eCH-0225 – Identity Federations impliquant un Broker – Implémentation avec OIDC

Nom	Identity Federations impliquant un Broker – Implémentation avec OIDC
eCH-nombre	eCH-0225
Catégorie	**Norme**
Stade	Défini
Version	1.0
Statut	Approuvé
Date de décision	2020-06-04
Date de publication	2020-07-29
Remplace la version	-
Condition préalable	eCH-0107, eCH-0219, eCH-0224, eCH-0170
Annexes	aucune
Langues	Allemand (original), français (traduction)
Auteurs	Groupe spécialisé 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
Editeur / distribution	Association eCH, Mainaustrasse 30, case postale, 8034 Zurich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Condensé

La présente norme décrit la mise en œuvre des modèles de fédération d'identités basés sur les modèles d'Identity Federations impliquant un Broker tirés de la norme eCH-0224 en utilisant OpenID Connect (OIDC). L'objectif est de garantir l'interopérabilité, en particulier pour les Relying Parties dans les scénarios G2G, G2B et G2C. Les interfaces et protocoles nécessaires à cet effet sont définis, et l'accent est mis en particulier sur les adaptations nécessaires d'OIDC. La norme s'adresse en priorité aux architectes informatiques et aux développeurs des composants de l'Identity Federation.

Sommaire

Remarque

En vue d'une meilleure lisibilité et compréhension, seul le genre masculin est utilisé pour la désignation des personnes dans le présent document. Cette formulation s'applique également aux femmes dans leurs fonctions respectives.

1 Statut

Approuvé: Le document a été approuvé par le Comité des experts. Il a pouvoir normatif pour le domaine d'utilisation défini dans le domaine de validité donné.

2 Introduction

Une Identity Federation en cyberadministration permet aux autorités de mettre leurs prestations à la disposition – en ligne – des collaborateurs d'autres autorités et des citoyens ainsi que des organisations et entreprises de leur pays (voir eCH-029 glossaire IAM [1] chapitre 2.62). Les autorités délèguent alors l'authentification et la confirmation d'attributs à différents prestataires de services IAM. Cette délégation repose sur une confiance mutuelle, elle-même basée sur des accords juridiques, organisationnels/architecturaux et techniques

Dans la norme eCH-0224 [2], l'espace réservé aux systèmes d'Identity Federation impliquant un Broker a été restreint aux trois modèles qui suivent, sur la base d'exigences plus rigoureuses de
cyberadministration ainsi que des exigences de protection de la sphère privée des sujets:

- Broker Double Blinding,
- Sources ouvertes Broker
- Protection de la confidentialité Broker.

La présente norme décrit la mise en œuvre de ces trois modèles au moyen d'OpenID Connect (OIDC) [3].

2.1 But du document

L'objectif est de garantir l'interopérabilité entre les différents composants des Identity Federations impliquant un Broker, en particulier pour les Relying Parties dans les scénarios G2G, G2B et G2C. Les interfaces et protocoles nécessaires à cet effet sont définis, et l'accent est mis en particulier sur les adaptations et extensions nécessaires d'OIDC.

2.2 Classification

La norme eCH-0107 [4] regroupe les concepts et documents auxiliaires complémentaires relatifs aux solutions IAM fédérées. Les concepts sont des descriptions concrètes, de ce à quoi ressemble une proposition de solution IAM, et comportent des concepts partiels et des architectures devant être pris en compte pour la mise en œuvre. Ces concepts sont étayés par des documents auxiliaires, qui fournissent des informations complémentaires. Ces derniers peuvent toutefois être pertinents pour plus d'un concept. Les modèles de qualité et de maturité représentés ici sont des exemples de document auxiliaire. La liste de documents auxiliaires n'est est exhaustive.

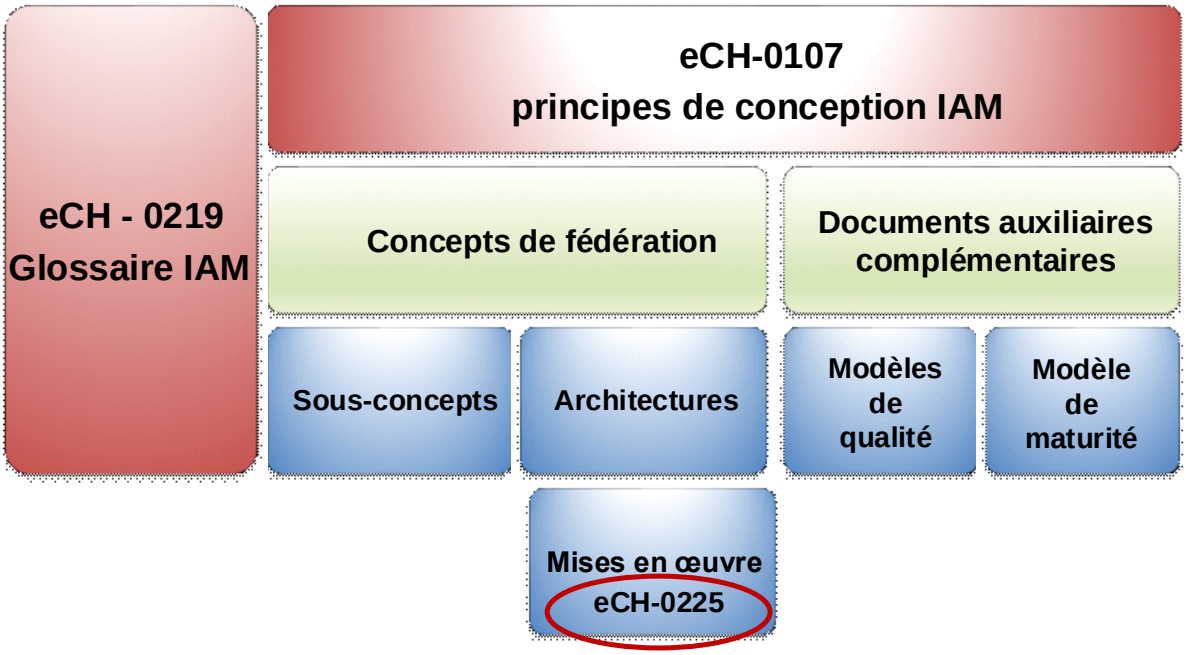

Figure 1: Classification de la norme eCH-0225

La présente norme eCH-0225 décrit la mise en œuvre des modèles d'architecture décrits dans la norme eCH-0224 [2] sur la base technologique d'OIDC [3].

2.3 Groupe cible

La norme est destinée en priorité aux architectes et développeurs informatiques d'Identity Federations ou de composants individuels (RP, Broker, IdP, AP) de systèmes d'Identity Federations dans la cyberadministration. Elle présuppose une connaissance des modèles de Broker de la norme eCH-0224 [2] ainsi qu'une bonne connaissance de la norme [3]..

Les chapitres 3 et 4 définissent les principes fondamentaux d'un système d'Identity Federation avec OIDC et doivent avoir été lus par les architectes informatiques ainsi que les développeurs.

Le chapitre 5 définit la mise en œuvre des modèles de Broker. Ce chapitre est également important pour les architectes informatiques et les développeurs. À l'aide du modèle de Broker à utiliser, il est fait référence aux interfaces à implémenter en conséquence par composant.

Les chapitres 6, 7, 8 et 9 définissent les interfaces des composants et s'adressent aux développeurs qui implémentent les interfaces correspondantes. Les architectes informatiques doivent donc lire ce chapitre pour comprendre les interactions entre les composants.

Les chapitre 10 et 11 fournissent des informations supplémentaires concernant les implémentations des interfaces aux développeurs.

2.4 Champ d'application

La norme décrit la mise en œuvre des modèles d'Identity Federation impliquant un Broker tirés de la norme eCH-0224 [2]. Il s'agit d'une directive concernant l'implémentation des interfaces et des protocoles nécessaires pour les composants individuels d'Identity Federation

afin de garantir l'interopérabilité entre les différents prestataires.

2.5 Délimitation

Cette norme décrit la mise en œuvre des exigences définies par la norme eCH-0224 [2] avec les moyens techniques actuels basés sur OIDC [3]. Seuls les modèles d'Identity Federations impliquant un Broker de eCH-0224 [2] sont pris en compte. Les autres modèles ou variantes d'OIDC [3] ou OAuth [5] ne sont pas abordés.

La mise en œuvre de la protection de la confidentialité (par rapport au Broker) concerne uniquement les confirmations d'attribut, c'est-à-dire qu'aucune confirmation d'authentification (ID Token) cryptée n'est transmise entre IdP et RP.

Dans le cas où un système IAM est mis en œuvre conformément à la LSIE [6], les dispositions légales, ainsi que leurs ordonnances et dispositions d'application, doivent être respectées avant que les directives définies dans les normes eCH puissent être appliquées. En cas de conflit, la LSIE [6] prime.

2.6 Caractère normatif des chapitres

Les chapitres de la présente norme sont de nature soit normative soit descriptive. Le tableau suivant définit la classification des chapitres.

Chapitre	Description
2 Introduction	**Descriptif**
3 Identity Federation basée sur OpenID Connect	**Normatif**
3.6 States, Single SignOn et Single Logout	**Descriptif**
4 Directives générales	**Normatif**
5 Interfaces des modèles de Broker	**Normatif**
6 Interfaces pour le Relying Party (RP)	**Normatif**
7 Interfaces pour le Broker	**Normatif**
8 Interfaces pour l'Identity Provider (IdP)	**Normatif**
9 Interfaces pour l'Attribute Provider (AP et IdP/AP)	**Normatif**
10 Directives concernant les messages	**Normatif**
10.4 Autres messages	**Descriptif**
11 Directives concernant les métadonnées	**Normatif**

Les annexes A et C sont également normatives. Toutes les autres annexes de cette norme sont descriptives.

3 Identity Federation basée sur OpenID Connect

Cette norme est basée sur les normes d'OpenID Connect [3] et d'OAuth 2.0 [5] . Le protocole d'OpenID Connect se distingue par sa grande flexibilité et laisse une grande marge de manœuvre pour la mise en œuvre. Par conséquent, les possibilités offertes par OpenID Connect sont limitées à l'essentiel dans la présente norme, afin que les exigences et les modèles de la norme eCH-0224 [2] puissent être mis en œuvre. Les sous-chapitres suivants décrivent les éléments OpenID Connect utilisés dans cette norme.

3.1 Authentication Flows disponibles

OpenID Connect prend en charge trois flux Authentication Flows distincts, avec chacun des propriétés et des objectifs qui leur sont propres (voir la description des flux au chapitre 3 de la spécification OpenID Connect [3]). Le Tableau 1 récapitule les Flows pour les sections de protocole 1 et 2 (les sections de protocole sont définies au chapitre 8.1.1 de la norme eCH-0224 [2]). A cet égard, les Flows ne répondant pas aux exigences des directives de la norme eCH-0107 [4] et eCH-0224 [2] sont exclus.

N°	*response_type*	Flow	Conflit d'exigence
1.	code	Authorization Code Flow	-
2.	id_token	Implicit Flow	224-LE-5 [2], IAM-11 [4] et LB-15 [4]
3.	id_token token	Implicit Flow	IAM-11 [4] et LB-15 [4]
4.	code id_token	Hybrid Flow	IAM-11 [4] et LB-15 [4]
5.	code token	Hybrid Flow	-
6.	code id_token token	Hybrid Flow	IAM-11 [4] et LB-15 [4]

Tableau 1: Authentication Flows disponibles

Les Flows comportant «id_token» pour «response_type» NE DOIVENT PAS être utilisés. Pour ces Flows, un ID Token est envoyé au RP avant son authentification. Cela va à l'encontre de certaines exigences (voir Tableau 1).

Conformément au Tableau 1, l'Authorization Code Flow (n° 1) ou l'Hybrid Flow (n° 5) DOIT être utilisé. Tous les autres Flows sont exclus et ne sont pas traités plus avant dans ce document.

3.2 Tokens

Dans les sections suivantes, les formats ID Token et Access Token sont définis.

3.2.1 ID Token

Un ID Token est une confirmation d'authentification (en format JWT) et DOIT être signé par l'émetteur. A titre facultatif, l'ID Token PEUT être crypté pour le destinataire. Si l'ID Token doit être signé et crypté, l'ID Token est d'abord signé, puis crypté pour donner un JWT imbriqué (nested JWT).

Les paramètres compris dans l'ID Token sont définis ci-dessous. Le contenu d'un ID Token

est représenté comme exemple dans le Listing 1. Les adaptations pour l'utilisation d'un Hybrid Flow sont décrites au chapitre 3.2.1.1.

Paramètre	Description	Référence
iss	DOIT comporter l'URL de l'émetteur (Issuer) du Token.	OIDC
sub	DOIT comporter un Identifier de sujet clair. Un nouvel Identifier DOIT être utilisé pour chaque RP afin de protéger la vie privée du sujet (voir également le chapitre 3.5).	OIDC
aud	DOIT comprendre le *client_id* du RP (Audience) qui a demandé l'ID Token. La valeur de ce paramètre correspond à un Array qui PEUT comprendre plusieurs *client_ids*.	OIDC
exp	DOIT définir la durée de validité du Token.	OIDC
iat	DOIT comporter la date de création du JWT.	OIDC
acr	DOIT comporter le niveau de confiance avec lequel le sujet a été authentifié auprès de l'Issuer (*iss*).	Directive 4[1]
nonce	PEUT être utilisé comme sécurité supplémentaire contre les attaques Replay et DOIT, si compris, être identique au paramètre *nonce* de l'Authentication Request.	OIDC
source_iss	PEUT comporter l'URL de l'instance de l'ID Token émettant réellement. Pour un scénario impliquant un Broker avec «sources ouvertes», ce paramètre DOIT comporter l'IdP émettant.	Extension d'OIDC
auth_time	PEUT être utilisé conformément à la spécification OIDC[3] .	OIDC
amr	PEUT être utilisé conformément aux spécifications d'OIDC[3] .	OIDC
azp	NE DOIT **PAS** être utilisé	Limitation d'OIDC

Tableau 2: ID Token - Paramètres

L'ID Token NE DOIT comporter AUCUN attribut.

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

Listing 1: Exemple d'un ID Token général

3.2.1.1 Adaptations pour l'utilisation de l'Hybrid Flow

Pour l'utilisation de l'Hybrid Flow, les paramètres suivants DOIVENT être ajoutés à l'ID Token.

[1] Voir chapitre 4: Modèle de qualité pour l'authentification

Paramètre	Description	Référence
nonce	DOIT être inclus.	OIDC
at_hash	DOIT être spécifié lorsqu'un Access Token est renvoyé (voir chapitre 3.3.2.11 de la spécification OpenID Connect [3]).	OIDC
c_hash	Conformément au chapitre 3.3.2.11 de la spécification OpenID Connect [3], ce paramètre est nécessaire si 'code' figure dans le paramètre *response_type* lors de la demande d'authentification. Ce paramètre DOIT donc également être compris dans l'ID Token.	OIDC

Tableau 3: ID Token - Paramètre dans l'Hybrid Flow

3.2.2 Access Token

L'Access Token permet de demander des attributs (Claims) auprès d'UserInfo Endpoints (par ex. auprès d'un AP). Le RP (ou le Broker) obtient un Access Token après l'authentification réussie du sujet. L'Access Token NE DOIT PAS être transmis à des tiers non autorisés.

Il existe deux types d'Access Token:

- le premier est constitué d'une chaîne de caractères conformément à la spécification OAuth [7] chapitre 1.4.
- Le deuxième type est un JWT Access Token et est défini au chapitre 3.2.2.1.

3.2.2.1 JWT Access Token

Un JWT Access Token comprend des informations sur le sujet. La chaîne de caractères DOIT être constituée de données et de la signature (voir l'annexe A.12 dans RFC6749 [7]). Le JWT Access Token DOIT être signé par l'émetteur (Issuer) pour que le destinataire (Audience) puisse vérifier l'intégrité et l'authenticité. Les paramètres du JWT Access Token sont définis comme suit. Un exemple de JWT Access Token figure dans le Listing 2.

Paramètre	Description
iss	DOIT comporter l'Identifier de l'émetteur (par ex. l'URL de l'Issuer) de l'Access Token.
aud	DOIT comporter l'Identifier de l'Audience (par ex. l'URL de l'AP) pour lequel l'Access Token a été émis.
sub	DOIT comporter un Identifier de sujet clair.
iat	DOIT comporter un Integer Timestamp de la date d'émission.
exp	DOIT comporter un Integer Timestamp. La durée de validité du JWT Access Token est ainsi indiquée.
scope	DOIT comporter les Scopes demandés par l'Audience. Dans le cas de plusieurs valeurs Scope, ces dernières doivent être séparées par un espace. Les valeurs Scope possibles figurent dans les chapitres 3.1.2.1 et 5.4 de la spécification OpenID Connect [3].

Tableau 4: JWT Access Token - Paramètres

```
{
 "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: Exemple d'un JWT Access Token

3.3 Claims

Les Claims correspondent aux confirmations d'attribut dans la spécification OpenID Connect [3] pour les confirmations d'attribut. Les trois types de Claims: *Normal Claims*, *Aggregated Claims* et *Distributed Claims* sont définies au chapitre 5.6 de la spécification OpenID Connect [3].

3.3.1 Normal Claims

Pour les Normal Claims, tous les attributs sont issus d'une source. Les attributs du sujet sont regroupés dans un objet JSON.

3.3.2 Aggregated Claims

Les Aggregated Claims PEUVENT être utilisées pour signer des attributs issus de la source d'origine (ils peuvent ainsi être vérifiés) et/ou regrouper les attributs issus de plusieurs sources et les renvoyer sous la forme d'une confirmation d'attributs combinée.

3.3.3 Distributed Claims

Les Distributed Claims PEUVENT être utilisées pour demander directement les attributs du destinataire ou ceux issus de plusieurs sources. Au lieu des valeurs d'attributs, le destinataire reçoit un (des) URL de référence.

3.4 OIDC Endpoints

Le Tableau 5 récapitule les Endpoints définis au chapitre 5 de la norme [2], les RFC 6749 [7] et RFC 7662 [8]. Il illustre les composants implémentés par l'Endpoint correspondant et les Endpoints facultatifs pour les composants.

Les définitions des OpenID Connect Endpoints figurent dans les spécifications suivantes:

- **Authorization Endpoint**: RFC 6749 [7] chapitre 3.1.
- **Token Endpoint**: RFC 6749 [7] chapitre 3.2.
- **UserInfo Endpoint**: Spécification OpenID Connect [9] chapitre 5.3.
- **Introspection Endpoint**: RFC 7662 [8] chapitre 2.

		Applications selon la norme eCH-0224 [2]	
		Authentification pure	Authentification avec transmission d'attributs
OpenID Connect Endpoints	Authorization Endpoint	Broker, IdP	
	Token Endpoint	Broker, IdP	
	UserInfo Endpoint	-	Broker, IdP/AP, AP
	Introspection Endpoint (opt)	-	Broker

Tableau 5: OIDC Endpoints des applications

3.5 identificateurs

La spécification OpenID Connect définit deux types d'identificateurs: *public identificator* et *pairwise identificator*. Le premier correspond à un identificateur réparti (Distributed ID) conformément au chapitre 5.1.3.1 de la norme eCH-0224 [2] et le second à un identificateur lié à un Relying Party (Persistent ID) conformément au chapitre 5.1.3.2 de la norme eCH-0224 [2]. Le *public identificator* NE DOIT PAS être utilisé à cause de l'absence de protection de la sphère privée (voir 224-LB-4 «Non-associativité» dans la norme eCH-0224 [2]).

Les seuls identificateurs employés dans cette norme sont des *Pairwise Pseudonymous Identifier* (PPID) afin de protéger la vie privée des sujets. Avec un PPID, chaque RP reçoit pour chaque sujet un Identifier clair pour lui, qui ne peut être mis en corrélation avec les Identifiers des autres RP. Afin d'aider les différents identificateurs (tirés du chapitre 5.1.3 de la norme eCH-0224 [2]), une définition élargie des types d'identificateurs est nécessaire et est décrite dans le Tableau 6.

Subject Identifier Type	Désignation eCH-0224	Description
public	Distributed ID	Identificateur réparti. **NE** DOIT **PAS** être utilisé
pairwise	Persistent ID	Identifier calculé une seule fois par RP et sujet.
transient	Transient ID	Un nouvel Identifier est calculé par authentification (session) d'un sujet.

Tableau 6: Identificateurs pour sujets

3.6 Single Sign-on et Single Logout (descriptif)

Pour le Single Sign-On et le Single Logout avec OIDC, il n'existe jusqu'à présent aucune expérience ou norme encore établie, encore moins dans le contexte des Identity Federations impliquant un Broker. Le sujet «Session Handling» étant très complexe, une description détaillée serait très longue, aussi n'est-il abordé que brièvement dans ces pages. Lorsque plus d'expériences à ce propos auront été acquises, le sujet «Session Handling, Single Sign-on et Single Logout inclus» devra être abordé en détail dans une norme dédiée.

Single Logout

Pour le Single Logout pour l'OIDC, la norme *Session Management Single* [10] est actuellement uniquement disponible sous forme d'ébauche. Celui-ci complète la spécification OIDC Core [3] pour la surveillance permanente du statut d'authentification des utilisateurs finaux par un IdP. Un RP est à cet effet autorisé à déconnecter un utilisateur dès qu'il s'est déconnecté auprès de l'IdP.

Dans une Identity Federation, le Broker assume le rôle d'IdP vis-à-vis du RP et devrait donc implémenter cette norme.

L'ébauche de la norme *Session Management Single Logout* [10] est complétée par deux autres spécifications étant également au stade d'ébauche:

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

Single Sign-on

Pour le Single Sign-on, il n'existe actuellement aucune norme ou ébauche OIDC précisant le Session Handling d'un IdP (voir également la norme eCH-0224 [2], chap. 3.3.1). Le Session Handling décrit dans l'ébauche de la norme *Session Management Single Logout* [10] pourrait être complété et adapté pour permettre les fonctionnalités nécessaires.

Dans une Identity Federation, le Broker assume le rôle d'IdP vis-à-vis du RP et devrait donc implémenter ces adaptations.

4 Directives générales

Sur la base des exigences pour une Identity Federation en cyberadministration selon les normes eCH-0224 [2] et eCH-0107 [4], les directives suivantes DOIVENT également être respectées pour cette norme.

Ces directives entraînent des limitations ou des extensions concernant la norme [3]. De plus, l'exigence selon laquelle la mesure technique doit être remplie est définie.

Directive 1 – Authentification de l'instance requérante: L'instance ayant reçu la demande DOIT authentifier les composants requérants avant de renvoyer les informations d'identité, d'attribut ou d'authentification d'un sujet.

L'authentification DOIT se dérouler conformément à une méthode définie au chapitre 9 de la norme OpenID Connect [3] (voir aussi chapitre 10.1.3). La méthode *none* NE DOIT PAS être utilisée.

Exigences satisfaites: 224-LE-5, IAM-11 et LB-15.

Directive 2 – Authenticité et intégrité de la réponse: Tous les ID Tokens et Claims renvoyés DOIVENT être signés par l'instance émettrice.

Si le Broker est l'instance d'acceptation, le Broker DOIT vérifier la signature. Le RP DEVRAIT effectuer une vérification de la signature.

Les algorithmes cryptographiques et les longueurs de clé recommandés DEVRAIENT être utilisés dans toutes les instances concernées.[2].

Exigence satisfaite: IAM-9

Directive 3 – Protection de la confidentialité des contenus des attributs: Afin de protéger la vie privée du sujet, l'instance émettrice (source) DEVRAIT émettre les attributs de sorte que seule l'instance autorisée puisse les consulter.

Si la communication a lieu via un Back Channel, le canal DOIT être sécurisé et authentifié.

Les algorithmes cryptographiques et les longueurs de clé recommandés DEVRAIENT être utilisés dans toutes les instances concernées.[2]

Exigence satisfaite: 224-LB-4

Directive 4 – Modèle de qualité pour l'authentification: Pour la qualité de l'authentification des sujets, les niveaux de confiance selon eCH-0170 [13] DOIVENT être utilisés:

- *ech0170.vs1*: Niveau de confiance 1 (confiance nulle ou minimale),
- *ech0170.vs2*: Niveau de confiance 2 (confiance faible),
- *ech0170.vs3*: Niveau de confiance 3 (confiance notable),
- *ech0170.vs4*: Niveau de confiance 4 (confiance élevée).

Le RP DOIT définir le niveau de confiance nécessaire lors de l'enregistrement dans ces métadonnées (voir chapitre 11.2) et PEUT l'envoyer dans la demande d'authentification (si elle

[2] Pour la sélection des paramètres adéquats, se reporter à ces sources: www.keylength.org

a besoin de plusieurs niveaux de confiance).

Le Broker PEUT envoyer le niveau de confiance demandé dans la demande d'authentification.

L'IdP DOIT définir les niveaux de confiance disponibles dans les métadonnées. Si plusieurs niveaux de confiance sont possibles, l'IdP DOIT envoyer également le niveau de confiance utilisé dans la confirmation d'authentification.

Le Broker DOIT dans tous les cas envoyer le niveau de confiance conjointement dans la confirmation d'authentification.

Exigences satisfaites: 224-IAM-4, 224-IAM-4.1

Directive 5 – Modèle de qualité pour la confirmation d'attributs: Pour la qualité de la confirmation d'attributs des sujets, le modèle de niveaux de confiance suivant ou un modèle similaire DOIT être utilisé.

- *ech0224.aq1*: Qualité d'attribut 1 (attributs non confirmés).
- *ech0224.aq2*: Qualité d'attribut 2 (attributs confirmés).
- *ech0224.aq3*: Qualité d'attribut 3 (attributs officiellement confirmés).

Le RP DOIT indiquer au Broker la qualité exigée pour quel attribut lors de l'enregistrement de la ressource.

Le Broker DOIT envoyer conjointement la qualité des attributs au RP dans la confirmation d'attributs. Le Broker reçoit la qualité des attributs soit pour la période d'exécution de la confirmation d'attributs de l'AP soit à partir des métadonnées de l'AP définies pour la période de définition.

Si le Broker demande des attributs auprès de l'AP, l'AP PEUT envoyer conjointement la qualité des attributs au Broker lors de la confirmation d'attributs.

Si un RP demande directement des attributs auprès de l'AP (Distributed Claim), l'AP DEVRAIT envoyer conjointement la qualité des attributs au RP lors de la confirmation d'attributs.

Exigence satisfaite: 224-IAM-5

Directive 6 – Vérification de la demande d'authentification: La demande d'authentification DOIT être vérifiée par un destinataire conformément au chapitre 3.1.2.2 de la spécification OpenID Connect [3]. Si la validation échoue, une annonce d'erreur est renvoyée conformément au chapitre 10.5.

Exigences satisfaites: 224-LB-5, 224-LB-8

Directive 7 - Dispositions concernant la confirmation d'attributs: Dans la confirmation d'attributs (UserInfo Response), les paramètres *sub*, *iss* et *aud* doivent avoir des valeurs identiques à celles de l'ID Token correspondant.

Exigence satisfaite: 224-LE-3.1

Directive 8 – Validation des attributs (User Consent): Le Broker DOIT obtenir l'autorisation du sujet pour que les attributs sélectionnés puissent être transmis au RP.

Exigence satisfaite: 224-LB-6

5 Interfaces des modèles de Broker

Ce chapitre définit la mise en œuvre des trois modèles de Broker de la norme eCH-0224 [2] avec les descriptions des interfaces correspondantes. Seules les interfaces entre les acteurs pour les processus de période d'exécution sont ici spécifiées Seules les interfaces entre les acteurs pour la norme eCH-0224 [2] à partir du chapitre 8.

Dans le chapitre 5.1, une vue d'ensemble des interfaces pour l'authentification est disponible et il est fait référence à la définition d'interface correspondante.

Dans le chapitre 5.2, une vue d'ensemble des interfaces pour l'authentification avec transmission d'attributs est disponible. Puisque les modèles de broker divergent du point de vue de la transmission d'attributs, ils sont précisés individuellement.

5.1 Authentification

Pour un modèle de Broker, l'authentification seule s'effectue en même temps dans les deux sections de protocole. La Figure 2 offre une vue d'ensemble de l'authentification.

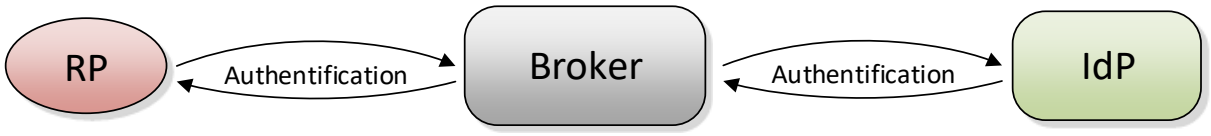

Figure 2: Vue d'ensemble des interfaces Authentification

Le RP envoie une demande d'authentification au Broker (chapitre 6.1.1). Le Broker DOIT valider la demande. Une fois un IdP sélectionné (Discovery), le Broker DOIT envoyer une demande d'authentification à l'IdP (chapitre 7.1.2). Le sujet DOIT être authentifié par l'IdP. L'IdP DOIT envoyer une confirmation d'authentification au Broker en cas d'authentification réussie (chapitre 8.2). Le Broker DOIT transformer la confirmation d'authentification et l'envoyer au RP (chapitre 7.1.4).

5.2 Authentification avec transmission d'attributs

Le Tableau 7 offre une vue d'ensemble de la mise en œuvre de l'authentification avec transmission d'attribut dans les modèles de Broker définis dans la norme eCH-0224. Cela est décrit en détail dans les chapitres suivants.

	Modèles de Broker		
	Double-Blinding	Sources ouvertes	Protection de la confidentialité
RP	Reçoit les attributs du Broker comme Normal Claim. (chapitre 6.2.2)	Visible: Normal Claim avec source comme Attribut. (chapitre 6.2.2)	Reçoit un Distributed Claim du Broker. (chapitre 6.2.4)
		Vérifiable: Aggregated Claim signé par l'AP (IdP/AP). (chapitre 6.2.3)	Les attributs sont demandés directement auprès de l'AP. (chapitre 6.2.4.2)

Broker	Transmet les attributs comme Normal Claim au RP. (chapitre 7.2.4.1)	Visible: Le Broker rend l'AP visible via un attribut. (chapitre 7.2.4.1)	Transmet les attributs comme Distributed Claim avec JWT Access Token
		Vérifiable: Le Broker prend l'Aggregated Claim signée par l'AP et la transmet au RP. (chapitre 6.2.3)	Access Token avec Introspection (chapitre 7.2.4.3)
AP	Transmet les attributs comme Normal Claim au Broker. (chapitre 9.3.1)	Visible: Transmet les attributs comme Normal Claim au Broker. (chapitre 9.3.1)	Envoie les attributs directement au RP. (chapitre 9.3.3)
		Vérifiable: Aggregated Claim, pour lequel le paramètre 'sub' se situe hors des Aggregated Claims. (chapitre 9.3.2)	

Tableau 7: Vue d'ensemble de l'authentification avec transmission d'attributs

5.2.1 Broker Double Blinding

La Figure 3 offre une vue d'ensemble des interfaces. La définition détaillée du modèle de Broker Double Blinding se trouve dans le document eCH-0224 [2] au chapitre 8.2.

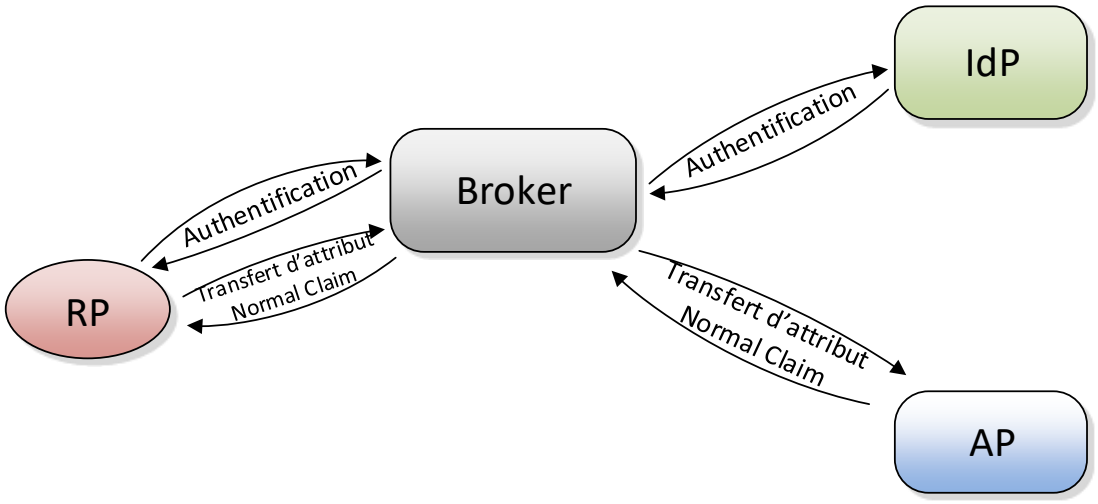

Figure 3: Vue d'ensemble des interfaces - Modèle de Broker Double-Blinding

Le Broker DOIT transmettre les attributs au RP comme Normal Claim (voir chapitre 3.3.1).

L'AP transmet les attributs au Broker. Cela DEVRAIT s'effectuer avec une Normal Claim, mais une Aggregated Claim ou une Distributed Claim est également possible. Il est important que le Broker transforme les confirmations d'authentification et d'attribut de sorte que le RP ne puisse pas identifier l'IdP ou l'AP.

5.2.2 Sources ouvertes Broker

La Figure 4 offre une vue d'ensemble des interfaces pour lesquelles les sources d'attribut sont vérifiables. La définition détaillée du modèle de Broker sources ouvertes se trouve dans le document eCH-0224 [2] au chapitre 8.3.

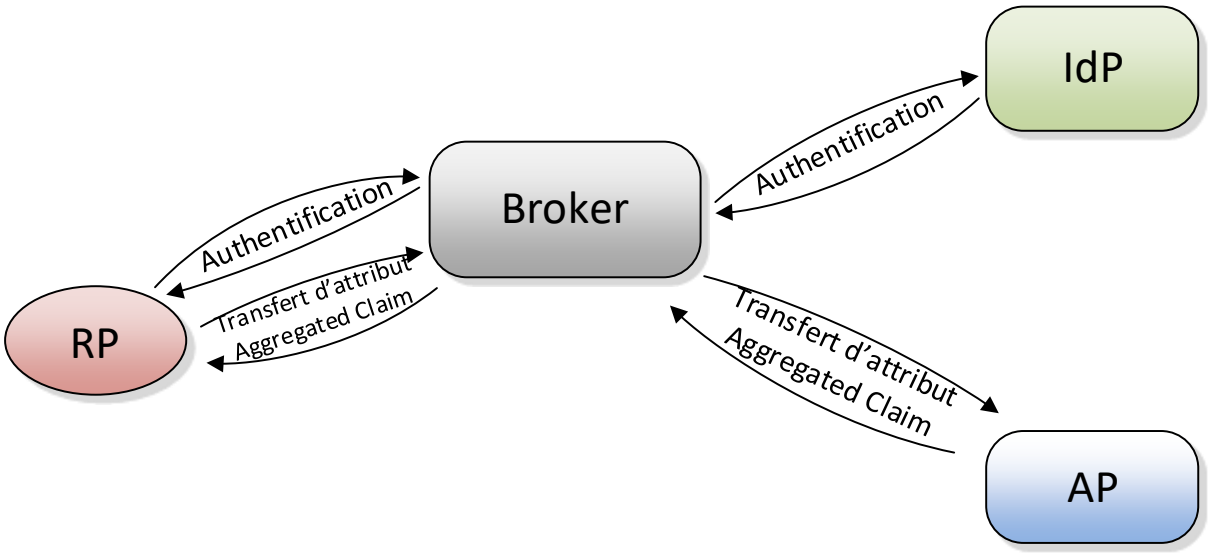

Figure 4: Vue d'ensemble Interfaces - Modèle de Broker sources ouvertes

Si les sources doivent uniquement être rendues visibles, le Broker PEUT procéder comme au chapitre 5.2.1 , mais il DOIT également divulguer la source avec un autre attribut comme Normal Claim.

Afin de pouvoir vérifier les sources d'attribut pour le RP, des Aggregated Claims DOIVENT être utilisées. À cet effet, l'AP DOIT envoyer une Aggregated Claim au Broker. Le Broker DOIT transmettre l'Aggregated Claim au RP. L'AP NE DOIT PAS savoir à qui les attributs ont été envoyés. Inversement, le RP PEUT vérifier la source des attributs.

5.2.3 Protection de la confidentialité Broker

La Figure 5 offre une vue d'ensemble des interfaces pour un Broker avec protection de la confidentialité. La définition détaillée du modèle de Broker Protection de la confidentialité se trouve dans le document eCH-0224 [2] au chapitre 8.4.

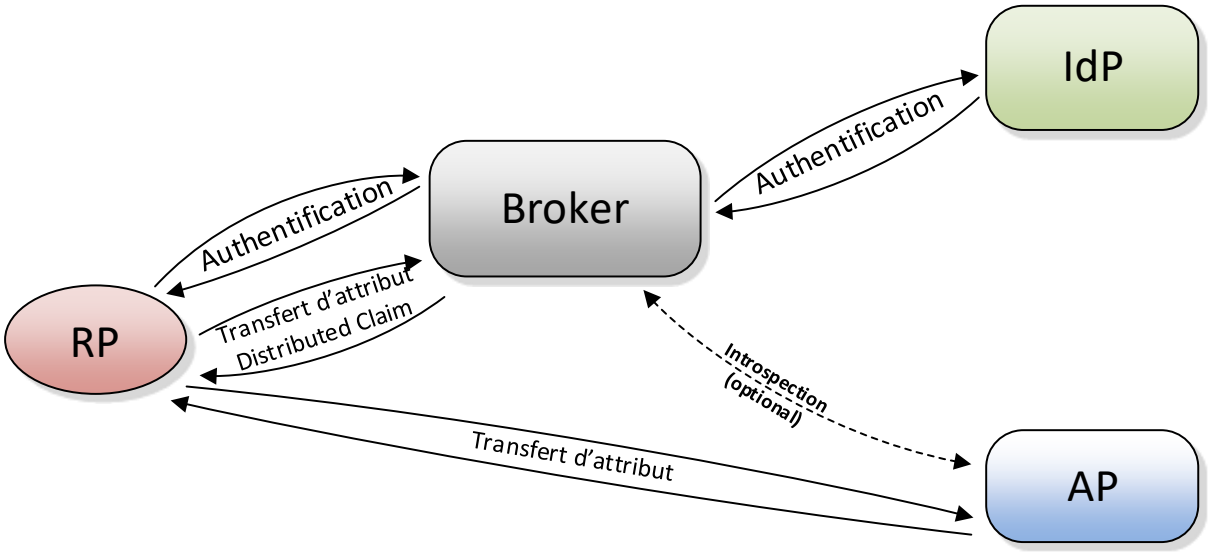

Figure 5: Vue d'ensemble des interfaces - Modèle de Broker protection de la confidentialité

Pour que la confidentialité des attributs soit garantie, le Broker DOIT envoyer une Distributed Claim avec l'Access Token correspondant au RP. Le RP PEUT grâce à cela récupérer directement les attributs auprès de l'AP. Le Broker NE DOIT PAS pouvoir consulter les valeurs d'attribut. Voir directive 3 (Protection de la confidentialité des contenus des attributs).

Le Broker DOIT transmettre un (ou plusieurs) JWT Access Token (voir 3.2.2.1) ou Access Token (voir 3.2.2) comme Distributed Claim.

Si le Broker utilise un JWT Access Token, le JWT Access Token DOIT être signé par le Broker et crypté pour l'AP. Il est ainsi garanti que le RP ne puisse pas consulter les informations concernant le sujet. Si le JWT Access Token est utilisé, l'AP peut comprendre quels attributs doivent être envoyés pour quel sujet. Puisque le JWT Access Token est signé, l'AP PEUT vérifier son authenticité.

Si le Broker utilise un Access Token comme Distributed Claim, le Broker DOIT proposer un Introspection Endpoint pour les AP. Si le RP demande des attributs avec un Access Token, l'AP DOIT demander d'autres informations (sujet, Scope) via l'Introspection Endpoint pour fournir les attributs correspondants au RP.

6 Interfaces pour le Relying Party (RP)

Ce chapitre spécifie les interfaces avec le RP. La procédure d'authentification via une infrastructure de Broker est définie au chapitre 6.1 et l'authentification avec transmission d'attributs au chapitre 6.2 . Le chapitre 6.2.4 définit un cas d'application dans lequel les attributs provenant de plusieurs sources (AP supplémentaires) sont demandés.

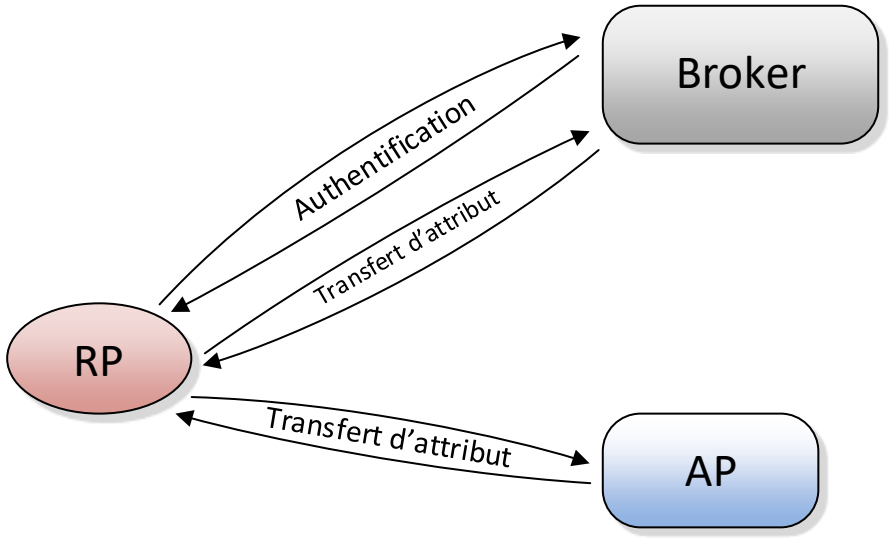

Figure 6: Vue d'ensemble des interfaces pour le RP

Le tableau suivant indique dans quel chapitre est spécifiée la demande (ou réponse) correspondante du RP.

Authentification	Demande d'authentification	chapitre 6.1.1
	Confirmation d'authentification	chapitre 6.1.2
Authentification avec transmission d'attributs	Demande d'attributs	chapitre 6.2.1
	Confirmation d'attributs	Comme Normal Claim (attributs du Broker): - Chapitre 6.2.2 Comme Aggregated Claim (attributs du Broker): - Chapitre 6.2.3 Comme Distributed Claim (référence d'attributs du Broker, attributs de l'AP): - Chapitre 6.2.4

Tableau 8: Vue d'ensemble de la spécification interfaces RP

6.1 Authentification

L'authentification DOIT toujours passer par le Broker. Le résultat d'une authentification réussi DOIT être un ID Token (chapitre 3.2.1).

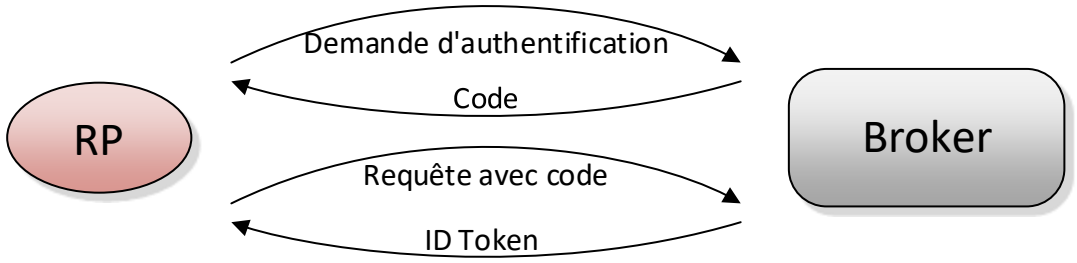

Figure 7: Interfaces des RP pour l'authentification

6.1.1 Demande d'authentification au Broker

Le processus de demande d'authentification est défini selon le chapitre 3.1.2 de la spécification OpenID Connect [3]. Le RP DOIT transmettre le sujet au Broker via la Client Platform. Se reporter au chapitre 10.1.1 [3]pour de plus amples renseignements concernant la demande d'authentification.

```
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

La Client Platform DOIT envoyer une Authentication Request au Broker à l'aide d'un HTTP-GET en raison de la transmission.

```
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

Dès que le sujet a été authentifié avec succès, le Broker DOIT envoyer un Authorization Code au RP via la Client Platform du sujet. La directive 1 (authentification de l'instance requérante) DOIT être respectée.

[3] Le paramètre *acr_value* PEUT également être utilisé dans cette requête si le RP prend en charge plusieurs niveaux de confiance.

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

Listing 5: Successful Authentication Response

Le RP DOIT vérifier la réponse conformément au chapitre 3.1.2.7 de la spécification OpenID Connect [3].

6.1.2 Confirmation d'authentification du Broker

La réception de la confirmation d'authentification est définie selon le chapitre 3.1.3 de la spécification OpenID Connect [3].

Le RP DOIT demander l'ID Token auprès du Token Endpoint du Broker à l'aide de l'Authorization Code reçu. Ci-après figure la Token Request avec *private_key_jwt* en tant que méthode d'authentification. A titre alternatif, *client_secret_basic* PEUT être utilisé en tant que méthode d'authentification (voir chapitre 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

Le contenu du JWT signé – tiré du paramètre *client_assertion* – du Listing 6 figure dans le Listing 7. Les paramètres du JWT sont définis au chapitre 9 de la spécification OpenID Connect [3].

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

Listing 7: Contenu de private_key_jwts – confirmation d'authentification RP

Le Broker DOIT vérifier la demande conformément au chapitre 3.1.3.2 de la spécification OpenID Connect [3]. Si la validation échoue, une annonce d'erreur DOIT être renvoyée conformément au chapitre 10.5 .

La confirmation d'authentification (Token Response) du Broker figure ci-après (dans le Listing 8). Afin de respecter la spécification OAuth 2.0 [7], l'Access Token DOIT également faire partie de la réponse. L'Access Token n'est pas nécessaire pour cette application et ne DOIT donc PAS être valide.

Les Refresh Tokens (spécification OpenID Connect [3] chapitre 12) PEUVENT être utilisés. La responsabilité en incombe à la communauté (direction IAM).

```
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

Le RP DOIT vérifier l'ID Token conformément au chapitre 3.1.3.7 de la spécification OpenID Connect [3].

Ci-après figure le JWT Payload de l'ID Token tiré du Listing 8. L'ID Token a été émis par le Broker pour le RP.

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

Listing 9: JWT Payload de l'ID Token

6.2 Authentification avec transmission d'attributs

Avant de transmettre des attributs au RP, le sujet DOIT être authentifié avec succès. La Figure 8 offre une vue d'ensemble de la communication entre le Broker et le RP. Le RP DOIT recevoir un Access Token avec la confirmation d'authentification (ID Token). À l'aide de l'Access Token, le RP PEUT demander des attributs auprès du Broker.

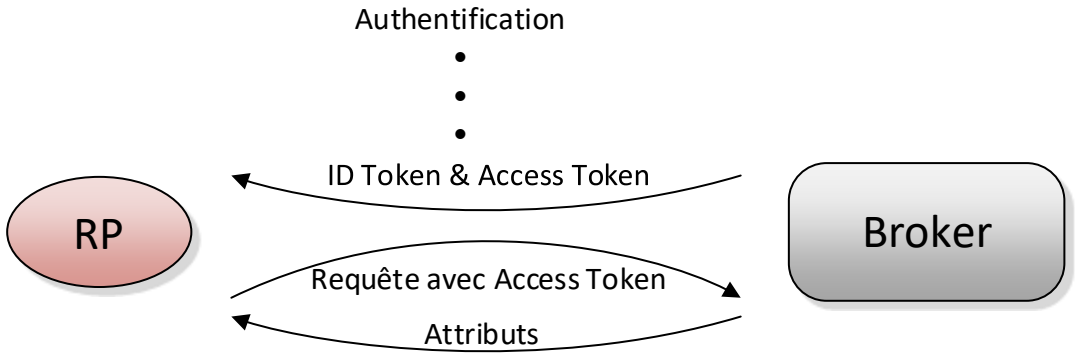

Figure 8: Interfaces du RP pour l'authentification avec transmission d'attributs

6.2.1 Demande d'attributs au Broker

Contrairement au chapitre 6.1.1, le RP DOIT demander le *scope* avec les attributs souhaités lors de l'Authentication Request.

```
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

Le sujet DOIT être transmis au Broker via sa Client Platform comme au chapitre 6.1.1.

Le RP DOIT demander l'ID Token et l'Access Token auprès du Token Endpoint du Broker à l'aide de la Token Request et de l'Authorization Code. La Token Request est identique à celle du chapitre 6.1.1.

Outre l'ID Token, qui doit être au moins signé conformément à la directive 2 (authenticité et

intégrité de la réponse), la Response du Broker doit également contenir un Access Token valide (voir chapitre 4). La validité de l'Access Token DOIT être limitée dans le temps. Dans le Listing 11, la validité de l'Access Token est limitée à une heure (3600 secondes).

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

Listing 11: Successful Token Response avec Access Token

Le Broker DOIT vérifier l'ID Token conformément au chapitre 3.1.3.7 et l'Access Token conformément au chapitre 3.1.3.8 de la spécification OpenID Connect [3] .

Le RP PEUT ensuite demander les attributs auprès de l'UserInfo Endpoint du Broker avec l'Access Token.

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

Listing 12: Demande d'attributs au Broker

En fonction du modèle de Broker, la confirmation d'attributs peut être structurée différemment (voir Tableau 7).

6.2.2 Confirmation d'attributs comme Normal Claim

Dans une Normal Claim, les attributs DOIVENT être envoyés au RP au format JWT signé par le Broker (voir chapitre 3.1.2 de la spécification OpenID Connect [3]). La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée:

Le contenu JWT avec les attributs de l'UserInfo Response est indiqué ci-après.

```
{
  "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: Confirmation d'attributs (du Broker au RP)

6.2.3 Confirmation d'attributs comme Aggregated Claim

Dans une Aggregated Claim, les attributs DOIVENT être envoyés au RP au format JWT par le Broker (voir chapitre 3.1.2 de la spécification OpenID Connect [3]). La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée:

Ci-dessous un exemple possible de JWT.

```
{
  "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 de l'UserInfo Response avec Aggregated Claim

6.2.4 **Confirmation d'attributs** comme Distributed Claim

Si les attributs doivent être demandés auprès d'un autre AP, le RP DOIT recevoir une référence comme Distributed Claim de la part du Broker, non un Access Token. La Distributed Claim est définie conformément au chapitre 5.6.2 de la spécification OpenID Connect [3]. À l'aide de cette référence, le RP PEUT demander directement les attributs auprès de l'AP.

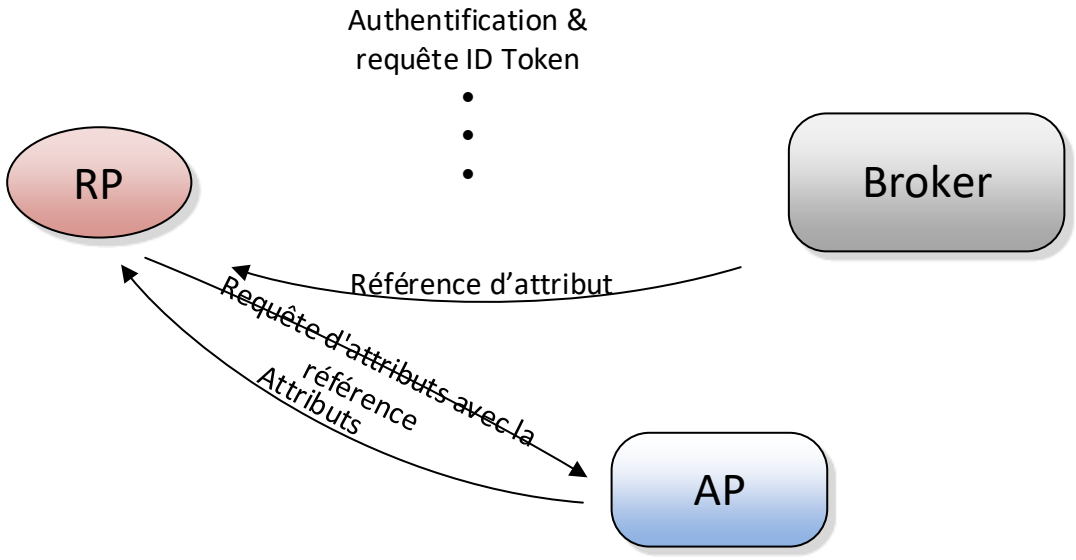

Figure 9: Interfaces RP - Confirmation d'attributs comme Distributed Claim

La réponse de la demande d'attributs auprès du Broker DOIT être une Distributed Claim qui

est envoyée au RP au format JWT signée par le Broker. Le contenu JWT, l'UserInfo Response renvoyée, au format JSON est indiqué ci-après. La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée:

```
{
  "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 de l'UserInfo Response avec Distributed Claim

Le RP PEUT envoyer directement une UserInfo Request à l'AP à l'aide de l'URL reçu (dans *endpoint* d'*ap1_src*) et de l'Access Token. Dans le Listing 15, une chaîne de caractères aléatoire est par exemple utilisée comme Access Token. Un JWT crypté et signé PEUT également être utilisé (voir chapitre 3.2.2).

6.2.4.1 Demande d'attributs à l'Attribute Provider (AP) avec référence

Pour la demande d'attributs, l'information fournie par le Broker dans la Response DOIT être utilisée.

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

Listing 16: Demande d'attributs auprès de l'AP

6.2.4.2 Confirmation d'attributs de l'Attribute Provider (AP)

La confirmation d'attributs DOIT être livrée sous la forme d'un JWT par l'AP. Le JWT PEUT être signé. Si le JWT est signé, la signature DOIT être vérifiée par le destinataire avant que les attributs puissent être envoyés.

Les attributs au format JSON figurent ci-dessous.

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

Listing 17: Valeurs d'attribut au format JSON

Les paramètres *sub*, *iss* et *aud* sont explicitement indisponibles.

7 Interfaces pour Broker

Le Broker est le point final du protocole du RP, de l'IdP et de l'AP. En principe, le RP considère le Broker comme IdP/AP. L'IdP et l'AP considèrent le Broker comme RP.

Le Broker est le lien entre les fournisseurs et les consommateurs d'identité. L'authentification DOIT toujours passer par le Broker. Suivant le modèle de Broker sélectionné, les attributs sont transmis par le Broker ou l'AP au RP. La Figure 10 offre une vue d'ensemble des interfaces depuis et vers le Broker.

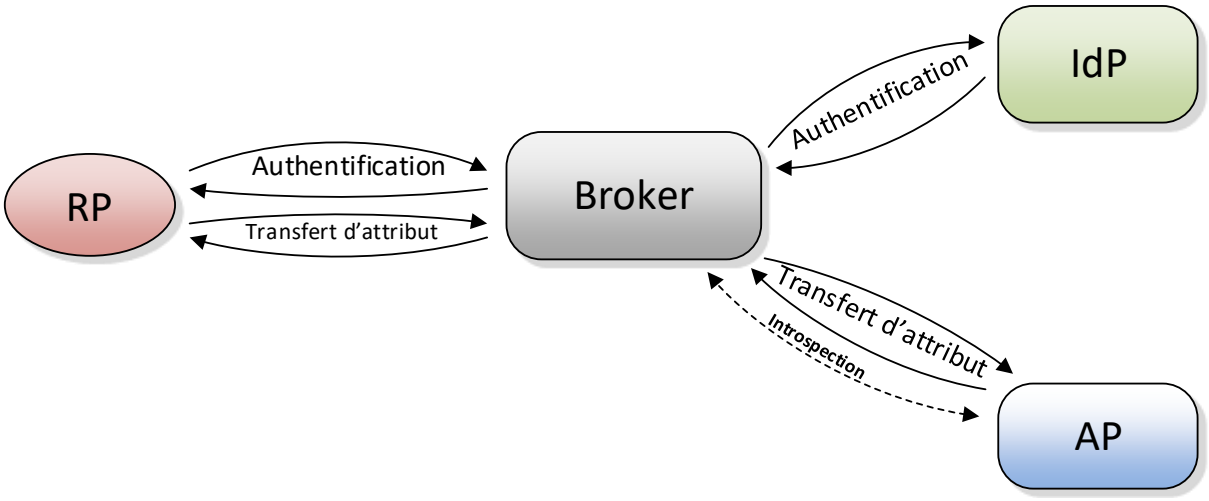

Figure 10: Interfaces vue d'ensemble Broker

Les chapitres suivants spécifient les interfaces. Les messages échangés par application sont définis du point de vue du Broker.

Authentification	Demande d'authentification du RP	chapitre 7.1.1
	Confirmation d'authentification au RP	chapitre 7.1.4
	Demande d'authentification à l'IdP	chapitre 7.1.2
	Confirmation d'authentification de l'IdP	chapitre 7.1.3
Authentification avec transmission d'attributs	Demande d'attributs du RP	chapitre 7.2.1
	Confirmation d'attribut au RP	Comme Normal Claim: - chapitre 7.2.4.1 Comme Aggregated Claim: - chapitre 7.2.4.2 Comme Distributed Claim (et Introspection): - chapitre 7.2.4.3

	Demande d'attributs du Broker	Demande d'attributs à un IdP/AP: - chapitre 7.2.2.1 Demandes d'attributs à un AP: - chapitre 7.2.2.2
	Confirmation d'attribut au Broker	Comme Normal Claim: - chapitre 7.2.3.1 Comme Aggregated Claim: - chapitre 7.2.3.2

Tableau 9: Vue d'ensemble de la spécification interfaces Broker

7.1 Authentification

Le Broker attend une demande d'authentification du RP (chapitre 7.1.1). Une fois un IdP sélectionné pour l'authentification, le Broker DOIT envoyer une demande d'authentification à l'IdP (chapitre 7.1.2).

L'IdP DOIT envoyer un code au Broker après une authentification réussie. Le Broker demande la confirmation d'authentification (ID Token) avec ce code (chapitre 7.1.3).

Puis, le Broker DOIT envoyer un code qu'il a généré au RP. Le RP demande ensuite la confirmation d'authentification avec ce code (chapitre. 7.1.4).

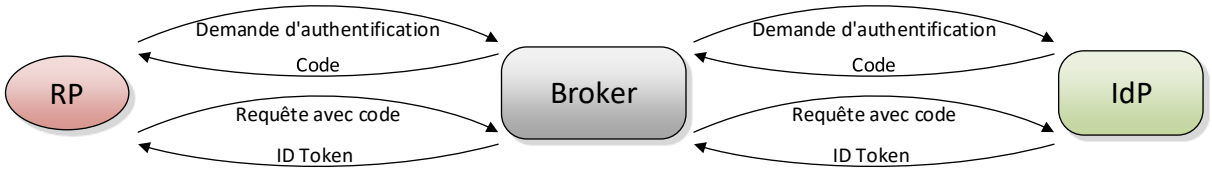

Figure 11: Authentification du point de vue du Broker

7.1.1 Demande d'authentification du Relying Party (RP)

Le Broker reçoit une demande d'authentification du RP. Se reporter au chapitre 10.1.1 pour de plus amples renseignements concernant la demande d'authentification.

```
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

Le Broker DOIT vérifier la demande d'authentification conformément à la directive 6 (vérification de la demande d'authentification).

7.1.2 Demande d'authentification à l'Identity Provider (IdP)

Le Broker envoie une demande d'authentification à l'IdP au moyen de HTTP-GET. Se reporter au chapitre 10.1.1 pour de plus amples renseignements concernant la demande d'authentification.

```
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

Le paramètre *acr_values* PEUT être utilisé pour indiquer à l'IdP la qualité avec laquelle le sujet doit être authentifié.

Le sujet est redirigé vers l'IdP via la Client Platform.

```
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

L'IdP DOIT vérifier la demande d'authentification conformément à la directive 6 (vérification de la demande d'authentification).

7.1.3 Confirmation d'authentification de l'Identity Provider (IdP)

Le sujet est authentifié par l'IdP. Une fois l'authentification réussie, un Authorization Code DOIT être renvoyé au Broker via la Client Platform du sujet. En cas d'annulation de la procédure de connexion par le sujet, une annonce d'erreur DOIT être renvoyée conformément au chapitre 10.5 et non l'Authorization Codes.

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

Listing 21: Successful Authentication Response

Le Broker DOIT vérifier la réponse conformément au chapitre 3.1.2.7 de la spécification OpenID Connect [3].

Le Broker demande l'ID Token auprès du Token Endpoint de l'IdP à l'aide de l'Authorization Code. *private_key_jwt* doit être utilisé comme méthode d'authentification par le Broker. La

définition des paramètres figure au chapitre 9 de la spécification d'OpenID Connect [3]. Il convient également de respecter la directive 1 (authentification de l'instance requérante).

```
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

Le contenu du JWT signé – tiré du paramètre *client_assertion* – du Listing 22 figure dans le Listing 23. Les paramètres du JWT sont définis au chapitre 9 de la spécification OpenID Connect [3].

```
{
  "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 de la confirmation d'authentification

L'IdP DOIT vérifier la demande conformément au chapitre 3.1.3.2 de la spécification OpenID Connect [3]. Si la validation échoue, une annonce d'erreur DOIT être renvoyée conformément au chapitre 10.5 .

La Token Response de l'IdP figure dans le Listing 24. Afin de respecter la spécification OAuth 2.0 [7], l'Access Token DOIT également faire partie de la réponse. L'Access Token n'est pas nécessaire pour cette application et n'est donc pas valide.

```
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

Le Broker DOIT vérifier l'ID Token conformément au chapitre 3.1.3.7 de la spécification OpenID Connect [3].

Le contenu de l'ID Token tiré du Listing 24 figure dans le Listing 25 suivant.

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

Listing 25: JWT Payload de l'ID Token

7.1.4 Confirmation d'authentification au Relying Party (RP)

Le contenu d'une confirmation d'authentification est indiqué dans le Listing 26. Pour conserver le Double Blinding, les paramètres *iss*, *sub* et *aud* DOIVENT être adaptés à la confirmation d'authentification (Listing 25) avant qu'ils ne soient envoyés au RP.

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

Listing 26: ID Token du Broker au RP

Le RP reçoit un code. Le paramètre *state* DOIT être le même que dans le Listing 18.

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

Listing 27: Successful Authentication Response

Le RP demande ensuite l'ID Token à l'aide du code.

```
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: Demande de Token du RP au Broker

Le Broker DOIT vérifier la demande conformément au chapitre 3.1.3.2 de la spécification OpenID Connect [3]. Si la validation échoue, une annonce d'erreur DOIT être renvoyée conformément au chapitre 10.5 .

7.2 Authentification avec transmission d'attributs

Le Broker attend une demande d'attributs du RP (chapitre 7.2.1). Le Broker demande ensuite les attributs auprès de l'AP (ou l'IdP/AP) correspondant (chapitre 7.2.2).

L'AP envoie une confirmation d'attributs au Broker (chapitre 7.2.3). Le Broker transforme la confirmation d'attributs et l'envoie au RP (chapitre 7.2.4).

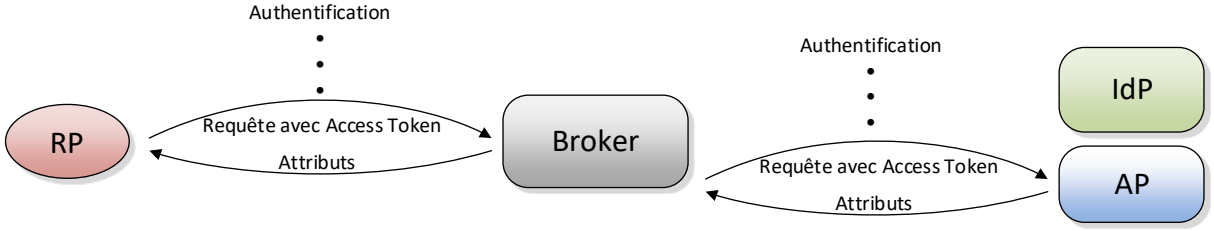

Figure 12: Transmission d'attributs du point de vue du Broker

7.2.1 Demande d'attributs du Relying Party (RP)

À l'aide du paramètre *scope*, les attributs exigés par le RP DOIVENT être indiqués lors de la demande d'authentification.

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

Listing 29: Paramètre scope des demandes d'authentification du RP

Outre l'ID Token, qui DOIT être au moins signé conformément à la directive 2 (authenticité et intégrité de la réponse), la Response de l'IdP/AP doit également contenir un Access Token valide (voir chapitre 3.2.2). La validité de l'Access Token DOIT être limitée dans le temps. Dans le Listing 30, la validité de l'Access Token est limitée à une heure (3 600 secondes).

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

Listing 30: Confirmation d'authentification avec Access Token

Le Broker DOIT vérifier l'ID Token conformément au chapitre 3.1.3.7 et l'Access Token conformément au chapitre 3.1.3.8 de la spécification OpenID Connect [3].

Le RP PEUT ensuite demander les attributs auprès de l'UserInfo Endpoint du Broker avec l'Access Token.

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

Listing 31: Demande d'attributs au Broker

7.2.2 Demandes d'attribut du Broker

Ce chapitre spécifie la demande d'attributs à un IdP/AP (chapitre 7.2.2.1) et un AP (chapitre 7.2.2.2).

7.2.2.1 Demandes d'attribut à un IdP/AP

Pendant la demande d'authentification, les jeux d'attributs demandés DOIVENT être indiqués avec le paramètre *scope*.

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

Outre l'ID Token, qui DOIT être au moins signé conformément à la directive 2 (authenticité et intégrité de la réponse), la Response de l'IdP/AP doit également contenir un Access Token valide (voir chapitre 3.2.2). La validité de l'Access Token DOIT être limitée dans le temps. Dans le Listing 32, la validité de l'Access Token est limitée à une heure (3600 secondes).

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

Listing 32: Confirmation d'authentification avec Access Token

Le Broker DOIT vérifier l'ID Token conformément au chapitre 3.1.3.7 et l'Access Token conformément au chapitre 3.1.3.8 de la spécification OpenID Connect [3].

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

Listing 33: Demande d'attributs à un IdP/AP

7.2.2.2 Demandes d'attributs à un AP

Du fait de l'ID Token reçu par l'IdP, le Broker émet un Access Token pour l'Attribute Provider qui possède les attributs du sujet. Pour la demande, un JWT Access Token DOIT (voir chapitre 3.2.2.1) être utilisé. Les paramètres *sub* et *scope* DOIVENT être disponibles pour que l'AP puisse reconnaître le jeu d'attributs correspondants à renvoyer. Si un Identity Mapping, concernant le Broker, est disponible, le Broker DOIT adapter le paramètre *sub* en conséquence.

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

Listing 34: JWT Access Token pour la demande d'attributs pour un AP

Le Broker DOIT envoyer le JWT Access Token signé dans le HTTP Basic Authentication Header Parameter de la UserInfo Request au UserInfo Endpoint de l'AP.

```
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 Confirmations d'attributs au Broker

Les attributs à renvoyer DOIVENT l'être signés conformément à la directive 2 (authenticité et intégrité de la réponse).

7.2.3.1 Normal Claim

La confirmation d'attributs comme Normal Claim est la forme de confirmation d'attributs la plus simple.

Le contenu d'un JWT avec les attributs d'un IdP/AP est indiqué ci-après (Listing 36) La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée:

```
{
  "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 d'un IdP/AP

Le Listing 37 présente un exemple de la façon dont les attributs pourraient être retournés par un AP au Broker.

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

Listing 37: Normal Claim d'un AP

7.2.3.2 Aggregated Claim

La Response renvoyée sous forme d'Aggregated Claim et signée par l'IdP/AP figure ci-après (Listing 38). La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée:

```
{
  "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 d'IdP/AP

7.2.4 Confirmations d'attribut au RP

Il existe différentes manières de transmettre les attributs du Broker au RP. La méthode utilisée DOIT être évaluée et choisie par la communauté.

Avant que le Broker n'envoie la confirmation, d'attribut au RP, le sujet DOIT valider les attributs. Voir directive 8 – Validation des attributs (User Consent).

La méthode la plus simple est la transmission à l'aide d'une **Normal Claim.** Les attributs sont transmis dans un JWT.

Les **Aggregated Claims** ou **Distributed Claims** peuvent être utilisées pour divulguer les sources ou transmettre les attributs devant être uniquement visibles pour le RP.

La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée dans tous les cas:

7.2.4.1 Transmission de la confirmation d'attributs comme Normal Claim

Le Listing 39 offre un exemple de la façon dont les attributs peuvent être transmis au RP comme Normal Claim.

```
{
  "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: Attributs comme Normal Claim pour RP

7.2.4.2 Transmission de la confirmation d'attributs comme Aggregated Claim

Le Listing 40 offre un exemple de la façon dont les attributs peuvent être transmis au RP comme Aggregated Claim.

```
{
  "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 comme Aggregated Claim pou RP

7.2.4.3 Transmission de la confirmation d'attributs comme Distributed Claim

La Figure 13 offre une vue d'ensemble de la confirmation d'attributs comme Distributed Claim du Broker.

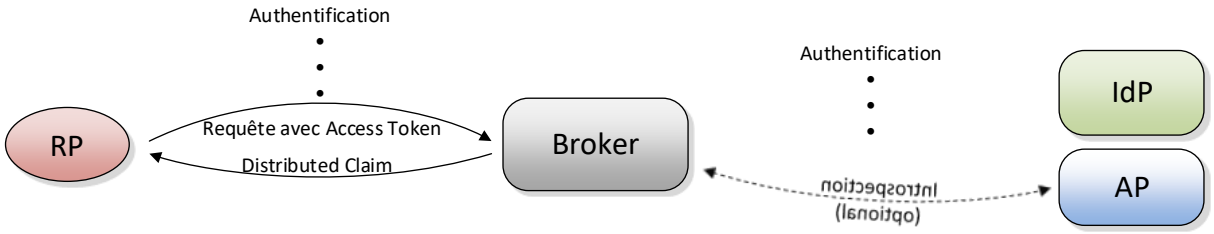

Figure 13: Attributs comme Distributed Claim du Broker

Le Listing 41 offre un exemple de la façon dont les attributs peuvent être transmis au RP comme Distributed Claim.

Un Access Token normal ou un JWT Access Token DOIT être utilisé pour la demande auprès de l'AP.

Si un Access Token normal est utilisé, le Broker DOIT mettre un Introspection Endpoint à disposition pour que l'AP puisse demander à partir de quel sujet les attributs doivent être transmis au RP (voir chapitre 7.2.4.3.1).

Un JWT Access Token contient déjà les informations correspondantes pour l'AP. Le JWT DOIT être signé par le Broker et crypté pour l'AP.

```
{
  "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 comme Distributed Claim pour RP

7.2.4.3.1 Protocole d'introspection

Un Broker PEUT proposer un Introspection Endpoint grâce auquel les AP peuvent demander des informations supplémentaires concernant l'Access Token.

```
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: Demande d'introspection de l'AP au Broker

Le Broker vérifie à l'aide d'une Basic Authentication si l'AP est autorisé à recevoir des informations supplémentaires concernant l'Access Token. Le Broker DOIT ensuite vérifier si l'Access Token est encore valide. Une fois la vérification réussie, le Broker DOIT envoyer un objet JSON. Un exemple de structure du JSON figure ci-après (Listing 43).

```
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 - Réponse du Broker à l'AP

Si la vérification échoue, le Broker DOIT envoyer une annonce d'erreur selon le chapitre 10.5.

8 Interfaces pour l'Identity Provider (IdP)

Ce chapitre spécifie les interfaces pour l'IdP. L'IdP est uniquement utilisé pour l'authentification. En conséquence, seules la demande et la confirmation d'authentification (ID token) sont définies. La Figure 14 offre une vue d'ensemble.

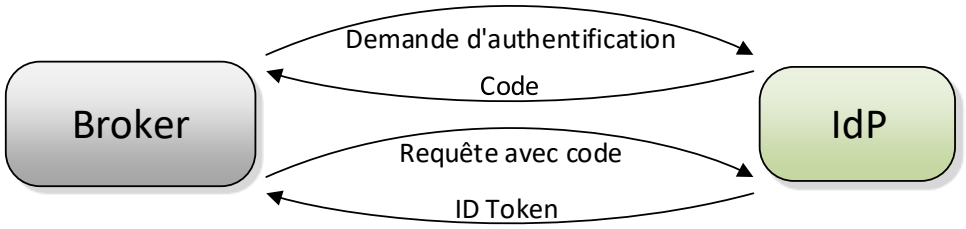

Figure 14: Interfaces de l'IdP

Authentification	Demande d'authentification du Broker	chapitre 8.1
	Confirmation d'authentification au Broker	Chapitre 8.2

Tableau 10: Vue d'ensemble de la spécification interfaces IdP

8.1 Demande d'authentification du Broker

Le Broker envoie une demande d'authentification à l'IdP au moyen de HTTP-GET. Se reporter au chapitre 10.1.1 pour de plus amples renseignements concernant la demande d'authentification.

```
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: Demande d'authentification du Broker à l'IdP (Redirect)

Le paramètre *acr_values* PEUT être utilisé pour indiquer à l'IdP la qualité avec laquelle le sujet doit être authentifié.

Le sujet est redirigé vers l'IdP via la Client Platform.

```
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: Demande d'authentification du Broker à l'IdP (HTTP-GET)

L'IdP DOIT vérifier la demande d'authentification conformément à la directive 6 (vérification de la demande d'authentification).

8.2 Confirmation d'authentification au Broker

Le Broker demande l'ID Token auprès du Token Endpoint de l'IdP à l'aide de l'Authorization Code. *private_key_jwt* doit être utilisé comme méthode d'authentification par le Broker. La définition des paramètres figure au chapitre 9 de la spécification d'OpenID Connect [3]. Voir directive 1 (authentification de l'instance requérante).

```
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: Demande de confirmation d'authentification au moyen de *private_key_jwt* du Broker à l'IdP

Le contenu du JWT signé – tiré du paramètre *client_assertion* – du Listing 46 figure dans le Listing 47. Les paramètres du JWT sont définis au chapitre 9 de la spécification OpenID Connect [3].

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

Listing 47: Contenu de *private_key_jwts* – confirmation d'authentification Broker

La Token Response de l'IdPs est représenté ci-dessous. Afin de respecter la spécification OAuth 2.0 [7], l'Access Token DOIT également faire partie de la réponse. L'Access Token n'est pas nécessaire pour cette application et ne DOIT donc PAS être valide.

```
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: Confirmation d'authentification réussie de l'IdP au Broker

Le Broker DOIT vérifier l'ID Token conformément au chapitre 3.1.3.7 de la spécification OpenID Connect [3].

Le Payload de l'ID Token tiré du Listing 48 figure ci-après (Listing 49)

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

Listing 49: Contenu de l'ID Token

9 Interfaces pour l'Attribute Provider (AP et IdP/AP)

Ce chapitre spécifie les interfaces avec l'AP. L'AP n'est utilisé que pour la transmission d'attributs. Le chapitre 9.1 définit le déroulement de la transmission d'attributs, dans l'éventualité où le Broker demande des attributs. Le chapitre 9.2 définit la transmission d'attributs, dans l'éventualité où le RP demande des attributs. La Figure 15 offre une vue d'ensemble des interfaces.

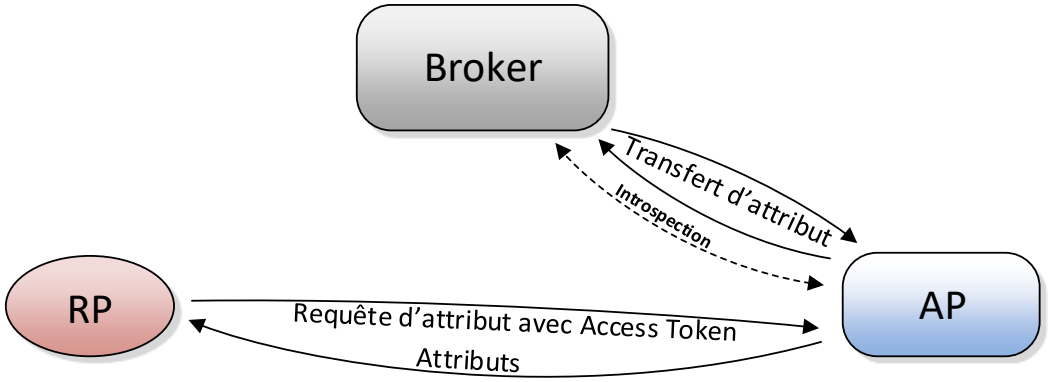

Figure 15: Interfaces du point de vue de l'AP

Authentification avec transmission d'attributs	Demande d'attributs du Broker	chapitre 9.1
	Confirmation d'attribut au Broker	Comme Normal Claim: - chapitre 9.3.1 Comme Aggregated Claim: - chapitre 9.3.2
	Demande d'attributs du RP	Chapitre 9.2
	Confirmation d'attribut au RP	Comme Normal Claim: - chapitre 9.3.3

Tableau 11: Vue d'ensemble de la spécification interfaces AP

9.1 Demande d'attributs du Broker

Le Broker PEUT demander des attributs auprès d'un IdP/AP (chapitre 9.1.1) ou d'un AP (chapitre 9.1.2).

À la différence d'un AP, un IdP/AP sait quels Access Tokens ont été émis et dispose donc d'informations suffisantes pour répondre à une demande d'attributs.

Un AP ne connaît pas l'état d'authentification du sujet. Ainsi, le Broker DOIT utiliser un JWT Access Token pour que l'AP sache par quel sujet les attributs sont demandés.

9.1.1 Demande d'attributs à un IdP/AP

Une demande d'attributs du Broker à un IdP/AP figure dans le Listing 50.

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

Listing 50: Demande d'attributs à un IdP/AP

9.1.2 Demande d'attributs à un AP

Du fait de l'ID Token reçu par l'IdP, le Broker émet pour lui un Access Token pour l'Attribute Provider qui possède les attributs du sujet. Le contenu du JWT Access Token figure dans le Listing 51. Le JWT Access Token est structuré comme au chapitre 3.2.2.1. Les paramètres *sub* et *scope* DOIVENT être disponibles pour que l'AP puisse reconnaître le jeu d'attributs correspondants à renvoyer.

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

Listing 51: JWT Access Token pour demande d'attributs à un AP indépendant

Le Broker envoie le JWT Access Token signé dans le HTTP Basic Authentication Header Parameter de la UserInfo Request au UserInfo Endpoint de l'AP.

```
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: Demande d'attributs à un AP indépendant

L'AP DOIT vérifier la demande d'attributs au moyen de la signature du JWT. Si l'Access Token n'est pas valide, une annonce d'erreur selon le chapitre 10.5 doit être renvoyée.

Une fois la vérification réussie, les attributs DOIVENT être renvoyés au Broker dans une Normal Claim comme JWT signé par l'AP conformément à la directive 2 (authenticité et intégrité de la réponse).

Le contenu de la confirmation d'attributs à renvoyer par l'AP (UserInfo Response) figure ci-après. La communauté a défini dans cet exemple le paramètre Scope «canton function» avec les deux attributs ***function*** et ***canton*** qui seront renvoyés dans la confirmation d'attributs.

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

Listing 53: Contenu de la confirmation d'attribut de l'AP.

9.2 Demande d'attributs du Relying Party (RP)

Un Relying Party PEUT demander des attributs à un AP avec un Access Token (chapitre 9.2.1) ou avec un JWT Access Token (chapitre 9.2.2).

9.2.1 Demande avec Access Token

Le RP PEUT demander les attributs avec un Access Token.

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

Listing 54: Demande d'attributs avec Access Token

Puisque l'AP ne connaît pas l'Access Token, il DOIT demander des informations supplémentaires auprès du Broker via l'Introspection Endpoint (voir chapitre 9.2.1.1).

9.2.1.1 Introspection Endpoint

L'AP demande à l'Introspection Endpoint du Broker des informations complémentaires concernant l'Access Token.

```
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: Demande d'introspection au Broker

Un exemple d'une possible réponse de l'Introspection Endpoint figure ci-après (listing 56). À l'aide des paramètres *sub* et *scope*, il est possible de voir pour quel sujet le RP attend les attributs correspondants.

```
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 - Réponse du Broker

9.2.2 Demande avec JWT Access Token

L'Access Token PEUT aussi être un JWT signé et (à titre facultatif, crypté). Celui-ci DOIT être signé par le Broker.

```
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: Demande d'attributs avec JWT Access Token

Le (Listing 58) est un exemple de ce à quoi pourrait ressembler le contenu du JWT.

```
{
 "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: Contenu JWT Access Token

9.3 Confirmation d'attributs

Les attributs à renvoyer DOIVENT l'être signés conformément à la directive 2 (authenticité et intégrité de la réponse). En fonction du modèle de Broker, la confirmation d'attributs peut

être structurée différemment (voir Tableau 7).

9.3.1 Confirmation d'attributs au Broker (Normal Claim)

La confirmation d'attributs comme Normal Claim est la forme de confirmation d'attributs la plus simple.

Le contenu d'un JWT avec les attributs d'un IdP/AP est indiqué ci-après (Listing 59) La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée.

```
{
  "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 d'un IdP/AP

Le Listing 60 présente un exemple de la façon dont les attributs pourraient être retournés par un AP au Broker.

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

Listing 60: Normal Claim d'un AP

9.3.2 Confirmation d'attributs au Broker (Aggregated Claim)

La Response renvoyée sous forme d'Aggregated Claim et signée par l'IdP/AP figure ci-après (Listing 63). La directive 7 (dispositions concernant la confirmation d'attributs) DOIT être respectée. Pour que le Broker puisse opérer correctement le modèle de un Broker «sources ouvertes», le paramètre «sub» NE DOIT PAS figurer dans le JWT de «_claim_sources».

```
{
  "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 d'IdP/AP

9.3.3 Confirmation d'attributs au RP

La confirmation d'attribut est transférée au RP sous forme de JWT. Le JWT PEUT être signé. Les attributs au format JSON figurent ci-dessous.


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


Listing 62: Valeurs d'attribut au format JSON

Les paramètres *sub*, *iss* et *aud* NE DOIVENT PAS être présents.

10 Directives concernant les messages

Ce chapitre précise les messages et leurs paramètres qui sont échangés entre les acteurs et ont été adaptés dans l'intérêt de la présente norme. La définition des paramètres restants figure au chapitre 9 de la spécification d'OpenID Connect [3].

Ce chapitre est limité - comme les différents scénarios - aux Flows applicables dans le Tableau 1 (chapitre 3.1). Les annonces d'erreur à renvoyer pour tous les Flows sont définis au chapitre 10.5.

10.1 Authorization Code Flow

Ce chapitre définit les directives pour la création des messages de l'Authorization Code Flow selon la spécification OpenID Connect [3].

10.1.1 Authentication Request

La liste suivante définit les paramètres de leur utilisation dans un système IAM basé sur un modèle de Broker. Dans le cas contraire, la spécification détaillée de l'Authentication Requests, qui est indiquée au chapitre 3.1.2.1 de la spécification OpenID Connect [3] s'applique.

- Le paramètre *scope* DOIT également inclure les Scopes des jeux d'attributs demandés (voir l'explication ci-après) pour la transmission d'attribut souhaitée.
- Le paramètre *prompt* PEUT être utilisé par le RP ou le Broker pour la réauthentification d'un sujet (par ex. demander un niveau de confiance plus élevé).
- Le paramètre *acr_values* PEUT être utilisé par le RP et le Broker pour authentifier un sujet avec un niveau de confiance spécifique de l'IdP. Le paramètre peut en particulier être utilisé si plusieurs niveaux de confiance sont nécessaires (par ex. pour une Step-Up Authentication). Le niveau de confiance demandé DOIT être enregistré ou inclus dans les métadonnées. Voir la directive 4 (modèle de qualité pour l'authentification).

La communauté PEUT définir des Scopes spécifiques qui peuvent être demandés au RP pour la période d'exécution. Cela se déroule pour la période de définition, mais jamais de manière dynamique pour la période d'exécution.

Le Broker PEUT indiquer la forme sous laquelle les attributs doivent être transmis à l'aide du paramètre *scope*. Le scope PEUT par exemple utilisé *signed* afin que l'AP envoie le jeu d'attributs sous forme d'Aggregated Claim au Broker. Une Normal Claim est la forme de transmission par défaut.

10.1.2 Authentication Response

La spécification de l'Authentication Response se trouve au chapitre 3.1.2.5 de la spécification OpenID Connect [3]. Aucune modification à cause de cette norme n'est nécessaire.

10.1.3 Token Request

La spécification du Token Request se trouve au chapitre 3.1.3.1 de la spécification OpenID Connect [3].

Pour l'authentification d'un RP, *private_key_jwt* DEVRAIT utilisé ou, à titre d'alternative, *client_secret_basic* devrait être utilisé.

L'authentification d'un Broker DOIT se dérouler via *private_key_jwt*.

La spécification des types et méthodes de l'authentification se trouve au chapitre 9 de la spécification OpenID Connect [3].

10.1.4 Token Response

La spécification des Token Responses se trouve au chapitre 3.1.3.3 de la spécification OpenID Connect [3].

- Le paramètre expires_in DOIT comporter la valeur 1 lors de l'utilisation d'une Authentication Request avec scope=openid (authentification du sujet uniquement) pour montrer que l'access_token envoyé n'est PAS valide et ne peut PAS être utilisé.

10.1.5 UserInfo Request

La spécification de l'UserInfo Request se trouve au chapitre 5.3.1 de la spécification OpenID Connect [3].

10.1.6 UserInfo Response

La spécification des UserInfo Responses se trouve au chapitre 5.3.2 de la spécification OpenID Connect [3].

La Response DOIT au moins être renvoyée sous forme de JWT signé (directive 2 – authenticité et intégrité de la réponse) et PEUT, à titre facultatif, être renvoyée cryptée. Le Header Type (content type) DOIT donc comporter *application/jwt*. S'il doit être signé et crypté, le JWT DOIT d'abord être signé, puis crypté, conformément au chapitre 10 de la spécification OpenID Connect [3] - voir la directive 2 (authenticité et intégrité de la réponse). Le résultat est un JWT imbriqué (nested). Les paramètres suivants dans JWT doivent être définis grâce à la signature.

- Le paramètre *iss* DOIT comporter l'URL de l'émetteur (IdP ou AP).
- Le paramètre *aud* DOIT comporter le client_id du RP ou du Broker.

10.1.6.1 Extensions Broker divulgation des sources

- Les paramètres *aud* et *sub* ne doivent PAS être inclus dans l'UserInfo Response de l'IdP/AP et/ou de l'AP (Aggregated Claim).

10.2 Hybrid Flow

Ce chapitre stipule les directives pour la création des différents messages de l'Hybrid Flow selon la spécification OpenID Connect [3]. Il se limite donc à l'Hybrid Flow avec *response_type=code token* conformément au Tableau 1 (chapitre 3.1).

10.2.1 Authentication Request

L'Authentication Request est effectuée conformément à l'Authorization Code Flow (voir chapitre 10.1.1 et chapitre 3.3.2.1 de la spécification OpenID Connect [3]) [3]).

- Le paramètre *response_type* DOIT comporter *code token* comme valeur pour l'Hybrid Flow et en raison des limitations tirés du Tableau 1. Toutes les autres valeurs ne doivent PAS être utilisées.

10.2.2 Authentication Response

L'Authentication Response se déroule conformément à l'Authorization Code Flow (voir chapitre 10.1.2 de la présente norme et chapitre 3.3.2.5 de la spécification OpenID Connect [3]), le paramètre *access_token* étant également renvoyé.

- Le paramètre *access_token* DOIT comporter un Access Token pour le RP ou le Broker pour le *response_type=code* conformément au chapitre 3.3.2.5 de la spécification OpenID Connect [3] für den *response_type=code token.*

10.2.3 Autres Hybrid Flow Messages

Tous les autres messages d'un Hybrid Flow avec *response_type=code token* se déroulent conformément aux messages de l'Authorization Code Flow tiré du chapitre 10.1.

10.3 Introspection

Ce chapitre définit les directives concernant les messages lors de l'échange d'une Introspection conformément à la spécification Token Introspection (RFC 7662) d'OAuth 2.0 [8].

10.3.1 Introspection Request

La spécification de l'Introspection Request et de leurs paramètres sont définis au chapitre 2.1 de la RFC 7662 [8].

10.3.2 Introspection Response

L'Introspection Response se déroule conformément au chapitre 2.2 de la spécification RFC7662 [8]. Les paramètres adaptés dans cette norme et renvoyés (par ex. un Access Token valide et une authentification réussie de l'instance requérante) d'une Introspection Response réussie figurent ci-après.

- Le paramètre *sub* DOIT inclure un Identifier connu de l'instance requérante pour le sujet (Identity Linking).
- Le paramètre *iss* DOIT représenter l'URL de l'émetteur du Token.

Si l'Introspection Request est classée comme invalide (par ex. Access Token invalide ou une des autres possibilités issues du chapitre 2.2 du RFC7662 [8]) un message d'erreur doit être envoyé conformément au chapitre 10.5.

10.4 Autres messages (descriptif)

Ce chapitre comporte des références à d'autres normes ou ébauches pouvant être utilisées. Si et dans quelle mesure ces normes sont utilisées relève de la responsabilité de la communauté (direction IAM).

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

Tableau 12: Autres messages

10.5 Annonces d'erreur

En cas de Requests erronées ou d'authentifications incorrectes, les annonces d'erreur (Error Responses) DOIVENT être émises conformément à la spécification OpenID Connect [3].

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

Les annonces d'erreur pour une Introspection Request DOIVENT être traitées conformément au RFC7662 [8].

- Introspection Error Response – chapitre 2.3

11 Directives sur les métadonnées

Ce chapitre définit les métadonnées des acteurs impliqués. Les métadonnées DOIVENT être accessibles à tous les participants via une URL et renvoyées sous forme de JWT signé.

11.1 Identity Provider

Les paramètres des métadonnées d'un IdP sont définis conformément au chapitre 3 de l'OpenID Discovery Specification [14] . Un exemple d'un métafichier correspondant au format JSON pour un IdP figure dans le Listing 63. Les extensions et les exceptions sont indiquées ci-après.

- Le paramètre *response_types_supported* DOIT au moins comporter l'Authorization Code Flow (code). De plus, les types de Response pris en charge sont définis au chapitre 3.1.
- Le paramètre *subject_types_supported* DOIT définir les types d'Identifier pris en charge dans un Array, pour les sujets. Les types pris en charge sont indiqués au chapitre 3.5 l'Identifier *public* est explicitement exclu.
- Le paramètre *scopes_supported* DOIT lister tous les Scopes pris en charge (Scopes définis par la communauté) dans un Array. *openid* DOIT toujours être inclus.
- Le paramètre *acr_values_supported* DOIT lister les niveaux de confiance pris en charge par l'IdP (voir directive 4 - modèle de qualité pour l'authentification) dans un Array.

Si l'IdP agit comme IdP/AP, les paramètres *userinfo_endpoint* et *userinfo_signing_alg_values_supported* DOIVENT être définis dans les métadonnées. La directive 2 authenticité et intégrité de la réponse) exige des Responses signées (voir aussi chapitre 11.4).

Pour prendre en charge le cryptage des Responses, les paramètres *id_token_encryption_alg_values_supported et id_token_encryption_enc_values_supported* DOIVENT faire partie des métadonnées.

```
{
 "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 Exemple de métadonnées pour les IdPs

11.2 Relying Party

Les métadonnées et leurs paramètres sont définis au chapitre 2 de l'OpenID Connect Dynamic Registration [15]. Les métadonnées du RP sont indiquées dans le Listing 64 à l'aide d'un exemple au format JSON.

- Le paramètre *response_types* DOIT comporter un des types de Response listés dans le Tableau 1.
- Le paramètre *subject_type* DOIT contenir une des valeurs figurant au chapitre 3.5 .
- Le paramètre *id_token_signed_response_alg* DOIT définir l'algorithme à utiliser pour signer la Token Response.
- Le paramètre *userinfo_signed_response_alg* DOIT définir l'algorithme à utiliser pour la signature de l'UserInfo Response.
  Le paramètre *default_acr_values* DOIT contenir les niveaux de confiance souhaités par le RP (tiré de la directive 4 - Modèle de qualité pour l'authentification).

Si les données à transmettre doivent également être protégées, les paramètres *id_token_encrypted_response_alg, id_token_encrypted_response_enc, userin-fo_encrypted_response_alg et userinfo_encrypted_response_enc* DOIVENT être définis en conséquence dans les métadonnées.

```
{
 "redirect_uris":
   ["https://oidc-rp.example.com/service1", "https://oidc-
rp.example.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 Exemple de métadonnées d'un RP

11.3 Broker

Pour la communication entre le RP et le Broker, le Broker opère comme IdP/AP par rapport au RP et DOIT présenter les métadonnées d'un IdP, chapitre 11.1.
Le paramètre *registered_idps* PEUT en outre figurer dans les métadonnées pour montrer aux RP les IdP que le Broker prend en charge. Cela est pertinent lorsque les RP veulent attendre une confirmation d'authentification d'un certain IdP concernant la période d'exécution.
Pour le modèle de Broker «Double Blinding», le paramètre *registered_idps* NE DOIT PAS être utilisé.
Puisque le Broker agit comme IdP/AP, les paramètres *userinfo_endpoint* et *userinfo_signing_alg_values_supported* DOIVENT figurer dans les métadonnées.

Pour tous les IdP pour lesquels le Broker opère comme RP, le Broker DOIT implémenter les métadonnées du RP conformément au chapitre 11.2.

11.4 Attribute Provider

Ce sous-chapitre définit les métadonnées d'un AP enregistré auprès du Broker ou de l'IdP.
Un exemple de métafichier et de ses paramètres figure dans le Listing 65.

- Le paramètre *iss* DOIT comporter l'URL de l'AP correspondant et PEUT être utilisé comme Identifier auprès du Broker ou de l'IdP. Le paramètre *iss* DOIT être identique au paramètre *iss* issu d'UserInfo Response.

Les paramètres *claims_parameter_supported, jwks_uri, subject_types_supported, userinfo_endpoint, userinfo_signing_alg_values_supported* et *scopes_supported* présentent les mêmes fonctionnalités et propriétés que les métadonnées de l'IdP (chapitre 11.1) et DOIVENT figurer dans les métadonnées de l'AP.

Si les données à transmettre doivent également être protégées, les paramètres *userinfo_encryption_alg_values_supported et userinfo_encryption_enc_values_supported* DOIVENT en outre faire partie des métadonnées (voir chapitre 11.1 pour la définition des paramètres).

```
{
 "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 exemple de métadonnées d'un AP

12 Exclusion de responsabilité - droits de tiers

Les normes élaborées par l'Association **eCH** et mises gratuitement à la disposition des utilisateurs ainsi que les normes de tiers adoptées, ont seulement valeur de recommandations. L'Association **eCH** ne peut en aucun cas être tenue pour responsable des décisions ou mesures prises par un utilisateur sur la base des documents qu'elle met à disposition. L'utilisateur est tenu d'étudier attentivement les documents avant de les mettre en application et au besoin de procéder aux consultations appropriées. Les normes **eCH** ne remplacent en aucun cas les consultations techniques, organisationnelles ou juridiques appropriées dans un cas concret.

Les documents, méthodes, normes, procédés ou produits référencés dans les normes **eCH** peuvent le cas échéant être protégés par des dispositions légales sur les marques, les droits d'auteur ou les brevets. L'obtention des autorisations nécessaires auprès des personnes ou organisations détentrices des droits relève de la seule responsabilité de l'utilisateur.

Bien que l'Association **eCH** mette tout en œuvre pour assurer la qualité des normes qu'elle publie, elle ne peut fournir aucune assurance ou garantie quant à l'absence d'erreur, l'actualité, l'exhaustivité et l'exactitude des documents et informations mis à disposition. La teneur des normes **eCH** peut être modifiée à tout moment sans préavis.

Toute responsabilité relative à des dommages que l'utilisateur pourrait subir par suite de l'utilisation des normes **eCH** est exclue dans les limites des règlementations applicables.

13 Droits d’auteur

Tout auteur de normes **eCH** en conserve la propriété intellectuelle. Il s'engage toutefois à mettre gratuitement, et pour autant que ce soit possible, la propriété intellectuelle en question ou ses droits à une propriété intellectuelle de tiers à la disposition des groupes de spécialistes respectifs ainsi qu'à l'Association **eCH** pour une utilisation et un développement sans restriction dans le cadre des buts de l'association.

Les normes élaborées par les groupes de spécialistes peuvent, moyennant mention des auteurs **eCH** respectifs, être utilisées, développées et déployées gratuitement et sans restriction.

Les normes **eCH** sont complètement documentées et libres de toute restriction relevant du droit des brevets ou de droits de licence. La documentation correspondante peut être obtenue gratuitement.

Les présentes dispositions s’appliquent exclusivement aux normes élaborées par **eCH**, non aux normes ou produits de tiers auxquels il est fait référence dans les normes **eCH**. Les normes incluront les références appropriées aux droits de tiers.

Annexe A – Références & 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] Département fédéral de la justice et de la police DFJP, «Loi E-ID.» [Online]. Available: https://www.bj.admin.ch/bj/fr/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

Annexe B – Collaboration & vérification

Burger Hans	Adnovum
Hangartner Nick	SwissSign
Hassenstein Gerhard	ICTM / Haute école spécialisée de Berne
Kunz Marc	ICTM / Haute école spécialisée de Berne
Laube-Rosenpflanzer Annett	ICTM / Haute école spécialisée de Berne
Leimer Bojan	ICTM / Haute école spécialisée de Berne

Annexe C – Abréviations et glossaire

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 Infrastructure
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

Annexe E – Liste des illustrations

Annexe F – Liste des listings

Annexe G – Liste des tableaux