attributwerte im JSON Format

Die Parameter *sub*, *iss* und *aud* DÜRFEN NICHT vorhanden sein.

10 Message Richtlinien

Dieses Kapitel spezifiziert die Messages und deren Parameter, die zwischen den Akteuren ausgetauscht werden und für diesen Standard angepasst wurden. Die Definition der restlichen Parameter ist in der OpenID Connect Spezifikation [3] zu entnehmen.

Dieses Kapitel beschränkt sich – wie die einzelnen Szenarien – auf die in Tabelle 1 (Kapitel 3.1) anwendbaren Flows. Zu retournierende Fehlermeldungen aller Flows sind in Kapitel 10.5 festgelegt.

10.1 Authorization Code Flow

In diesem Kapitel sind die Richtlinien für die Erstellung der Messages des Authorization Code Flows nach der OpenID Connect Spezifikation [3] festgelegt.

10.1.1 Authentication Request

Die folgende Liste definiert die Parameter wie diese in einem Vermittlermodell basierten IAM-System verwendet werden. Ansonsten gilt die detaillierte Spezifikation des Authentication Requests, welche in Kapitel 3.1.2.1 der OpenID Connect Spezifikation [3] spezifiziert ist.

- Der *scope* Parameter MUSS bei gewünschter Attributübergabe, zusätzlich die Scopes der geforderten Attributsets (siehe nachfolgende Erklärung) enthalten.
- Der *prompt* Parameter KANN von der RP bzw. Vermittler für die Reauthentifizierung eines Subjektes verwendet werden (z.B. höhere Vertrauensstufe fordern).
- Der *acr_values* Parameter KANN von der RP und vom Vermittler verwendet werden, um ein Subjekt mit einer bestimmten Vertrauensstufe vom IdP authentifizieren zu lassen. Der Parameter kann insbesondere verwendet werden, wenn mehrere Vertrauensstufen erforderlich sind (z.B. für eine Step-Up Authentication). Die angeforderte Vertrauensstufe MUSS in den Metadaten registriert bzw. enthalten sein. Siehe Richtlinie 4 (Qualitätsmodell für die Authentifizierung).

Die Gemeinschaft KANN spezifische Scopes definieren, die von der RP zur Laufzeit angefragt werden können. Dies geschieht zur Definitionszeit, aber nie dynamisch zur Laufzeit.

Der Vermittler KANN mit Hilfe des *scope* Parameters angeben, in welcher Form die Attribute zu übermitteln sind. Als Beispiel KANN der Scope *signed* verwendet werden, damit die AP das Attributset als Aggregated Claim an den Vermittler sendet. Ein Normal Claim ist die standardmässige Übertragungsform.

10.1.2 Authentication Response

Die Spezifikation der Authentication Response befindet sich in Kapitel 3.1.2.5 der OpenID Connect Spezifikation [3]. Es sind keine Anpassungen aufgrund dieses Standards notwendig.

10.1.3 Token Request

Die Spezifikation des Token Requests befindet sich in Kapitel 3.1.3.1 der OpenID Connect Spezifikation [3].

Für die Authentifizierung einer RP SOLLTE *private_key_jwt* oder als Alternative KANN

client_secret_basic verwendet werden.

Die Authentifizierung eines Vermittlers MUSS über *private_key_jwt* erfolgen.

Die Spezifikation der Authentifizierungsarten und -methoden, befindet sich in Kapitel 9 der OpenID Connect Spezifikation [3].

10.1.4 Token Response

Die Spezifikation der Token Responses befindet sich in Kapitel 3.1.3.3 der OpenID Connect Spezifikation [3].

- Der *expires_in* Parameter MUSS bei Verwendung des Authentication Requestes mit *scope=openid* (lediglich Authentifizierung des Subjektes) den Wert 1 aufweisen, um zu zeigen, dass das mitgesandte *access_token* KEINE Gültigkeit besitzt und NICHT verwendet werden kann.

10.1.5 UserInfo Request

Die Spezifikation des UserInfo Requests befindet sich in Kapitel 5.3.1 der OpenID Connect Spezifikation [3].

10.1.6 UserInfo Response

Die Spezifikation der UserInfo Response befindet sich in Kapitel 5.3.2 der OpenID Connect Spezifikation [3].

Die Response MUSS mindestens als signiertes (Richtlinie 2 - Authentizität und Integrität der Antwort) und KANN optional als verschlüsseltes JWT retrouniert werden. Der Header Type (content-type) MUSS daher *application/jwt* aufweisen. Wird signiert und verschlüsselt, MUSS das JWT gemäss Kapitel 10 der OpenID Connect Spezifikation [3] zuerst signiert und dann verschlüsselt werden - Siehe Richtlinie 2 (Authentizität und Integrität der Antwort). Das Resultat ist ein verschachteltes (nested) JWT. Durch die Signatur sind die nachfolgenden Parameter im JWT zu setzen.

- Der *iss* Parameter MUSS die URL des Ausstellers (IdP oder AP) enthalten.
- Der *aud* Parameter MUSS die *client_id* der RP bzw. des Vermittlers ausweisen.

10.1.6.1 Erweiterungen Vermittler Offenlegung der Quellen

- Die *aud* und *sub* Parameter dürfen in der UserInfo Response der IdP/APs und/oder der APs NICHT enthalten sein (Aggregated Claim).

10.2 Hybrid Flow

Dieses Kapitel legt die Richtlinien für die Erstellung der verschiedenen Messages für den Hybrid Flow nach der OpenID Connect Spezifikation [3] fest. Dabei wird gemäss Tabelle 1 (Kapitel 3.1) auf den Hybrid Flow mit *response_type = code token* eingeschränkt.

10.2.1 Authentication Request

Der Authentication Request erfolgt gemäss dem des Authorization Code Flows (siehe Kapitel

10.1.1 und Kapitel 3.3.2.1 der OpenID Connect Spezifikation [3]).

- Der *response_type* Parameter MUSS beim Hybrid Flow und aufgrund der Einschränkungen aus Tabelle 1, *code token* als Wert aufweisen. Alle anderen Werte dürfen NICHT verwendet werden.

10.2.2 Authentication Response

Die Authentication Response erfolgt gemäss der aus dem Authorization Code Flow (siehe Kapitel 10.1.2 in diesem Standard und Kapitel 3.3.2.5 der OpenID Connect Spezifikation [3]), wobei zusätzlich der *access_token* Parameter retourniert wird.

- Der *access_token* Parameter MUSS gemäss Kapitel 3.3.2.5 der OpenID Connect Spezifikation [3] für den *response_type=code token* ein Access Token für die RP bzw. Vermittler enthalten.

10.2.3 Weitere Hybrid Flow Messages

Alle weiteren Messages eines Hybrid Flows mit *response_type = code token*, erfolgen entsprechend den Messages des Authorization Code Flows aus Kapitel 10.1.

10.3 Introspection

Dieses Kapitel definiert die Richtlinien der Messages beim Austausch einer Introspection nach der OAuth 2.0 Token Introspection Spezifikation (RFC 7662) [8].

10.3.1 Introspection Request

Die Spezifikation des Introspection Requests und dessen Parameter sind in Kapitel 2.1 des RFC 7662 [8] definiert.

10.3.2 Introspection Response

Die Introspection Response erfolgt gemäss Kapitel 2.2 der RFC7662 Spezifikation [8]. Nachfolgend die in diesem Standard angepassten und retournierten Parameter einer erfolgreichen Introspection Response (z. B. gültiges Access Token und erfolgreiche Authentifizierung der abfragenden Instanz).

- Der *sub* Parameter MUSS einen von der abfragenden Instanz bekannten Identifier für das Subjekt aufweisen (Identity Linking).
- Der *iss* Parameter MUSS die URL des Ausstellers des Tokens repräsentieren.

Falls der Introspection Request als ungültig eingestuft wird (z. B. ungültiges Access Token oder eine der weiteren Möglichkeiten aus Kapitel 2.2 des RFC7662 [8]) MUSS eine Fehlermeldung gemäss siehe Kapitel 10.5 gesendet werden.

10.4 Weitere Messages (deskriptiv)

In diesem Kapitel befinden sich Referenzen auf weitere Standards bzw. Drafts, die verwendet werden können. Ob und in welchem Umfang diese Standards zum Einsatz kommen, liegt in der Verantwortung der Gemeinschaft (IAM-Führung).

Session Management Single Logout [10]
OpenID Connect Front-Channel Logout [11]
OpenID Connect Back-Channel Logout [12]

Tabelle 12: Weitere Messages

10.5 Fehlermeldungen

Bei fehlerhaften Requests oder falscher Authentisierung MÜSSEN die Fehlermeldungen (Error Responses) gemäss der OpenID Connect Spezifikation [3] ausgegeben werden.

- Authentication Error Response – Kapitel 3.1.2.6
- Token Error Response – Kapitel 3.1.3.4
- UserInfo Error Response – Kapitel 5.3.3

Fehlermeldungen für einen Introspection Request MÜSSEN gemäss RFC7662 [8] behandelt werden.

- Introspection Error Response – Kapitel 2.3

11 Metadaten Richtlinien

Dieses Kapitel definiert die Metadaten der beteiligten Akteure. Die Metadaten MÜSSEN über eine URL für alle Teilnehmer abrufbar sein und als signiertes JWT retourniert werden.

11.1 Identity Provider

Die Parameter der Metadaten eines IdPs sind gemäss Kapitel 3 der OpenID Discovery Spezifikation [14] definiert. In Listing 63 ist ein Beispiel einer entsprechenden Metadatei im JSON-Format für einen IdP. Erweiterungen und Ausnahmen sind nachfolgend aufgeführt.

- Der *response_types_supported* Parameter MUSS mindestens den Authorization Code Flow (*code*) aufweisen. Zusätzlich unterstützte Response Typen sind in Kapitel 3.1 definiert.
- Der *subject_types_supported* Parameter MUSS in einem Array die unterstützten Identifier Typen – für die Subjekte – festlegen. Die unterstützten Typen sind in Kapitel 3.5 ersichtlich, wobei der *public* Identifier explizit ausgeschlossen wird.
- Der *scopes_supported* Parameter MUSS alle unterstützten Scopes (Scopes von der Gemeinschaft definiert) in einem Array auflisten. *openid* MUSS immer enthalten sein.
- Der *acr_values_supported* Parameter MUSS die vom IdP unterstützten Vertrauensstufen (siehe Richtlinie 4 - Qualitätsmodell für die Authentifizierung) in einem Array aufführen.

Aggiert der IdP als IdP/AP, MÜSSEN die Parameter *userinfo_endpoint* und *userinfo_signing_alg_values_supported* in dessen Metadaten definiert sein. Richtlinie 2 (Authentizität und Integrität der Antwort) fordert signierte Responses (siehe auch Kapitel 11.4).

Um die Verschlüsselung der Responses zu unterstützen, MÜSSEN die Parameter *id_token_encryption_alg_values_supported* und *id_token_encryption_enc_values_supported* Teil der Metadaten sein.

```
{
 "issuer": "https://oidc-rp.example.com",
 "authorization_endpoint": "https://oidc-rp.example.com/authorize",
 "token_endpoint": "https://oidc-rp.example.com/token",
 "userinfo_endpoint": "https://oidc-rp.example.com/userinfo",
 "end_session_endpoint": "https://oidc-rp.example.com/endsession",
 "response_types_supported":
   ["code", "code token"],
 "jwks_uri": "https://oidc-rp.example.com/jwks.json",
 "subject_types_supported": ["pairwise", "transient"],
 "id_token_signing_alg_values_supported": ["RS256", "ES256"],
 "id_token_encryption_alg_values_supported": ["RSA1_5", "A128KW"],
 "id_token_encryption_enc_values_supported": ["A128CBC-HS256"],
 "scopes_supported":
   ["openid"],
 "acr_values_supported": ["ech0170.vs2", " ech0170.vs1"],
 "claim_types_supported": ["normal", "aggregated", "distributed"],
 "token_endpoint_auth_methods_supported": ["private_key_jwt"],
 "token_endpoint_auth_signing_alg_values_supported": ["RS256",
   "ES256"]
}
```

Listing 63 Beispiel Metadaten für IdPs

11.2 Relying Party

Die Metadaten und deren Parameter sind in Kapitel 2 der OpenID Connect Dynamic Registration [15] definiert. In Listing 64 sind die Metadaten der RP anhand eines Beispiels im JSON-Format aufgezeigt.

- Der *response_types* Parameter MUSS einen aus Tabelle 1 aufgelisteten Response Typen enthalten.
- Der *subject_type* Parameter MUSS einer der Werte aus Kapitel 3.5 enthalten.
- Der *id_token_signed_response_alg* Parameter MUSS den für das Signieren der Token Response zu verwendende Algorithmus definieren.
- Der *userinfo_signed_response_alg* Parameter MUSS den für das Signieren der UserInfo Response zu verwendende Algorithmus definieren.
  Der *default_acr_values* Parameter MUSS die von der RP gewünschten Vertrauensstufen (aus Richtlinie 4 - Qualitätsmodell für die Authentifizierung) enthalten.

Sollen die zu übermittelnden Daten zusätzlich geschützt werden, MÜSSEN die Parameter *id_token_encrypted_response_alg*, *id_token_encrypted_response_enc*, *userinfo_encrypted_response_alg* und *userinfo_encrypted_response_enc* in den Metadaten entsprechend gesetzt werden.

```
{
 "redirect_uris":
   ["https://oidc-rp.example.com/service1", "https://oidc-rp.exam-
ple.com/service2"],
 "jwks_uri": "https://oidc-rp.example.com/my_public_keys.jwks",
 "id_token_signed_response_alg": "RS256",
 "id_token_encrypted_response_alg": "RSA1_5",
 "id_token_encrypted_response_enc": "A128CBC-HS256",
 "userinfo_signed_response_alg": "RS256",
 "userinfo_encrypted_response_alg": "RSA1_5",
 "userinfo_encrypted_response_enc": "A128CBC-HS256",
 "token_endpoint_auth_method": "private_key_jwt",
 "token_endpoint_auth_signing_alg": "RS256",
 "subject_type": "pairwise",
 "default_acr_values": ["ech0170.vs1", "ech0170.vs3"],
}
```

Listing 64 Beispiel Metadaten einer RP

11.3 Vermittler

Für die Kommunikation zwischen RP und Vermittler, agiert der Vermittler gegenüber der RP als IdP/AP und MUSS die Metadaten eines IdPs – aus Kapitel 11.1 – ausweisen.
Zusätzlich KANN der Parameter *registered_idps* in den Metadaten erfasst sein, um den RPs die IdPs anzuzeigen, die der Vermittler unterstützt. Dies ist sinnvoll, wenn die RPs eine Authentifizierungsbestätigung von einem gewissen IdP zur Laufzeit erwarten wollen. Beim Vermittlermodell «Double Blinding» DARF der Paramter *registered_idps* NICHT zum Einsatz kommen.
Da der Vermittler als IdP/AP agiert, MÜSSEN die Parameter *userinfo_endpoint* und *userinfo_signing_alg_values_supported* in den Metadaten erfasst sein.

Für alle IdPs, bei denen der Vermittler als RP agiert, MUSS der Vermittler die Metadaten der RP – nach Kapitel 11.2 – implementieren.

11.4 Attribute Provider

Dieses Unterkapitel definiert die Metadaten eines beim Vermittler oder IdP registrierten APs. Ein Beispiel einer Metadatei und deren Parameter ist in Listing 65.

- Der *iss* Parameter MUSS die URL zum jeweiligen AP enthalten und KANN beim Vermittler oder IdP als Identifier verwendet werden. Der *iss* Parameter MUSS gleich dem *iss* Parameter aus der UserInfo Response sein.

Die Parameter *claims_parameter_supported*, *jwks_uri*, *subject_types_supported*, *userinfo_endpoint*, *userinfo_signing_alg_values_supported* und *scopes_supported* weisen dieselben Funktionalitäten und Eigenschaften, wie die Metadaten des IdPs (Kapitel 11.1) auf und MÜSSEN in den Metadaten des APs enthalten sein.

Sollen die zu übermittelnden Daten zusätzlich geschützt werden, MÜSSEN die Parameter *userinfo_encryption_alg_values_supported* und *userinfo_encryption_enc_values_supported* ebenfalls Teil der Metadaten sein (siehe Kapitel 11.1 für die Definition der Parameter).

```
{
 "iss": "https://oidc-ap.example.com",
 "jwks_uri": "https://oidc-ap.example.com/my_public_keys.jwks",
 "userinfo_endpoint": "https://oidc-ap.example.com/userinfo",
 "subject_types_supported": ["pairwise", "transient"],
 "userinfo_signing_alg_values_supported": ["RS256", "ES256"],
 "userinfo_encryption_alg_values_supported": ["RSA1_5", "A128KW"],
 "userinfo_encryption_enc_values_supported": ["A128CBC-HS256",
   "A128GCM"],
 "scopes_supported": ["signed", "scope1", "scope2"],
}
```

Listing 65 Beispiel Metadaten eines AP

12 Haftungsausschluss/Hinweise auf Rechte Dritter

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellen oder welche **eCH** referenzieren, 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.

13 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 weiterentwikkelt 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 & Bibliographie

[1] A. Laube-Rosenpflanzer, A. Spichiger, M. Kunz, T. Kessler, and A. Müller, “eCH-0219 IAM Glossar,” version 1.0, 2019 [Online]. Available: http://www.ech.ch/dokument/b1a76129-0c3e-4021-a6ff-c3123554fd95

[2] A. Laube-Rosenpflanzer, G. Hassenstein, M. Kunz, and B. Leimer, “eCH-0224,” version 1.0, 2018.

[3] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, “OpenID Connect Core 1.0 incorporating errata set 1,” 2014 [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html

[4] A. Laube-Rosenpflanzer, A. Spichiger, M. Kunz, T. Kessler, T. Gruoner, and M. Heerkens, “eCH-0107 Gestaltungsprinzipien für die Identitäts- und Zugriffsverwaltung (IAM),” version 3.0, 2019 [Online]. Available: https://www.ech.ch/dokument/167ccf46-b902-4b58-88f2-eed05ed58c05

[5] E. D. Hardt, “The OAuth 2.0 Authorization Framework [RFC 6749],” 2012 [Online]. Available: https://tools.ietf.org/html/rfc6749

[6] Eidgenössisches Justiz- und Polizeidepartement EJPD, “E-ID-Gesetz.” [Online]. Available: https://www.bj.admin.ch/bj/de/home/staat/gesetzgebung/e-id.html

[7] D. Hardt, “[OAuth 2.0] The OAuth 2.0 Authorization Framework [RFC 6749],” *RFC 6749*, pp. 1–76, 2012.

[8] J. P. Richer, “[OAuth 2.0] OAuth 2.0 Token Introspection [RFC 7662],” *RFC 7662*, pp. 1–17, 2015.

[9] Natsakimura, “OpenID Connect | OpenID.” [Online]. Available: http://openid.net/connect/. [Accessed: 10-Oct-2016]

[10] B. de Medeiros, N. Agarwal, N. Sakimura, J. Bradley, and M. Jones, “OpenID Connect Session Management,” 2017 [Online]. Available: http://openid.net/specs/openid-connect-session-1_0.html

[11] M. Jones, “OpenID Connect Front-Channel Logout,” 2017. [Online]. Available: https://openid.net/specs/openid-connect-frontchannel-1_0.html

[12] M. Jones and J. Bradley, “OpenID Connect Back-Channel Logout,” 2017. [Online]. Available: http://openid.net/specs/openid-connect-backchannel-1_0.html

[13] A. Laube-Rosenpflanzer, G. Hassenstein, M. Kunz, T. Gruoner, A. Spichiger, and T. Selzam, “eCH-0170 Qualitätsmodell zur Authentifizierung von Subjekten,” version 2.0, 2017 [Online]. Available: https://www.ech.ch/alfresco/s/ech/download?nodeid=54cce841-215f-4887-9382-25620dcbf9b1

[14] N. Sakimura, J. Bradley, M. Jones, and E. Jay, “OpenID Connect Discovery 1.0,” 2014 [Online]. Available: http://openid.net/specs/openid-connect-discovery-1_0.html

[15] N. Sakimura, J. Bradley, and M. Jones, “OpenID Connect Dynamic Client Registration 1.0,” 2014 [Online]. Available: http://openid.net/specs/openid-connect-registration-1_0.html

Anhang B – Mitarbeit & Überprüfung

Burger Hans	Adnovum
Hangartner Nick	SwissSign
Hassenstein Gerhard	ICTM / Berner Fachhochschule
Kunz Marc	ICTM / Berner Fachhochschule
Laube-Rosenpflanzer Annett	ICTM / Berner Fachhochschule
Leimer Bojan	ICTM / Berner Fachhochschule

Anhang C – Abkürzungen und Glossar

2FA	2 Factor Authentication
AE	Authorization Endpoint
AP	Attribute Provider
CSP	Credential Service Provider
SE	EndSession Endpoint
IdP	Identity Provider
IE	Introspection Endpoint
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWK	JSON Web Key
JWS	JSON Web Signature
JWT	JSON Web Token
LoA	Level of Assurance
OAuth	Open Authorization
OIDC	OpenID Connect
OP	OpenID Provider
PII	Personally Identifiable Information
PKI	Public Key Infrastructur
PPID	Pairwise Pseudonymous Identifier
RP	Relying Party
SAML	Security Assertion Markup Language
SLO	Single Logout
SSO	Single Sign-On
TLS	Transport Layer Security

TE	Token Endpoint
UE	UserInfo Endpoint
URL	Uniform Resource Locator
URI	Uniform Resource Indicator

Anhang E – Abbildungsverzeichnis

Abbildung 1: Einordnung des Standards eCH-0225 7
Abbildung 2: Übersicht Schnittstellen Authentifizierung 18
Abbildung 3: Übersicht Schnittstellen - Vermittlermodell Double-Blinding 19
Abbildung 4: Übersicht Schnittstellen - Vermittlermodell Offene Quellen 20
Abbildung 5: Übersicht Schnittstellen - Vermittlermodell Vertraulichkeitsschutz 21
Abbildung 6: Übersicht Schnittstellen für RP 22
Abbildung 7: Schnittstellen der RP für Authentifizierung 23
Abbildung 8: Schnittstellen der RP für Authentifizierung mit Attributübergabe 26
Abbildung 9: Schnittstellen RP - Attributbestätigung als Distributed Claim 28
Abbildung 10: Schnittstellen Übersicht Vermittler 30
Abbildung 11: Authentifizierung aus Sicht vom Vermittler 31
Abbildung 12: Attributübergabe aus Sicht des Vermittlers 35
Abbildung 13: Attribute als Distributed Claim vom Vermittler 41
Abbildung 14: Schnittstellen vom IdP 43
Abbildung 15: Schnittstellen aus Sicht der AP 46

Anhang F – Listingsverzeichnis

Listing 1: Beispiel eines generellen ID Tokens 10
Listing 2: Beispiel eines JWT Access Tokens 12
Listing 3: HTTP Redirect Authentication Request Response 23
Listing 4: HTTP-GET Authentication Request 23
Listing 5: Successful Authentication Response 24
Listing 6: HTTP-POST Token Request 24
Listing 7: Inhalt des private_key_jwts – Authentifizierungsbestätigung RP 25
Listing 8: Successful Token Response 25
Listing 9: JWT Payload des ID Tokens 26
Listing 10: HTTP-GET Authentication Request 26
Listing 11: Successful Token Response mit Access Token 27
Listing 12: Attributabfrage an Vermittler 27
Listing 13: Attributbestätigung (von Vermittler an RP) 27
Listing 14: JWT Payload der UserInfo Response mit Aggregated Claim 28
Listing 15: JWT Payload der UserInfo Response mit Distributed Claim 29
Listing 16: Attributabfrage beim AP 29
Listing 17: Attributwerte im JSON Format 29
Listing 18: HTTP Redirect Authentication Request Response 31
Listing 19: HTTP Redirect Authentication Request Response 32
Listing 20: HTTP-GET Authentication Request 32
Listing 21: Successful Authentication Response 32
Listing 22: HTTP-POST Token Request 33
Listing 23: JSON von der Authentifizierungsbestätigung 33
Listing 24: Successful Token Response 34
Listing 25: JWT Payload des ID Tokens 34
Listing 26: ID Token vom Vermittler an die RP 34

Listing 27: Successful Authentication Response 35
Listing 28: Token Anfrage RP an Vermittler 35
Listing 29: Scope Parameter der Authentifizierungsanfragen der RP 36
Listing 30: Authentifizierungsbestätigung mit Access Token 36
Listing 31: Attributabfrage an Vermittler 36
Listing 32: Authentifizierungsbestätigung mit Access Token 37
Listing 33: Attributabfrage an einen IdP/AP 37
Listing 34: JWT Access Token für Attributanfrage bei einem AP 37
Listing 35: HTTP-GET UserInfo Request) 37
Listing 36: Normal Claim von einem IdP/AP 38
Listing 37: Normal Claim von einem AP 38
Listing 38: Aggregated Claim von IdP/AP 39
Listing 39: Attribute als Normal Claim für RP 40
Listing 40: Attribute als Aggregated Claim für RP 40
Listing 41: Attribute als Distributed Claim für RP 41
Listing 42: Introspection Anfrage von AP an Vermittler 41
Listing 43: Introspection Antwort von Vermittler an AP 42
Listing 44: Authentifizierungsanfrage von Vermittler an IdP (Redirect) 43
Listing 45: Authentifizierungsanfrage von Vermittler an IdP (HTTP-GET) 44
Listing 46: Abfrage der Authentifizierungsbestätigung mittels *private_key_jwt* von Vermittler an IdP 44
Listing 47: Inhalt des *private_key_jwts* – Authentifizierungsbestätigung Vermittler 45
Listing 48: Erfolgreiche Authentifizierungsbestätigung von IdP an Vermittler 45
Listing 49: Inhalt des ID Tokens 45
Listing 50: Attributabfrage an einen IdP/AP 47
Listing 51: JWT Access Token für Attributanfrage bei einem unabhängigen AP 47
Listing 52: Attributanfrage an einen unabhängigen AP 47

Listing 53: Inhalt der Attributbestätigung des AP 48
Listing 54: Attributanfrage mit Access Token 48
Listing 55: Introspection - Anfrage an Vermittler 48
Listing 56: Introspection - Antwort vom Vermittler 49
Listing 57: Attributanfrage mit JWT Access Token 49
Listing 58: Inhalt JWT Access Token 49
Listing 59: Normal Claim von einem IdP/AP 50
Listing 60: Normal Claim von einer AP 50
Listing 61: Aggregated Claim von IdP/AP 51
Listing 62: Attributwerte im JSON Format 51
Listing 63 Beispiel Metadaten für IdPs 56
Listing 64 Beispiel Metadaten einer RP 57
Listing 65 Beispiel Metadaten eines AP 58

Anhang G – Tabellenverzeichnis

Tabelle 1: Zur Verfügung stehende Authentication Flows 9
Tabelle 2: ID Token - Parameter 10
Tabelle 3: ID Token - Parameter im Hybrid Flow 11
Tabelle 4: JWT Access Token - Parameter 11
Tabelle 5: OIDC Endpoints zu Anwendungsfällen 13
Tabelle 6: Identifikatoren für Subjekte 13
Tabelle 7: Übersicht Authentifizierung mit Attributübermittlung 19
Tabelle 8: Übersicht Spezifikation Schnittstellen RP 22
Tabelle 9: Übersicht Spezifikation Schnittstellen Vermittler 31
Tabelle 10: Übersicht Spezifikation Schnittstellen IdP 43
Tabelle 11: Übersicht Spezifikation Schnittstellen AP 46

Tabelle 12: Weitere Messages..... 55