EDUCATIEVE CONTENTKETEN

DISTRIBUTIE EN TOEGANG VAN DIGITALE LEERMIDDELEN

Technisch Model

Auteur(s)	:	Zie documentgeschiedenis
Versienummer	:	Zie documentgeschiedenis
Totstandkoming	:	Dit document is tot stand gekomen in samenwerking met vertegenwoordigers van aanbieders en afnemers van digitaal leermateriaal

EduStandaard

Documentgeschiedenis

Versie	Datum	Auteur	Omschrijving
1.0	april 2012	Erwin Reinhoud	Eerste versie .
1.1	02 mei 2012	Wijnand Derks	Aanpassingen nav afspraken LiMBO-programma
1.11	15 mei 2012	Erwin Reinhoud	Verwerking wijzigingen functioneel model versie 1.11
1.2	22 mei 2012	Erwin Reinhoud, Wijnand Derks	Naar Engels vertalen van functiehuizen, operaties en attributen
1.3	24 mei 2012	Erwin Reinhoud	Verwerking wijzigingen functioneel model versie 1.3
1.4	juni 2012	Erwin Reinhoud	Namespace Common types aangepast Versie aanduidingen aangepast AdministrationId weggelaten, deze is in deze versie enkel bedoeld om identifiers uniek te maken binnen de keten.
1.5	december 2012	Lennaert van der Linden	Extra verwijzing naar Algemene specificaties toegevoegd voor beschrijving datatypen DistributionService::ReadEducationalContentList levert alle relevante items indien geen criteria worden meegegeven Wijzigingen verwerkt van functioneel model versie 1.5
1.6 werkversie	23 september 2013	Mark Dobrinic	Samenvoegen van alle services in een document en diverse technische uitwerkingen (zie betreffende bijlage).

Inhoudsopgave

1. Inleiding en leeswijzer

In dit document worden de technische specificaties van ECK-DTDL beschreven. Het document sluit hierbij aan op het Functioneel Model. De technische specificaties zijn gebaseerd op een tweetal modellen, namelijk een model voor Service to Service communicatie dat is gebaseerd op Web Services, en een model voor gebruikers authenticatie dat is gebaseerd op SAML2.

In het eerste deel van dit document worden de algemene uitgangspunten beschreven van de wijze waarop de Web Services worden gebruikt binnen ECK-DTDL. Daarnaast wordt het algemene gedrag van Web Services gespecificeerd. De daadwerkelijke Service-specificatie wordt beschreven in WSDL en XSD bestanden, die als bijlage bij dit document worden opgenomen.
Het tweede deel beschrijft hoe gebruikers authenticatie plaatsvindt, en is gesplitst in een toelichting bij de uitgangspunten, als ook en beschrijving van het proces.
Tenslotte wordt de technische onderbouwing gegeven bij ECK-DTDL specifieke kenmerken.

2. Service to Service specificatie

De functies van Services die elkaar onderling aanroepen, zijn gebaseerd op Web Services. In dit hoofdstuk worden de technische uitgangspunten voor de koppelvlakken van deze Services in het algemeen beschreven. Tevens zijn er per specifieke Service ook WSDL en XSD bestanden beschikbaar die de technische details van elke Service specificeren.

2.1 Algemene principes

Principe (1)	**De gegevensuitwisseling tussen Services is op basis van Web Services. Het berichtformaat hierbij is SOAP 1.1**
Uitleg	Door koppelvlakken op basis van een beperkt aantal standaarden te implementeren worden beheer- en implementatielasten beperkt.
Rationale	Er zijn vele manieren en standaarden om service georiënteerd gegevens uit te wisselen. Een populaire manier van gegevensuitwisseling binnen een enterprise keten is SOAP 1.1. Deze standaard is volwassen en kent vele aanvullende standaarden op basis waarvan andere functies geïmplementeerd kunnen worden. Zie ook: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

Principe (2)	**Web services worden beschreven op basis van de WSDL standaard versie 1.1**
Uitleg	Een WSDL beschrijving kan door software toolkits geconsumeerd worden om automatische de koppelvlakken te genereren, inclusief een beschrijving van de gebruikte invoer- en uitvoergegevens. Zie ook: http://www.w3.org/TR/wsdl
Rationale	De de-facto standaard om SOAP 1.1 web services te beschrijven is WSDL versie 1.1. De functies moeten eenduidig beschreven kunnen worden. Afnemers kunnen deze beschrijving gebruiken bij de implementatie van het koppelvlak

Principe (3)	**Bij het opstellen van een WSDL worden de elementen op basis van Upper Camel Case genoteerd.**
Uitleg	Upper Camel Case is een vaak toegepaste notatiewijze om eenduidig termen in technische documenten, zoals WSDL, te definiëren (zie ook: http://en.wikipedia.org/wiki/CamelCase). Een WSDL wordt beschreven op basis van XML waarbij de elementen door kleine en hoofdletters beschreven kunnen worden. In het geval van Upper Camel Case, begint een term altijd met een hoofdletter.
Rationale	Toolkits gaan verschillend om met het verwerken van kleine en hoofdletters. Om problemen hiermee te beperken worden de onderdelen van een WSDL op een standaard manier beschreven. Een standaard structuur verhoogt ook de leesbaarheid van de standaard.

Principe (4)	**Er wordt één Web Service per Service gedefinieerd**
Uitleg	Per Service wordt een Web Service beschreven in een WSDL specificatie. Deze WSDL specificatie beschrijft de individuele functies van en Service als Operations van een Web Service.
Rationale	Er zijn momenteel geen redenen om meerdere web service per Service te definiëren. Nieuwe beveiligingsprincipes of performance optimalisaties zouden een reden kunnen zijn om een Service verder op te delen in meerdere Web Services. Dit is momenteel niet aan de orde. Op basis van PartyId's en afspraken rond namen voor Web Services kunnen Web Services eenvoudiger gevonden worden. Hoe minder Web Services nodig zijn, hoe eenduidiger dit uitgevoerd kan worden.

2.2 Transport

Principe (1)	**Als communicatiekanaal wordt internet toegepast**
Uitleg	De onderwijssectoren PO, VO en MBO beschikken niet over een eigen afgesloten ketennetwerk. Partijen moeten elkaar goed kunnen bereiken en beheerkosten beperkt blijven. Services die communiceren met ketenpartners gebruiker hiervoor het internet netwerk
Rationale	Bij het ontbreken van een eigen netwerk ligt toepassing van internet voor de hand.

Principe (2)	**De berichten worden over HTTP 1.1 of HTTPS verstuurd.**
Uitleg	Een gangbaar protocol om SOAP berichten te kunnen versturen is HTTP 1.1. Dit protocol wordt al in de keten toegepast. Indien de berichtuitwisseling versleuteld dient te worden, wordt HTTP/1.1 over SSLv3/TLS als transportprotocol gebruikt.
Rationale	Voor de wijze van uitwisselen van SOAP berichten is geen beperking opgelegd, echter voor de SOAP toepassing binnen ECK worden de kenmerken van HTTP/1.1 gewaardeerd, zoals dat HTTP door firewalls goed gefilterd kan worden, er foutcorrect faciliteiten zijn, etcetera. Daarbij is HTTP het meest gebruikte en ondersteunde protocol voor de uitwisseling van SOAP berichten. Alternatieven zoals SMTP minder vaak toegepast, en de wijze om dit goed toe te passen zijn vaak minder goed bekend dan voor het toepassen van HTTP. Zie specificatie HTTP: https://tools.ietf.org/html/rfc2616

Principe (3)	**Bij communicatie over het internet wordt port 80 of 443 toegepast**
Uitleg	Bij beveiligde gegevensuitwisseling met toepassing van TLS/SSL wordt port 443 gebruikt. Bij geen beveiliging op transportkanaal wordt poort 80 toegepast
Rationale	Omdat de transportprotocollen HTTP en HTTPS gebruikt worden, worden de standaard IP poorten van deze protocollen gebruikt.

Principe (4)	**Versleuteling van de gegevensuitwisseling tussen Services wordt op transportkanaal toegepast.**
Uitleg	Berichten worden via HTTP/1.1 over SSLv3/TLS verstuurd. SSLv3/TLS zorgt voor de versleuteling van alle gegevens die worden uitgewisseld.
Rationale	Versleuteling van berichten in transport is eenvoudiger te realiseren dan beveiliging op applicatieniveau. End to end versleuteling van protocolberichten is niet vereist, omdat het transport naar Services altijd eindigt binnen het domein van de ontvangende deelnemer. Dit is de reden dat transportkanaal versleuteling toereikend is.

2.3 Diacrieten, tekenset en encoding

Principe (1)	**Er gelden geen beperkingen aan de te gebruiken karakters anders dan dat ze tot de Unicode karakterset moeten behoren.**
Uitleg	Er moeten afspraken gemaakt worden gemaakt met betrekking tot de toegestane karakters bij gegevensuitwisseling. De Unicode karakterset is ook bekend als 'Universal Character Set' of ook als 'ISO 10646'.
Rationale	Unicode wordt breed ondersteund: in de gangbare Besturingssystemen, in XML en HTML, in Java en .Net. Hierbij wordt vooralsnog geen subset gedefinieerd voor de karakters die niet gebruikt worden. Deze aanscherping kan eventueel in later stadium uitgevoerd worden.

Principe (2)	**Een SOAP bericht wordt volgens UTF-8 ge-encodeerd.**
Uitleg	Voor een succesvolle gegevensuitwisseling moet er naast de toegestane karakters ook afspraken rond de (de)encoding gemaakt worden
Rationale	UTF-8 is een zeer gangbare encoding voor Unicode en wordt zeer goed ondersteund. Gebruik van UTF-8 garandeert dat alle nodige inhoud zoals diakritische tekens en het Euroteken ook daadwerkelijk gecodeerd kunnen worden. Ook gegevens die in een bericht gecodeerd dienen te zijn, vallen binnen het UTF-8 formaat.

2.4 Service en Client Identificatie

Principe (1)	**Iedere Rechtspersoon bepaalt een ketenbreed unieke PartyId. Dit PartyId wordt door een Client gebruikt ter identificatie bij systeem-naar-systeem (S2S) koppelingen.**
Uitleg	Een Rechtspersoon kan met een PartyId uniek worden geïdentificeerd.
Rationale	Het PartyId wordt als basis voor vertrouwensrelaties gebruikt voor toegang tot Services.

Principe (2)	**Een Service wordt geïdentificeerd met zijn endpoint.**
Uitleg	Het operationele endpoint (URL) van een Service is tegelijkertijd zijn identifier.
Rationale	Services dienen uniek geïdentificeerd te kunnen worden.

Principe (3)	**Een endpoint van een Service is duurzaam. Eventuele wijzigingen dienen aan de afnemers te worden gecommuniceerd**
Uitleg	Endpoints dienen zo duurzaam mogelijk te zijn. Bij wijzigingen dienen alle afnemers van de dienst geïnformeerd worden.
Rationale	De endpoints liggen aan de basis van de gegevensuitwisseling. Wijzigingen hierin kunnen dit ernstig verstoren. De versie van een versie bepaalt eenduidig hoe operaties moeten worden aangeroepen, en moeten als zodanig onderdeel uitmaken van een Service identifier. Het ontkoppelen van de Service van de versie zou leiden tot tabel naar de te gebruiken endpoints per versie, er is geen reden om dit nu zo te doen.

Principe (4)	**Iedere aparte versie van een Service heeft een verschillend endpoint.**
Uitleg	Indien een nieuwe versie van een Service beschikbaar wordt gesteld aan partners, dan is deze toegankelijk via een verschillend endpoint dan de oude Service.
Rationale	Er wordt vanuit gegaan dat verschillende partners op dezelfde momenten in de tijd verschillende versies van de ECK-standaard gebruiken. Dit betekent doorgaans dat er meerdere versies van dezelfde Service in gebruik zijn. Dit maakt verschillende endpoints voor verschillende versies noodzakelijk.

Principe (5)	**Iedere Rechtspersoon communiceert zijn PartyId, zijn AdministrationId's en de endpoints van de relevante Services per AdministrationId.**
Uitleg	Vooraf aan het gebruik van een service dient e.e.a. geregeld te zijn rond authenticatie, autorisatie en gebruiksvoorwaarden. Bij deze afspraken zullen KetentPartners ondermeer elkaars identificerend gegeven uitwisselen (PartyId, Service Identifiers) als basis voor vertrouwen. Per AdministrationId worden de endpoints van de relevante Services gecommuniceerd.
Rationale	De vertrouwensbasis wordt gelegd via de PartyId . De werkrelatie wordt gelegd met AdministrationId's en Service-endpoints. Op die manier weten partners op basis van een AdministrationId welke AccountService of ProfileService zij kunnen aanroepen voor authenticatie of profielgegevens respectievelijk. Merk op dat ook een intermediaire ketenpartner op deze wijze services kan aanbieden.

2.5 Service Authenticatie en Autorisatie

Principe (1)	**Indien een Rechtspersoon voor gebruik van een service geauthenticeerd moet worden, geschiedt dit op basis van tweezijdige SSLv3/TLS authenticatie met behulp van X.509 certificaten**
Uitleg	Bij het tot stand komen van een SSL-verbinding om een geauthenticeerde Web Service te benaderen, wordt zowel een Server-certificaat als een Client-certificaat gebruikt.
Rationale	Tweezijdige TSL/SSLv3 zorgt voor beveiliging en ondersteunt authenticatie van de communicerende Services. Er is geen publieke standaardvoorziening die voor een organisatie kan valideren of zij een ECK PartyId mag voeren, waardoor deze vertrouwensfunctie niet met behulp van een bestaande PKI opgelost kan worden.

Principe (2)	**Het SSL Server-certificaat van een Service identificeert de hostnaam van de Service**
Uitleg	De server wordt geauthenticeerd op basis van de hostnaam van de server die in het Common Name attribuut van het Server-certificaat is opgenomen.
Rationale	Bij het gebruiken van een SSL Server Certificaat wordt gebruik gemaakt van de publieke PKI infrastructuur. Het voordeel hiervan is dat uitgifte beheerd wordt door een publieke Certificate Authority (CA). Er is geen publieke CA die voor een organisatie kan valideren of zij een ECK PartyId mag voeren, waardoor deze vertrouwensfunctie niet met behulp van een bestaande PKI opgelost kan worden. Er is geen intentie om voor ECK een eigen CA infrastructuur te beheren.

Principe (3)	**Het Client-certificaat identificeert de PartyId van de Service gebruiker**
Uitleg	De Client wordt geauthenticeerd op basis van de PartyId die in het Common Name attribuut van het Client-certificaat is opgenomen.
Rationale	Het valideren of een certificaat, en daarmee de identiteit zoals opgenomen in het CN-attribuut, vertrouwd kan worden, is niet mogelijk met een publiek beschikbare PKI. Het specificeren hoe dit vertrouwen geïmplementeerd kan worden, valt buiten de context van de ECK standaard. Er worden een aantal Best Practices beschreven om dit op te lossen.

Principe (4)	**Een SSL Server Certificaat en een Client Certificaat zijn gemaakt op basis van een RSA sleutel van tenminste 2048 bits en met toepassing van het SHA-256 algoritme voor het digest**
Uitleg	Het implementeren van SSL, gebeurt met certificaten gebaseerd op 2048 bits RSA sleutels en SHA-256 digest algoritmes.
Rationale	Een RSA sleutellengte van 2048 bits en een SHA-256 algoritme biedt voldoende veiligheid voor het opzetten van een SSL/TLS verbinding en voor authenticatie.

Principe (5)	**Rechtspersonen maken bilateraal afspraken over de gegevens die zij onderling uitwisselen**
Uitleg	Partijen die gegevens afnemen worden geauthenticeerd, waarna autorisatie plaatsvindt. Partijen maken vooraf afspraken over welke gegevens geleverd mogen worden. Bij deze afspraken behoort ook het overleggen van de benodigde endpoints die bij betreffende services horen, de PartyId waartoe deze behoren en het uitwisselen van publieke sleutels (certificaten voor TLS/SSL communicatie). Koppelingen kunnen niet dynamisch tot stand komen.
Rationale	Afspraken over welke gegevens uitgewisseld mogen worden tussen partijen, is onderdeel van een overeenkomst tussen twee partijen, en geen onderdeel van de standaard. De standaard wil hier op aansluiten en niet ingrijpen op dit principe, maar wil wel op dit principe kunnen vertrouwen.

Toelichting bij de beveiliging tussen Ketenpartners
De beschreven principes met betrekking tot beveiliging en authenticatie op transport- en applicatieniveau, worden in de onderstaande figuur weergegeven.

Uit de figuur volgt dat de client de server op hostnaam authenticeert, en dat de server de client op PartyId authenticeert. Op applicatie-niveau kan gebruik gemaakt worden van de identiteit zoals deze is vastgesteld op transport-niveau. Als zodanig heeft een Service de beschikking over de PartyId die op transport-niveau is vastgesteld, en kan hier autorisatie op baseren.

Best practice: Client authenticatie

Wanneer authenticatie op basis van Client certificaten plaatsvindt, zijn er verschillende manieren om te bepalen hoe een Certificaat leidt tot een identiteit.
Het voornaamste verschil is of het certificaat zelf bij de server bekend moet zijn, of dat bekend moet zijn welke vertrouwde partijen de attributen hebben gecontroleerd die onderdeel uitmaken van een certificaat.

Certificaten zelf beheren
In het eerste geval is een eigen certificaat administratie vereist. Nadat vastgesteld is dat de Client beschikt over het geheime deel dat bij het certificaat hoort, wordt het certificaat opgezocht in een lokale tabel. In deze tabel wordt elke certificaat aan een identiteit gekoppeld. Op basis van het geauthenticeerde certificaat kan nu de identiteit bepaald worden uit de waarde die in de tabel is opgeslagen.

Certificaten vertrouwen op basis van Certificate Authories (Third Party)
Het tweede geval is gebaseerd op een Chain of Trust. In een lokale tabel worden de certificaten van vertrouwde Certificate Authories (CAs) bijgehouden.
Wanneer vastgesteld is dat de Client beschikt over het geheime deel dat bij het certificaat hoort, wordt gevalideerd of een vertrouwde CA het Client-certificaat heeft goedgekeurd (de CA heeft een digitale handtekening bij het Client-certificaat geplaatst). Indien dit klopt, dan is het mogelijk om de attributen van een certificaat als betrouwbaar te beschouwen. Hieruit volgt dat de identiteit van de Client bepaald kan worden uit het Common Name attribuut van het certificaat.

In de praktijk is het mogelijk om ofwel zelf de functie van Certificate Authority uit te voeren, ofwel deze functie aan een derde partij over te geven. Er zijn standaard tools beschikbaar om een Certificate Authority functie uit te voeren; deze zijn vaak gebaseerd op OpenSSL.

2.6 Foutafhandeling

Principe (1)	**De Web Services retourneren een standaard foutmelding bericht. Dit bericht bevat een foutomschrijving en een foutcode. De foutcodes geven een indicatie of het een bericht succesvol is verwerkt (code = 0), of er een fout is opgetreden (code < 0), of dat het bericht verwerkt is maar dat er een waarschuwing van toepassing is (code > 0).**
Uitleg	Alle Web Services hanteren eenduidige foutmeldingen waar mogelijk. Deze algemene foutcodes zijn in de bijlage van dit document gedefinieerd.
Rationale	Het toepassen van dezelfde generieke foutmeldingen bevordert het kunnen oplossen van problemen en het creëren van een eenduidigheid binnen de hele keten.

3. Implementatie Web Services specificatie

Dit hoofdstuk beschrijft de standaarden die worden gebruikt, hoe deze worden toegepast en aan welke eisen moet worden voldaan.

3.1 Web service standaarden

Voor de web service specificaties en implementaties worden de volgende standaarden toegepast:

Standaard	Beschrijving
XML Schema V1.0	Alle datamodellen worden op basis van een XML Schema (XSD) opgesteld.
SOAP V1.1	Web service berichten worden op basis van de SOAP standaard opgesteld.
WSDL V1.1	De web service interface wordt op basis van de WSDL standaard opgesteld.
HTTP V1.1	Als transportprotocol voor de SOAP berichten wordt het Hypertext Transfer Protocol (HTTP) toegepast
TLS 1.0	Voor beveiliging van het transportkanaal wordt HTTP over TLS Transport Layer Security (RFC2818 en RFC2246) toegepast
X.509 PKI	Internet X.509 Public Key Infrastructure Certificate and CRL Profile (RFC 3280) wordt toegepast bij beveiliging van het transportkanaal. Dit zijn de certificaten met asynchrone sleutels die de basis voor de versleuteling ondersteunen.
SHA256	Als versleuteling algoritme wordt het Secure Hash Algoritme toegepast met een minimale sleutel van 256

3.2 Operaties

Er zijn geen algemene operaties die voor alle web services gelden.

3.3 Berichten

Er zijn geen berichten die bij meerdere operaties en functiehuizen gebruikt worden. Wel zijn er bepaalde elementen die vaak binnen berichten gebruikt worden. Deze zijn bij de Types opgenomen.

3.4 Types

Een aantal types worden bij meerdere web services en operaties gebruikt, zoals bijvoorbeeld een bevestiging. Deze algemene types worden hieronder weergegeven. Bij de specifieke Web Service beschrijvingen wordt mogelijk naar deze types verwezen. De implementatie-details (cardinaliteit, domein-specificatie, etc.) van elk type wordt gespecificeerd in het XSD-bestand CommonTypesSchema.xsd. Dit bestand is als bijlage bij dit document opgenomen.
Verdere specificaties van Identifier-types worden in het daartoe bestemde hoofdstuk beschreven.

Naam	Beschrijving
ConfirmationType	Element dat gebruikt wordt om aan te geven dat een verwerking succesvol is uitgevoerd
ContentIdType	Identificatie van een Leermiddel binnen een Administratie
CurrencyType	Geeft valuta aan
AmountType	Een bedrag, zoals bijvoorbeeld gebruikt bij betalingen
DeliveryMethodType	Levermethode van het digitaal materiaal
CodeType	Basisvorm van een code
AvailabilityStatusType	Type geeft opties aan mbt de beschikbaarheid van een leermiddel
IdentifierType	Een identifier van een gebruiker, gegeven, unit of andere identificeerbare eenheid. Deze kan gecombineerd worden met een waarde voor een AdministrationId om een Identifier ketenbreed uniek te maken.
EductionalContentIdType	Basis voor Leermiddel Identifier.
AdministrationIdType	Het type voor een identificerend gegeven van een administratie
PartyIdType	Het type voor een identificerend gegeven van een ketenpartner (rechtspersoon)
UnitIdentiferType	Identificatie van een Unit binnen een Administratie
CreditorIdentiferType	Identificatie van een Creditor binnen een Administratie
ContentIdentifierType	Identificatie van een Leermiddel binnen een Administratie
FaultMessageType	Element dat gebruikt wordt om een opgetreden fout bij het uitvoeren van een opdracht te beschrijven in combinatie met een beschrijving van de opgetreden fout.
RoleType	Rol van een gebruiker (Student, Teacher, Administrator)
an0_10	Basistype voor een string met een lengte tot 10 karakters

Merk op: de rijen met grijze achtergrond zijn samengestelde types (bijvoorbeeld een IdentifierType gecombineerd met een AdministrationId).

3.5 Gegevensuitwisselingspatroon

De gegevensuitwisseling is synchroon: een vraagbericht wordt in een dezelfde synchrone HTTP request/response sessie verzonden, als waarin het resultaatbericht ontvangen wordt. Het is niet altijd zo dat een inhoudelijke melding gewenst is. Zo kan men er voor kiezen om bij het succesvol uitvoeren van een vraagbericht op http niveau e.e.a. af te handelen. De afspraak voorziet momenteel echter nog in de ondersteuning van een optionele bevestiging waarin een code en een omschrijving opgenomen kan worden.

3.6 Namespaces

De gebruikte namespaces worden samengesteld op basis van de domeinnaam waaronder een Service of Type-specificatie operationeel is, een verwijzing naar de betreffende Service of Type-specificatie, en de versie van de specificatie.

Bijvoorbeeld: de namespace van de WSDL-specificatie van de AccountService versie 1.4, is "http://accountservice.dtdl.eck2.nl/service/accountservice/v1.4" . De gebruikte Type-definities zijn in de XSD-specificatie van de AccountService versie 1.4 te vinden in namespace "*http://accountservice.dtdl.eck2.nl/schema/v1.4".*

3.7 WSDL

Voor elke Service wordt in principe één WSDL-specificatie opgesteld welke alle operaties van die Service beschrijft. Hiermee worden alle operaties van een Service via hetzelfde URL endpoint aangeboden. Elke Service bevat hierdoor dus dezelfde voorwaarden voor wat betreft het gebruikte transportkanaal.

3.7.1 Binding

De WSDL binding beschrijft hoe een Service aan een Messaging-protocol gekoppeld is. Voor de SOAP berichten wordt SOAP 1.1 en "document-literal binding" gehanteerd. Bij document–literal mag het SOAP "body" element slechts één XML element bevatten. Hierbinnen kunnen eventueel wel meerdere elementen opgenomen worden. Verder wordt aanbevolen de richtlijnen van het WS-I Basic Profile 1.2 op te volgen.

3.7.2 XSD Elements

Elk WSDL document verwijst naar een XSD document, waarin de types die specifiek zijn voor de betreffende Service worden gedefinieerd.
De namen van elementen in de XSD's zijn conform de binding van het gegeven in de DTDL gegevensbeschrijving. Toolkits gaan verschillend om met het verwerken van kleine en hoofdletters. Om problemen hiermee te beperken worden de onderdelen van een WSDL op een standaard manier beschreven. Bij het opstellen van een WSDL worden de elementen en typen op basis van Camel Case genoteerd, bijvoorbeeld "UserName".

3.7.3 WSDL Service en operatie

De naam van een WSDL-Service is bepaald door de naam van een DTDL-Service, gecombineerd met een versie. De functies die in een DTDL-Service worden gespecificeerd, worden als Web Service operaties gedefinieerd.
De waarde voor het soapAction attribuut van een WSDL-operatie wordt bepaald door de waarde van de Target NameSpace aan te vullen met de naam van de operatie.

Bijvoorbeeld: de DTDL-Service "PaymentService" volgens versie 1.6, wordt als WSDL-Service "PaymentService-v1.6" gedefinieerd. Hierbinnen is de functie "*RegisterPayment*" als WSDL-Operation gedefinieerd. Voor deze operation is de waarde van het soapAction-attribuut "*http://paymentservice.dtdl.eck.nl/service/paymentservice/v1.6/RegisterPayment*".

3.7.4 WSDL messages

De WSDL-messages die door WSDL-operaties worden gespecificeerd, krijgen een naam die verwijst naar de operatie (bv RegisterPayment) en het berichttype (Request of Response).

Bijvoorbeeld: het bericht dat als input-bericht dient voor de RegisterPayment operatie, heeft als naam "*RegisterPaymentRequest*".

3.7.5 WSDL namespace prefix

De XML-namespace prefixes worden volgens de Notational Conventions van de WSDL 1.1 standaard toegepast, zie tabel 1.

Prefix	Namespace	Definitie
wsdl	http://schemas.xmlsoap.org/wsdl/	WSDL namespace for WSDL framework.
soap	http://schemas.xmlsoap.org/wsdl/soap/	WSDL namespace for WSDL SOAP binding.
http	http://schemas.xmlsoap.org/wsdl/http/	WSDL namespace for WSDL HTTP GET & POST binding.
mime	http://schemas.xmlsoap.org/wsdl/mime/	WSDL namespace for WSDL MIME binding.
soapenv	http://schemas.xmlsoap.org/soap/envelope/	Envelope namespace as defined by SOAP 1.1
xsi	http://www.w3.org/2000/10/XMLSchema-instance	Instance namespace as defined by XSD [10].
xsd	http://www.w3.org/2000/10/XMLSchema	Schema namespace as defined by XSD [10].
tns	http://servicenaam.domein/services/service_implementatie_naam	The "this namespace" (tns) prefix is used as a convention to refer to the current document.

Table 1WSDL namespace prefixen

3.8 Versie-aanduiding

Versie-aanduiding op technische documenten, zoals WSDL's en XSD's hebben gevolgen voor het uitrollen van de web service in de praktijk. Er zijn een aantal elementen in een WSDL waaraan een versie aanduiding moet worden toegevoegd. Dit zijn:

- WSDL/namespace
- WSDL/Servicenaam
- WSDL/PortType
- WSDL/Type(s) (XSD) namespace

Een wijzigingen in de naam van de namespace zal er toe leiden dat de XSD of WSDL niet compatible is met een andere versie, immers de twee verschillende releases hebben geen gemeenschappelijke namespace meer. Er zijn verschillende mogelijkheden om in code hier mee om te gaan en partijen zijn vrij om hierin een eigen keuze te maken. Voor de interface geldt het uitgangspunt dat een nieuwe versie ook een nieuwe service betekent. De afnemers hoeven niet direct over te gaan naar de nieuwe interface indien de dienstaanbieder de oude versie nog beschikbaar blijft houden. Men zou zo bijvoorbeeld twee versies in productie kunnen houden.

Voorbeelden bij versie-aanduidingen, zijn:

- WSDL namespace "http://profileservice.domein/services/profileservice/v1.4"
- XSD namspace "http://profileservice.domein/schema/v1.4".
- Servicenaam "Profileservice-v1.3"
- PortType "IProfileservice-v1.3"

4. SAML2 specificatie

Voor het authenticeren van gebruikers wordt een identiteitenleverancier gebruikt. Dit houdt in dat de dienst die een identiteit nodig heeft, aan een externe partij vraagt om deze identiteit vast te stellen. Wanneer de identiteit van de gebruiker geauthenticeerd is, wordt er een verklaring van deze identiteit afgegeven aan de aanvragende dienst. Het protocol dat door ECK-DTDL gebruikt wordt voor deze vorm van authenticatie, is het SAML2 protocol. De dienst die de identiteit levert, is (het SAML2-deel van) de AccountService.

In dit hoofdstuk worden allereerst de principes toegelicht die ten grondslag liggen aan het gebruik van SAML2 voor DTDL. Daarna wordt een toelichting gegeven hoe het protocol uitgevoerd wordt en worden de implementatiedetails van de SAML2 specificatie beschreven.

4.1 Basisprincipes

Principe (1)	**Voor gebruikers-authenticatie wordt het SAML Web Browser SSO profiel toegepast: SP initiated SSO with POST/artifact bindings**
Uitleg	De SAML2 specificatie beschrijft een aantal profielen. Het Web Browser SSO profiel, uitgevoerd middels SP initiated SSO with POST/artifact bindings, zorgt ervoor dat de daadwerkelijke SAML2-berichten niet via browser redirects worden uitgewisseld, maar dat er door de browser van de gebruiker alleen artifacts worden uitgewisseld. Deze artifacts kunnen door SP en IDP rechtstreeks via een back-channel worden gebruikt om de specifieke berichten op te halen. Hierdoor beschikt de (browser van) de gebruiker nooit over de gegevens die onderdeel uitmaken van de SAML2 AuthnRequest en Response berichten.
Rationale	Door gegevens niet via de browser van de gebruiker te communiceren maar via een service-to-service backchannel, worden de uitgewisselde gegevens niet aan de gebruiker blootgesteld, en kan beveiliging van transport van deze gegevens gecentraliseerd worden uitgevoerd. Deze veiligheid wordt hier verkozen boven het nadeel dat back-channelcommunicatie ten koste gaat van de transparantie van de uitgewisselde gegevens.

Principe (2)	**Profielinformatie van een gebruiker wordt via de ProfileService ontsloten, en niet via de AccountService**
Uitleg	Een authenticatie door de AccountService geeft als resultaat alleen de identifier van de geautheniceerde gebruiker, eventueel aangevuld met een verwijzing naar de ProfileService die kan worden gebruikt om profielinformatie van de gebruiker op te vragen.
Rationale	De authenticatiefunctie leidt tot het minimale antwoord wat er is, namelijk slechts de geauthenticeerde identiteit. Het verrijken van het account met profielgegevens van de gebruiker is losgekoppeld van het vaststellen van de identiteit.

Principe (3)	**De gegevensuitwisseling om de gebruiker te authenticeren is beveiligd. Het transportkanaal is versleuteld op basis van SSLv3/TLS.**
Uitleg	Een gebruiker authenticeert zich via een AccountService. Door gebruik te maken van een HTTPS-verbinding, worden de gegevens die voor authenticatie van de gebruiker nodig zijn over een versleuteld kanaal naar de AccountService gestuurd. Een eventuele sessiecookie die wordt gezet om een sessie tussen een Gebruiker en een AccountService te gebruiken, wordt alleen via een beveiligd transportkanaal uitgewisseld.
Rationale	De gegevens worden over internet verstuurd. Door deze te beveiligen is de gegevensoverdracht versleuteld, en kan de gebruiker de server authenticeren. Een sessiecookie tussen AccountService en Gebruiker kan leiden tot het authenticeren van een identiteit zonder dat daarvoor een username/password controle wordt uitgevoerd. Deze sessiecookie wordt beschermd door het gebruiken van een SSLv3/TLS-verbinding.

Principe (4)	**De SAML2 Artifact Resolution endpoints moeten met SSL worden beveiligd**
Uitleg	Bij het herleiden van een Artifact, wordt een SSL-verbinding gebruikt om te communiceren met de endpoints van de AccountService en de dienstaanbieder. Met behulp van het X.509 SSL Server-certificaat, wordt de hostname van de AccountService geauthenticeerd door de aanroepende partij.
Rationale	De gegevens worden over internet verstuurd. Met behulp van een SSL-verbinding kan de hostname van de server worden gevalideerd en zijn de gegevens door middel van versleuteling beschermd tegen afluisteren. Authenticatie van de Client wordt voorzien op applicatie (SAML2) niveau, en is hier dus geen vereiste (zie ander principe). De versleuteling op SSL-niveau garandeert geen end-to-end beveiligd transportkanaal (bijv. bij gebruik van SSL-offloaders en loadbalancing configuraties), echter wordt aangenomen dat de SSL-verbinding getermineerd wordt binnen het domein van de dienst; de eventuele verdere gegevensoverdracht van bijvoorbeeld een loadbalancer tot aan de server die het verzoek daadwerkelijk afhandelt, valt buiten de verantwoording van de DTDL technische specificatie.

Principe (5)	**SAML2 fouten worden conform de SAML2 specificatie gerapporteerd**
Uitleg	Wanneer er een fout is opgetreden bij het beantwoorden van een authenticatieverzoek, wordt hierover gerapporteerd in het Status-element van een SAML2 Response-bericht.
Rationale	De foutafhandeling van een authenticatie is zodanig generiek, dat hiervoor geen specifieke DTDL foutspecificatie hoeft te worden opgesteld. SAML2 biedt de mogelijkheid om een fout tekstueel toe te lichten als onderdeel van een Status-element. Dit biedt voldoende ruimte voor een AccountService om, indien gewenst, aanvullende informatie te rapporteren rondom een foutsituatie. Zie SAML2 Specificatie: saml-core-2.0-os:3.2.2.2

Principe (6)	**Alle uitgewisselde SAML-berichten moeten worden gelogd.**
Uitleg	Elk SAML-bericht dat wordt ontvangen of wordt verstuurd, wordt in een applicatie-log opgeslagen. Deze logfile moet gebruikt kunnen worden om achteraf de gegevens rondom een specifieke authenticatie-transactie te kunnen herleiden. De logfile is alleen beschikbaar voor de daartoe geautoriseerde personen.
Rationale	Er wordt logging uitgevoerd zodat de context van foutsituaties kan worden herleid, maar ook om achteraf te kunnen bepalen op welke wijze een SAML2-transactie zich heeft afgespeeld.

4.2 SAML profiel en signing

Principe (1)	**De SAML2 berichten die tussen dienstaanbieder en AccountService worden uitgewisseld, moeten door de versturende partij worden ondertekend met een SAML2 Signing certificaat**
Uitleg	Alle SAML2 berichten worden op berichtniveau ondertekend. Dit houdt in dat de AuthnRequest, Response, ArtifactResolve en ArtifactResponse berichten voorzien zijn van een ds:Signature en ds:Digest dat wordt bepaald aan de hand van het SAML2 Signing certificaat
Rationale	De berichtuitwisseling tussen dienstaanbieder en AccountService mag alleen maar door geautoriseerde partijen worden uitgevoerd. Om dit te controleren, worden alle berichten voorzien van een digitale handtekening waarmee de ontvanger de authenticiteit en integriteit van het bericht kan controleren.

Principe (2)	**Voor het bepalen van de SAML2 EntityId van een AccountService wordt een richtlijn conform URN-notatie voorgesteld**
Uitleg	Bij voorkeur wordt een consistente URN-notatie gebruikt om een SAML2 entiteit (Dienstverlener of AccountService) te identificeren. Deze is dan opgebouwd als volgt: "urn:eck:partyId:acctsvc:someId", bijvoorbeeld: urn:eck:kennisnet:acctsvc:federatiehub Zie URN specificatie: IETF 2141
Rationale	Een standaardformaat maakt het makkelijk om extra informatie uit een identifier af te leiden, maar dit is niet noodzakelijk. Wellicht ondersteunen niet alle SAML2 implementaties een verplicht voorgeschreven formaat.

Principe (3)	**Een SAML2 Assertion wordt voorzien van een digitale handtekening**
Uitleg	De Assertion die de geauthenticeerde gebruiker bevat, wordt altijd ondertekend door de AccountService die deze assertion uitgeeft. De digitale handtekening houdt in dat de Assertion een ds:Signature en een ds:Digest bevat. Voor het bepalen van de handtekening wordt het SAML2 Signing Certificaat van de uitgevende AccountService gebruikt.
Rationale	Door het ontkoppelen van de Assertion van het ArtifactResponse of Response bericht, kan ten alle tijden bepaald worden van wie de Assertion afkomstig is, bijvoorbeeld bij offline verificatie of bij het gebruik van een SAML2 proxy.

Principe (4)	**Het gebruikte zekerheidsniveau voor authenticatie is PasswordProtectedTransport**
Uitleg	In een AuthnRequest bericht mag een voorwaarde gesteld worden aan de wijze waarop authenticatie plaatsvindt. Binnen DTDL wordt gebruikersnaam en –wachtwoord als authenticatiemethode gespecificeerd. Wanneer gebruik gemaakt wordt van de (optionele) RequestedAuthnContext faciliteit van SAML2 in het AuthnRequest bericht, moet tenminste "urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport" als AuthnContext worden toegestaan. Zie SAML2 Specificatie: saml-core-2.0-os:3.3.2.2.1
Rationale	De DTDL specificatie is gebaseerd op gebruikersnaam en –wachtwoord authenticatie, en deze methode moet altijd geaccepteerd worden voor het beantwoorden van een authenticatieverzoek.

Principe (5)	**Het gebruik van SubjectLocality als onderdeel van de authenticatieverklaring is optioneel**
Uitleg	De AccountService kan optioneel een verklaring afgeven over het IP-adres waarvandaan de Gebruiker is geauthenticeerd, door de SubjectLocality faciliteit van een SAML2 Authentication Context in een Assertion te gebruiken. Indien dit gevuld is, is het aan de Dienstverlener om dit te gebruiken om te bepalen de Gebruiker toegelaten wordt. Zie SAML2 Specificatie: saml-core-2.0-os:2.7.2.1
Rationale	Het vergelijken van het IP-adres dat gebruikt wordt voor authenticatie bij een AccountService en voor communicatie met de Dienstverlener kan mogelijkerwijs sessie-kapingen detecteren.

Principe (6)	**Een Subject in een SAML2 Assertion wordt middels het bearer principe overgedragen van een AccountService aan een Dienstverlener**
Uitleg	In de verklaring van de identiteit van een Gebruiker die door de AccountService wordt verstrekt, wordt altijd aangegeven dat de drager van de Assertion ook degene is die in de Assertion is genoemd, het zogenaamde bearer principe. Dit wordt (verplicht) aangegeven in het SubjectConfirmation/Method element van een Assertion, door hiervoor de waarde "urn:oasis:names:tc:SAML:2.0:cm:bearer" te gebruiken. Zie SAML2 Specificatie: saml-core-2.0-os:2.4.1.1
Rationale	Door deze verklaring op te nemen, wordt in de Assertion eenduidig opgenomen hoe deze door degene die de Assertion ontvangt (Dienstverlener) dient te worden geinterpreteerd.

Principe (7)	**De NameID van een geauthenticeerd Subject moet conform het "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified" formaat worden uitgegeven**
Uitleg	Het NameID formaat unspecified wordt toegepast. Indien er geen NameID Format wordt opgenomen in een verzoek, wordt dit impliciet aangenomen. Indien een ander NameID Format wordt gevraagd in een AuthnRequest, dan wordt het verzoek afgewezen. Zie SAML2 Specificatie: saml-core-2.0-os:8.3
Rationale	Er is geen standaardiseerd formaat dat de specifieke condities specificeert van een Gebruikers-identifier zoals deze in ECK wordt gebruikt.

4.3 Vertrouwensrelaties en SAML2 metadata

Principe (1)	**Elke AccountService of Dienstverlener ondersteunt het verwerken van SAML2 Metadata**
Uitleg	Wanneer een partij metadata beschikbaar stelt aan een andere partij, dan kan deze verwerkt worden.
Rationale	De SAML2 specificatie voorziet in een metadata standaard, waarmee alle protocolspecifieke kenmerken van een SAML2 entiteit worden gedefinieerd. Hiermee kunnen alle SAML2 configuraties met elkaar worden uitgewisseld, en is er geen noodzaak meer voor andere manieren om een SAML2 partij te beschrijven.

Principe (2)	**De verschillende SAML2 AccountServices en Dienstverleners wisselen SAML2 metadata onderling met elkaar uit**
Uitleg	De SAML2 metadata wordt rechtstreeks tussen partijen uitgewisseld die met elkaar willen communiceren. Dit is een out-of-band proces (buiten het authenticatieprotocol) waarmee onder meer de gebruikte endpoints, maar ook de gebruikte SAML2 Signing certificaten worden uitgewisseld. De SAML2 certificaten worden gebruikt voor het controleren van de digitale handtekeningen in de SAML2 protocol-berichten en Assertions. Uitwisseling kan op twee manieren geschieden, ofwel doordat een partij een URL ter beschikking stelt waar zijn eigen metadata opgehaald kan worden, ofwel door het uitwisselen van een bestand, dat de metadata bevat, met de partij die de metadata wil gebruiken. Indien de metadata via een URL beschikbaar wordt gesteld, dient dit te gebeuren door een met SSL beveiligde verbinding.
Rationale	Door het bilateraal uitwisselen van metadata, is geen centrale registratie van deelnemende partijen vereist. Elke deelnemer bepaalt en beheert zelfstandig met welke partijen hij wil communiceren en welke digitale handtekeningen hij wil accepteren.

Principe (3)	**Een SAML2 Signing Certificaat is op basis van een RSA sleutel van tenminste 2048 bits en met toepassing van het SHA-256 algoritme voor het digest**
Uitleg	Een certificaat dat gebruikt kan worden om een digitale handtekening te controleren, kan door elke partij zelf worden gegenereerd. Het vertrouwen van een certificaat gebeurt door middel van het vertrouwen van de metadata zoals ontvangen van de andere partij. Er is dus geen chain of trust van toepassing op het certificaat voor het controleren van de digitale handtekeningen die voor SAML2 berichten worden gebruikt.
Rationale	Een RSA sleutellengte van 2048 bits en een SHA-256 algoritme biedt voldoende veiligheid voor het bepalen van authenticiteit en integriteit van een digitiale handtekening van een bericht.

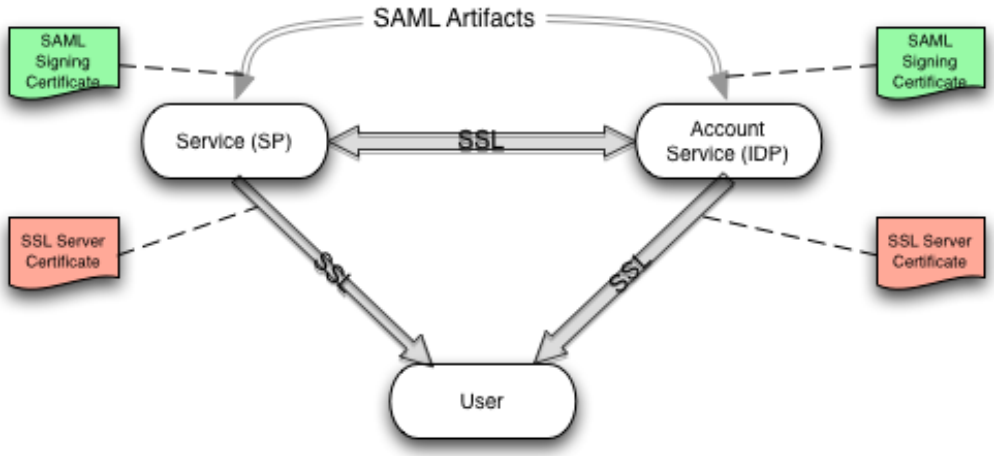


Figure 1 Overzicht van het gebruik van verschillende certificaten bij authenticatie

4.4 Authenticatieproces

De bovenstaande principes worden gehanteerd bij het uitvoeren van het proces om een authenticatieverzoek af te handelen. De volgende rolverdeling is hierop van toepassing:

- Dienstaanbieder: SAML Service Provider (SP), gebruiker van (authenticatie)verklaringen.
- AccountService: SAML Identity Provider (IdP), deze levert de verklaringen.
- Gebruiker: de natuurlijk persoon en daarmee de identiteit die met de verklaringen geassocieerd wordt.

De Dienstaanbieder is de partij die een dienst beschikbaar stelt voor de gebruiker, terwijl de AccountService verantwoordelijk is voor de verklaringen over de gebruiker. De authenticatievraag begint bij een Gebruiker die een dienst wil afnemen waarvoor een identiteit is vereist door de dienst. Hierop zal de Dienstaanbieder de Gebruiker doorverwijzen naar de AccountService met de vraag of hij deze kan Gebruiker kan authenticeren (SAML2 AuthnRequest).

Authenticatie van de Gebruiker vindt plaats tussen te Gebruiker en de AccountService, en gebeurt door middel van het controleren van een gebruikersnaam en –wachtwoord. Indien de identiteit van de gebruiker is vastgesteld, wordt er door de AccountService een verklaring gemaakt over de identiteit en de uitgevoerde authenticatieprocedure (SAML2 Assertion). De verklaring wordt digitaal ondertekend door de AccountService die de gebruiker heeft geauthenticeerd. De Gebruiker zorgt ervoor dat de Dienstaanbieder deze verklaring krijgt, en zo beschikt over de identiteit van de Gebruiker aan wie de dienst nu verleend kan worden.

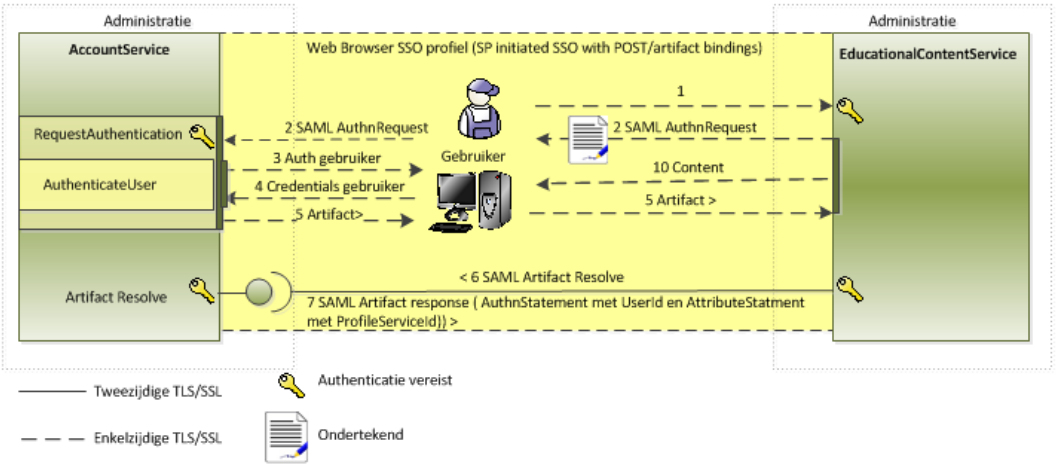
SAML2 Koppelvlakken:

Het authenticeren van gebruikers en het uitwisselen van SAML2 protocol-berichten wordt op basis van het SAML2 Web Browser SSO profiel (SP initiated SSO with POST/artifact bindings) geïmplementeerd. Dit houdt in de Gebruiker zogenaamde artifacts doorgeeft van Dienstaanbieder naar AccountService (en vice versa), en dat de Dienstaanbieder respectievelijk de AccountService deze artifacts kunnen gebruiken om rechtstreeks bij de andere partij het SAML2 protocolbericht te achterhalen.

Het proces wordt hieronder stap voor stap toegelicht.

4.4.1 *Authenticatiefunctie op basis van het SAML Web Browser SSO profiel*

In Figuur 2 wordt het SAML2 Web Browser SSO Profiel (SP initiated SSO with Redirect and POST/artifact bindings) weergegeven met de relevante functies.



Figuur 2 Web Browser SSO profiel (SP initiated SSO with Redirect and POST/artifact bindings)

1) Gebruiker wil dienst afnemen waarvoor identificatie is vereist

De dienst stelt vast dat er een authenticatie plaats dient te vinden, en bepaalt welke AccountService dit moet doen. Het is mogelijk dat meerdere AccountServices de authenticatiefunctie kunnen uitvoeren.
Wanneer de gebruiker wordt gevraagd om een keuze te maken bij welke AccountService hij of zij kan authenticeren, kan de dienst deze keuze voor de gebruiker onthouden.

2) Versturen van <AuthnRequest> door Dienstaanbieder

Op basis van de in stap 1 bepaalde AccountService, wordt in de lokale administratie het endpoint van die AccountService bepaald waar een AuthnRequest bericht naartoe gestuurd kan worden. De Dienstaanbieder maakt een AuthnRequest bericht, ondertekent het en verstuurt dit via een Redirect naar de AccountService.

3 en 4) AccountService authenticeert de gebuiker

De Gebruiker authenticeert zich bij de AccountService. Indien er al een sessie tussen de AccountService en de Gebruiker bestond, zou de sessie hervat kunnen worden zonder dat de Gebruiker expliciet hoeft in te loggen (Single Sign On of SSO). De Dienstaanbieder kan overigens expliciete authenticatie afdwingen in het AuthnRequest-bericht.
Een AccountService gebruikt een gebruikersnaam en –wachtwoord om de Gebruiker te authenticeren, en daarmee de UserId en UserAdministrationId te bepalen van de Gebruiker.
Wanneer authenticatie succesvol was, wordt er een Assertion aangemaakt, die een verklaring geeft over de geauthenticeerde identiteit en de context van de authenticatie (zoals bijvoorbeeld SubjectLocality, het IP-adres waarmee de Gebruiker met de AccountService heeft gecommuniceerd)

5) AccountService verstuurt het Artifact

Nadat authenticatie van de Gebruiker is uitgevoerd (succesvol of niet) creëert de AccountService een Artifact als verwijzing naar het antwoord op het AuthnRequest bericht. De AccountService levert de Artifact af op de Assertion Consumer URL van de Dienstaanbieder. Deze URL wordt bepaald aan de hand van de SAML2 Metadata van de Dienstaanbieder, bij voorkeur doordat in het

AuthnRequest uit stap 1, een AssertionConsumerServiceIndex is meegegeven, die kan worden opgezocht in de Metadata van de Dienstaanbieder.

Het Artifact wordt door middel van een POST-verzoek van de AccountService, via de (browser van) de gebruiker, aan de Dienstverlener doorgegeven. Onderdeel van het Artifact is een (gecodeerde) verwijzing naar de AccountService, zodat deze door Dienstverlener bepaald kan worden om het gekoppelde bericht te kunnen ophalen.

6) De dienstaanbieder haalt op basis van het Artifact het antwoordbericht bij de AccountService op
De dienstverlener bepaalt eerst de AccountService waar de Artifact ingeleverd kan worden voor een Response-bericht als antwoord op het oorspronkelijke AuthnRequest-bericht uit stap 1. Vervolgens wordt de aan te roepen URL bepaald uit de SAML2 Metadata van de AccountService. Middels een ArtifactResolve bericht aan de AccountService en ArtifactResponse bericht aan de Dienstverlener, wordt het Response-bericht bepaald.
Wanneer authenticatie niet succesvol was, wordt er een fout gerapporteerd in het Response-bericht, en wordt er geen Assertion teruggegeven. Wanneer authenticatie geslaagd was, wordt door de Dienstverlener de Assertion bepaald uit het ontvangen Response-bericht.
Indien in stap 1 een RelayState parameter was meegegeven, wordt deze waarde zoals deze in de AuthnRequest uit stap 1 is meegegeven als parameter meegegeven naast de SAMLart parameter.

7) Dienstaanbieder autoriseert de Gebruiker
De Dienstaanbieder kan het Response-bericht interpreteren, en indien authenticatie geslaagd was, een Assertion bepalen uit het Response-bericht. Van deze Assertion wordt de authenticiteit en integriteit bepaald aan de hand van de digitale handtekening die hierop geplaatst is door de AccountService.
In de Assertion wordt informatie verschaft over de context waarin authenticatie heeft plaatsgevonden, zoals bijvoorbeeld de SubjectLocality. Daarnaast bevat Assertion informatie over de geauthenticeerde identiteit van de gebruiker in de vorm van een 'UserId@UserAdministrationId' identifier. Op basis van de authenticatie context en de identiteit kan de Dienstverlener haar eigen autorisatie toepassen.
De Assertion bevat informatie waarmee de Dienstverlener de ProfileService kan bepalen, die mogelijkerwijs aanvullende informatie bij de geauthenticeerde gebruiker kan leveren. Dit is verder gespecificeerd in de context van de ProfileService.

4.4.2 IDP Initiated authenticatieprofiel

Naast een SP-initiated profiel, is het ook mogelijk dat de IDP het authenticatieproces initieert. In dit geval wordt het authenticatieproces zoals beschreven voor SP initiated SSO vanaf stap 5 uitgevoerd. De specifieke voorwaarden die van toepassing zijn op de berichtuitwisseling, volgen het Unsolicited Response specificatie van SAML[1].

Indien gebruik gemaakt wordt van een architectuur waarin een Dienstverlener de authenticatievraag aan een SAML proxy of hub stelt, wordt verwezen naar de manier waarop de beheerder van de proxy of hub een dergelijke flow uitvoert om de gebruikerservaring van een IDP Initiated profiel te krijgen.

[1] Zie het hoofdstuk **4.1.5 Unsolicited Responses** in saml-profiles-2.0

4.5 Bericht specificaties Authenticatie

4.5.1 AuthnRequest

Het AuthnRequest is de authenticatievraag van de Dienstaanbieder aan een AccountService. Een voorbeeld van een AuthnRequest is opgenomen in de bijlage SAML2 Berichten. Het AuthnRequest bericht wordt ondertekend door de Dienstaanbieder en wordt verstuurt op basis van HTTP Redirect binding met DEFLATE encoding[2].

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht, vulling conform SAML specificatie
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt, vulling conform SAML specificatie
Destination	URL van de AccountService waarnaar het bericht verstuurd wordt. Deze komt overeen met de aangegeven URL in de metadata van de AccountService.
Consent	NIET opnemen
Issuer	Moet de EntityId van de Dienstaanbieder bevatten
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Issuer .SPProviderID	NIET opnemen
Signature	Het hele bericht MOET ondertekend zijn door de Dienstaanbieder met haar SAML2 Signing certificaat.
Extensions	NIET opnemen
Subject	MAG opgenomen worden en bevat dan de User identifier (UserId@UserAdministration) die moet worden geauthenticeerd
NameIDPolicy	NIET opnemen
Conditions	NIET opnemen
RequestedAuthnContext	MAG opgenomen worden en MOET dan de waarde Comparison="minimum" bevatten en een AuthnContextClassRef met PasswordProtectedTransport
Scoping	MAG opgenomen zijn met een SAML <IDPList> met hierin een suggestie met de betreffende AccountServices die authenticatie zouden moeten uitvoeren. Scoping van IDP's voorziet in het doorgeven van de AccountServiceId-waarden die een ContentService kan ontvangen.
ForceAuthn	MAG opgenomen worden. Dienstaanbieder vraagt hiermee om expliciete authenticatie, dus ook wanneer de gebruiker al geldige een sessie met AccountService heeft. Wanneer zowel ForceAuthn als IsPassive de waarde "true" bevatten, mag op de AccountService geen gebruikersinteractie plaatsvinden om het antwoord op het verzoek te kunnen geven, en zal het verzoek dus afgewezen worden conform IsPassive specificatie.
IsPassive	MAG opgenomen zijn met waarde "true". In dat geval kan

[2] Zie SAML 2.0 Bindings paragraaf 3.4.4.1 DEFLATE Encoding ()

	de AccountService niet de user interface (browser) overnemen. Als hier niet aan tegemoet kan worden gekomen (bijvoorbeeld omdat de gebruiker nog geen sessie bij AccountService heeft of Cookie niet gezet is) moet er een antwoordbericht gestuurd worden met een statuscode "urn:oasis:names:tc:SAML:2.0:status:NoPassive".
AssertionConsumerServiceIndex	MAG opgenomen worden, indien in de metadata van de Dienstverlener meer dan 1 AssertionConsumerService gedefinieerd wordt.
AssertionConsumerServiceURL	NIET opnemen
ProtocolBinding	NIET opnemen
AttributeConsumingServiceIndex	MAG opgenomen worden, maar is niet gespecificeerd in DTDL.
ProviderName	MAG naam van dienst bevatten

4.5.2 *Response*

Het Response bericht is het antwoord zoals de AccountService dit geeft op basis van een AuthnRequest vraag. Het wordt via Artifact resolution verstrekt aan de Dienstverlener. Indien authenticatie succesvol was, bevat het Response-bericht een Assertion. In alle andere gevallen bevat het Response-bericht een Responder element met een indicatie van het soort fout dat is opgetreden.
Het Response-bericht wordt NIET voorzien van een digitale handtekening door de AccountService die het bericht uitgeeft, omdat de bron van het Response-bericht reeds door het ArtifactResolution profiel wordt geauthenticeerd.

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht
InResponseTo	Uniek kenmerk van het AuthnRequest waarop dit Response bericht het antwoord is.
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt, vulling conform SAML specificatie
Destination	URL van de Dienstverlener waarop het bericht wordt aangeboden. MOET overeenkomen met de metadata van de Dienstverlener, de ArtifactResolutionService URL.
Consent	NIET opnemen
Issuer	Moet de EntityId van de AccountService bevatten
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Issuer .SPProviderID	NIET opnemen
Signature	NIET opnemen.
Extensions	NIET opnemen
Status	MOET een element StatusCode bevatten
Status.StatusCode	Status van de authenticatie. In geval van succes wordt deze met "urn:oasis:names:tc:SAML:2.0:status:Success" gevuld en wordt een Assertion in het Response-bericht meegegeven. SAML2 specificatie: saml-core-2.0-os:3.2.2.2
Status.StatusMessage	MAG opgenomen worden, en is vrij invulbaar.
Status.StatusDetail	NIET opnemen
Assertion	Bevat een verklaring over de authenticatie met daarin

	een verklaring over de bevoegdheid (zie hieronder)

4.5.3 Assertion

Een Assertion wordt verstrekt als onderdeel van een Response-bericht, maar als verklaring wordt het specifiek beschreven als protocol-bericht.

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt, vulling conform SAML specificatie
Issuer	MOET de EntityId van de AccountService bevatten die de Gebruiker heeft geauthenticeerd. In het geval dat een SAML2 proxy een Assertion verschaft, hoeft dit dus niet dezelfde te zijn als de Issuer van een Response-bericht! Indien het SAML< Format> attribuut opgenomen is, moet dit de waarde <urn:oasis:names:tc:SAML:2.0:nameidformat:entity> bevatten.
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Signature	NIET opnemen
Subject	MOET opgenomen worden en bevat de volgende elementen: 1 SAML <NameID> 2 Eén SAML <SubjectConfirmation> element Het SAML <NameID> MOET gevuld zijn met de identiteit van de gebruiker in het formaat "UserId@UserAdministrationId", conform de specificatie in het hoofdstuk over Identifiers. Het NameID element is van het formaat urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified Het SAML <SubjectConfirmation> element bevat een SAML <Method> met de waarde urn:oasis:names:tc:SAML:2.0:cm:bearer Het SAML <SubjectConfirmation> bevat een SAML <SubjectConfirmationData> element met hierin een Recipient attribuut met de URL van de van SAML/SSO handler van de Dienstaanbieder, een <NotOnOrAfter> attribuut dat de tijd aangeeft waarbinnen de SAML Assertion geleverd moet worden en een <InResponseTo> attribuut met referentie naar het Authnrequest. <saml:Subject> <saml:NameID Format= urn:oasis:names:tc:SAML:1.1:nameid-

	format:unspecified >**UserId**</saml:NameID> <saml:SubjectConfirmation **Method**=”urn:oasis:names:tc:SAML:2.0:cm:bearer”> <saml:SubjectConfirmationData **Recipient**= **DienstAanbieder sso endpoint** **NotOnOrAfter**=”2012-12-31T12:00:00” **InResponseTo**= Authn_request_identifier_1234567 > </saml:SubjectConfirmationData> </saml:SubjectConfirmation> </saml:Subject>
Conditions	MOET worden opgenomen. De attributen NotBefore en NotOnOrAfter MOETEN worden gevuld met respectievelijk het tijdstip van uitgifte van de assertion en **120 seconden** na de uitgifte van de assertion. De SAML <Audience> moet een SAML <AudienceRestriction> bevatten met hierin de EntityId van de Dienstaanbieder. Andere Audience elementen MOGEN NIET worden opgenomen. Andere Conditions MOGEN NIET worden opgenomen.
Advice	NIET opnemen
AuthnStatement	Het bevat de volgende elementen: 1 Een SAML <AuthnInstant> volgens SAML specificatie, bevat het tijdstip van de authenticatie 2 Een SAML <AuthnContext> volgens SAML specificatie. Hierin is in het SAML <AuthenticatingAuthority> element de EntityId van de AccountService opgenomen die de Gebruiker heeft geauthenticeerd, en een SAML <AuthnContextClassRef> met de wijze waarop authenticatie heeft plaatsgevonden. Voor de aanduiding dat beveiliging op transportniveau is uitgevoerd wordt optioneel ook het AuthenticatorTransportProtocol met “SSL” gevuld. 3 Een optioneel SAML <SubjectLocality> met hierin het IP-adres vanwaar de Gebruiker heeft ingelogd bij de AccountService. 4 Een SAML <SessionIndex>, die later gebruikt kan worden voor een Logout request.
AttributeStatement	De AttributeStatement bevat de ProfileServiceId waar voor deze gebruiker aanvullende gegevens op gehaald kunnen worden**.**

	<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" Name="ProfileServiceId"> <saml:AttributeValue xsi:type="xs:string"> 343DD34-1 </saml:AttributeValue> </saml:Attribute>

4.5.4 *ArtifactResolve*

Het uitwisselen van het Response-bericht gebeurt op basis van het Artiface Resolution Profile. Het ArtifactResolve bericht is een verzoek dat van de Dienstverlener aan de AccountService wordt verstuurd, en door de Dienstverlener wordt ondertekend met haar SAML2 Signing certificaat.

SAML specificatie: saml-profiles-2.0-os:5

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht, vulling conform SAML specificatie
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt, vulling conform SAML specificatie
Destination	NIET opnemen
Consent	NIET opnemen
Issuer	MOET de EntityId van de Dienstverlener bevatten
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Issuer .SPProviderID	NIET opnemen
Signature	MOET de elektronische handtekening van de Dienstverlener over het hele bericht bevatten.
Extensions	NIET opnemen
Artifact	Bevat het Artifact dat als antwoord op het AuthnRequest was ontvangen

4.5.5 *ArtifactResponse*

Het antwoord op een ArtifactResolve bericht, is een ArtifactResponse. Dit bericht wordt door de AccountService gemaakt en ondertekend met haar SAML2 Signing certificaat. In het ArtifactResponse-bericht wordt het Response-bericht verstuurd dat als antwoord op het AuthnRequest bericht is aangemaakt.

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht, vulling conform SAML specificatie
InResponseTo	Uniek kenmerk van het AuthnRequest waarop dit Response bericht het antwoord is.
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt.
Destination	NIET opnemen

Consent	NIET opnemen
Issuer	MOET de EntityId van de AccountService bevatten
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Issuer .SPProviderID	NIET opnemen
Signature	MOET de elektronische handtekening van de AccountService over het hele bericht bevatten.
Extensions	NIET opnemen
Status	MOET een element StatusCode bevatten
Status.StatusCode	Bevat de waarde "urn:oasis:names:tc:SAML:2.0:status:Success" Zie saml-core-2.0-os:3.5.3
StatusDetail	NIET opnemen
##any	MOET een Response bericht bevatten.

4.6 Logoutproces

Voor het uitloggen wordt het SAML Single Logout Profile gebruikt. In principe wordt het uitloggen geïnitieerd door de Dienstaanbieder. Hierbij wordt eerst bepaald bij welke AccountService de gebruiker een Login-sessie is aangegaan. De gebruiker wordt bij deze AccountService uitgelogd, en dit wordt door deze AccountService doorgegeven wordt aan alle andere Dienstaanbieders die met de betreffende gebruiker een sessie zijn aangegaan. De logout-functie kan zowel bij een Dienstaanbieder als AccountService geïnitieerd worden.

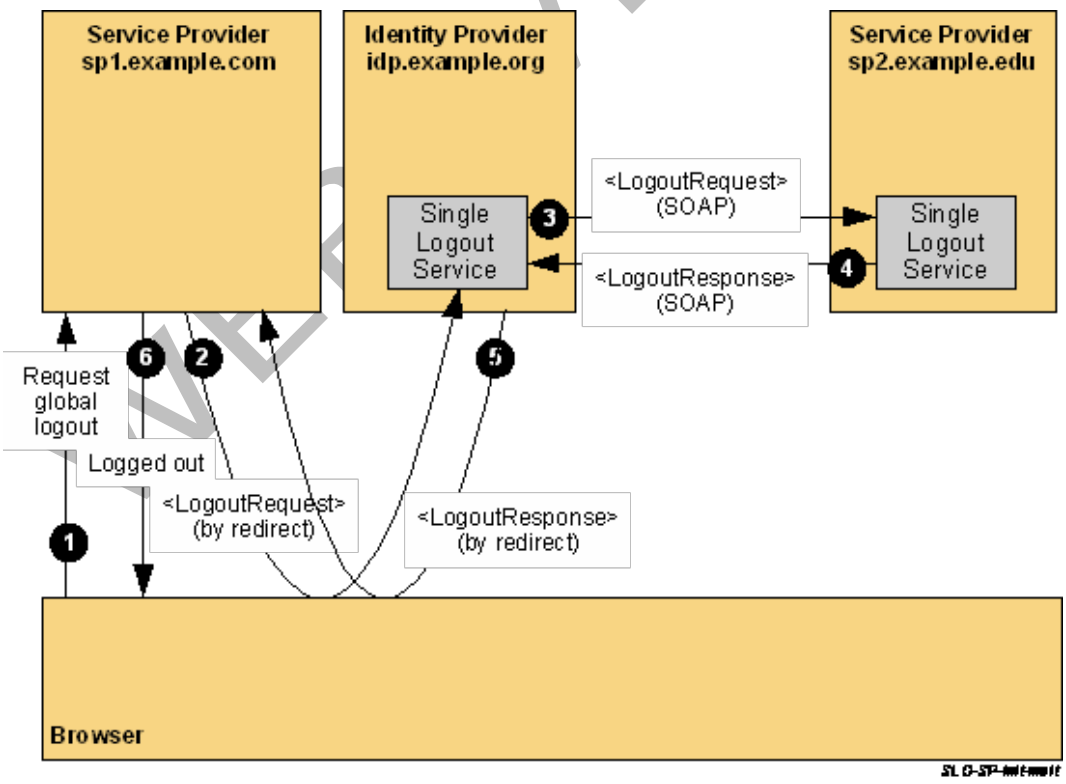


Figure 3 Single Logout proces

Het proces zoals het wordt geïnitieerd vanuit de Service Provider (Dienstaanbieder), wordt hieronder uiteengezet. Merk op dat wanneer het proces wordt geïnitieerd vanuit een Identity Provider (AccountService), dit vanaf stap 3 tot hetzelfde berichtenverkeer leidt.

1) Gebruiker geeft bij dienstaanbieder aan dat het wil uitloggen
De gebruiker heeft een actieve sessie bij de dienst, en geeft aan uit te willen loggen. De dienst initieert het logout-proces door te bepalen bij welke AccountService de gebruiker is ingelogd, en bereidt het verzoek voor om de gebruiker eerst hier uit te loggen. Tevens wordt op dit moment de sessie met de gebruiker beëindigd.

2) Versturen van <LogoutRequest> door Dienstaanbieder
Er wordt een LogoutRequest van de Dienstaanbieder aan de AccountService gestuurd. Dit gebeurt via een HTTP-Redirect binding. Hierdoor heeft de AccountService de mogelijkheid om een eventuele sessiecookie die de AccountService met de gebruiker heeft, de kunnen opruimen. In het LogoutRequest wordt de geauthenticeerde identiteit van de gebruiker als Subject meegegeven, als ook de SessionIndex die ontvangen is in het antwoord op het oorspronkelijke AuthnRequest-bericht. Het LogoutRequest bericht bevat een digitale handtekening van de ServiceProvider.

3-4) AccountService bepaalt sessie deelnemers om de Gebruiker uit te loggen
Bij ontvangst van een LogoutRequest, zal de AccountService bepalen welke andere Dienstaanbieders gebruik hebben gemaakt van de Single Sign On sessie die moet worden beeindigd. De AccountService zal aan elk van deze Dienstaanbieders een LogoutRequest versturen, met daarin de Subject- en SessionIndex-attributen die tussen de AccountService en de betreffende Dienstaanbieder van toepassing is. De Dienstaanbieder beëindigt de sessie met de gebruiker, en antwoordt met een LogoutResponse-bericht aan de AccountService. Zowel het LogoutRequest- als het LogoutResponse-bericht zijn digitaal ondertekend door de versturende partij.
Deze door de AccountService geïnitieerde berichtenuitwisseling gebeurt via een SOAP-binding.

5) Versturen van <LogoutResponse> door AccountService
Nadat de AccountService een bevestiging heeft gekregen van het uitloggen bij alle Dienstaanbieders in de sessie, verstuurt de AccountService een LogoutResponse aan de aanvragende Dienstaanbieder. Dit bericht wordt digitaal getekend door de AccountService, en via HTTP-Redirect binding verstuurd.

6) Dienstaanbieder informeert gebruiker van resultaat
Het resultaat van het oorspronkelijke logout-verzoek wordt door de Dienstaanbieder teruggekoppeld aan de gebruiker.

Daarbij dient opgemerkt te worden, dat alle berichten via een SSLv3/TLS-verbinding moeten worden uitgewisseld.

4.7 Bericht specificaties Logout

4.7.1 LogoutRequest (SP naar IdP, IdP naar SP)

Het LogoutRequest is het verzoek tot beeindigen van een login-sessie van de Dienstaanbieder aan de AccountService, maar ook van de AccountService aan de Dienstaanbieder. Een voorbeeld van een LogoutRequest is opgenomen in het hoofdstuk met SAML2 Berichten. Het LogoutRequest bericht wordt altijd ondertekend door de verzendende partij. Wanneer het bericht van Dienstaanbieder naar AccountService wordt gestuurd, wordt HTTP-Redirect binding gebruikt.

Wanneer het bericht van AccountService naar Dienstaanbieder wordt gestuurd, wordt SOAP-binding gebruikt.

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht, vulling conform SAML specificatie
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt, vulling conform SAML specificatie
Destination	URL van de ontvangende partij waarnaar het bericht verstuurd wordt. Deze komt overeen met de aangegeven URL in de metadata (LogoutService) van de ontvangende partij
Issuer	Moet de EntityId van de verzendende partij bevatten
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Issuer .SPProviderID	NIET opnemen
Signature	Het hele bericht MOET ondertekend zijn door de Verzendende partij met haar SAML2 Signing certificaat.
NameID	Opnemen, dit MOET de waarde bevatten zoals deze in het Response-bericht aan de Dienstaanbieder is doorgegeven
NameID.Format	Indien deze in het Response-bericht van de authenticatie is opgenomen, MOET het attribuut met diezelfde waarde hier opgenomen worden.
SessionIndex	Opnemen, en MOET dezelfde waarde bevatten als in het AuthnStatement van het Response-bericht is doorgegeven

4.7.2 LogoutResponse (IdP naar SP, SP naar IdP)

Het LogoutResponse-bericht omvat het antwoord op een LogoutRequest bericht, en rapporteert hoe het verzoek is afgehandeld. Een voorbeeld van een LogoutRequest is opgenomen in het hoofdstuk met SAML2 Berichten. Het LogoutResponse bericht wordt altijd ondertekend door de verzendende partij. Het bericht wordt altijd via dezelfde binding verstuurd als waarop het LogoutRequest-bericht is ontvangen (een antwoord van IDP aan SP wordt via HTTP-Redirect binding verstuurd, en een antwoord van SP aan IDP wordt via SOAP binding verstuurd).

Element	**Beschrijving**
ID	Uniek kenmerk van het bericht, vulling conform SAML specificatie
Version	Versie van het SAML protocol. De waarde MOET "2.0" zijn.
IssueInstant	Tijd waarop het bericht is aangemaakt, vulling conform SAML specificatie
Destination	URL van de ontvangende partij waarnaar het bericht verstuurd wordt. Deze komt overeen met de aangegeven URL in de metadata (LogoutService) van de ontvangende partij
InResponseTo	MOET een verwijzing bevatten naar de ID-waarde van het LogoutRequest-bericht waar dit bericht op antwoordt

Issuer	Moet de EntityId van de verzendende partij bevatten
Issuer .NameQualifier	NIET opnemen
Issuer .SPNameQualifier	NIET opnemen
Issuer .Format	NIET opnemen
Issuer .SPProviderID	NIET opnemen
Signature	Het hele bericht MOET ondertekend zijn door de Verzendende partij met haar SAML2 Signing certificaat.
Status	MOET een element StatusCode bevatten
Status.StatusCode	MOET opgenomen worden en het resultaat van het LogoutRequest bevatten. Indien het verzoek geslaagd is, moet het de waarde "urn:oasis:names:tc:SAML:2.0:status:Success" bevatten. Overige waardes voor StatusCodes zijn conform SAML2 specificatie, zie saml-core-2.0-os, 3.2.2.2
Status.StatusMessage	MAG opgenomen worden. Indien opgenomen moet het een beschrijving van het resultaat bevatten.
Status.StatusDetail	NIET opnemen

4.8 Metadata

De SAML2 specificatie bevat een metadata specificatie. Hierin zijn de kenmerken gedefinieerd van een systeem dat als SAML2 entiteit functioneert, zoals de beschikbare endpoints en het SAML2 Signing certificaat. Hieronder worden aanvullende afspraken gespecificeerd welke de Services ondersteunen.

- AccountService en Dienstverlener (zoals bijvoorbeeld OrderListServices) waarvoor Gebruikers zich dienen te authenticeren, dienen in staat te zijn SAML2 Metadata te kunnen verwerken.
- In de SAML2 Metadata het SAML2 Signing certificaat van de betreffende SAML2 entiteit te worden opgenomen
- HTTP Redirect en SOAP binding moet door alle Dienstaanbieders en AccountServices ondersteund worden.

4.8.1 *Uitwisselen metadata*

Partijen die met elkaar via SAML2 willen communiceren, dienen zelf af te spreken hoe ze metadata met elkaar uitwisselen. Uitwisseling is gebaseerd op de principes in het hoofdstuk 4.3

4.9 SAML Identity Provider Discovery Profile

De Gebruiker kan bepalen welke van de eventueel beschikbare AccountService hij wil gebruiken voor authenticatie. Voor het bepalen van de AccountService wordt het SAML Identity Provider Discovery Profile gebruikt. Hierbij wordt gebruik gemaakt van een Cookie in de browser van de Gebruiker of metadata in de aanroep naar de dienst van de Dienstverlener. De Cookie kan meerdere EntityId waarden van AccountServices bevatten. In de lijst is ook de EntityId aangegeven van de AccountService die het laatst gebruik is. Mochten deze beide opties niet beschikbaar zijn, omdat er geen sessie met de Gebruiker is of er in de sessie met de gebruiker geen Cookie beschikbaar is, dan zal de Dienstaanbieder een lijst met vertrouwde AccountServices tonen waaruit de gebruiker kan kiezen.

4.10 Voorbeelden

4.10.1 Voorbeeld bericht AuthnRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="_c7055387-
af61-4fce-8b98-e2927324b306 Version="2.0" IssueInstant="2012-03-17T18:43:10.738Z"
ForceAuthn="true" Destination=" Accountservice URL" AssertionConsumerServiceIndex="1"
AttributeConsumingServiceIndex="1">
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
urn:eck:abc:ecsvc:dienst
</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>
                <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
                c14n#"/>
                <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
                more#rsa-sha256"/>
                <ds:Reference URI=" ">
                <ds:Transforms>
                <ds:Transform
                Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
                sha256"/>
                <ds:DigestValue>
                </ds:DigestValue>
                </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue>
                </ds:SignatureValue>
                <ds:KeyInfo>
                <ds:X509Data>
                <ds:X509Certificate>
…
                </ds:X509Certificate>
                </ds:X509Data>
                </ds:KeyInfo>
        </ds:Signature>
                <samlp:RequestedAuthnContext Comparison="minimum">
                <saml:AuthnContextClassRef
                xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
                urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
                </saml:AuthnContextClassRef>
                </samlp:RequestedAuthnContext>
        </samlp:AuthnRequest>
```

4.10.2 Voorbeeld Artifact response

HTTP redirect Artifact: https://dienst.domein/sso/Artifact?SAMLart=
AAQAABhQELuXX0Vl6wIkAxtOjnRogkgRKWEt7Nxr9EL9VmJLD9ZkmSseVis=

4.10.3 Voorbeeld ArtifactResolve

```
<?xml version="1.0" encoding="UTF-8"?>
<soap11:Envelope
xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"><soap11:Body>
<samlp:ArtifactResolve xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
ID="_163a9ac7aa5f6551fb024f95bcdc52db" IssueInstant="2013-03-24T09:36:39.967Z" Versi
on="2.0">
```

```
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">urn:eck:abc:ecsvc:dienst</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#_163a9ac7aa5f6551fb024f95bcdc52db">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="ds saml samlp"/>
</ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>j5bGfujUaNBZZyhifvdsxhje3rY=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
g43y+1m49Kpn0QpqdnAXEI1A+cXup2IdJn1ZxhjhZmrEoDY2UBy0O8n+pkL78y2MHcwCbiFFt4coF
mahQhP+wDSAampmAMrZve2voa/KwwX1gSCI/dVgbImykRHzuhbGG0LYOLJvBS5QACtlg7olsTpVA
s12My8/nf7BElz2tns=
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>

</ds:Signature>
<samlp:Artifact>AAQAABhQELuXX0Vl6wIkAxtOjnRogkgRKWEt7Nxr9EL9VmJLD9ZkmSse
Vis=</samlp:Artifact>
</samlp:ArtifactResolve>
</soap11:Body>
</soap11:Envelope>
```

4.10.4 *Voorbeeld ArtifactResponse, Response en Assertion*

```
<?xml version="1.0" encoding="UTF-8"?>
<soap11:Envelope
xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"><soap11:Body>
<samlp:Response ID="_ c7055387-af61-4fce-8b98-e2927324b306"
  IssueInstant="2012-03-17T18:45:10.738Z" Version="2.0">

<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">urn:eck:kennisnet:acctsvc:federatiehub</saml:Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>
                        <ds:CanonicalizationMethod
                                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        <ds:SignatureMethod
                        Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
                        "/>
                        <ds:Reference URI="#_c7055387-af61-4fce-8b98-e2927324b306">
                        <ds:Transforms>
                                <ds:Transform
```

```
                                    Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped
                                    signature"/>
                                    <ds:Transform
                                    Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                    <InclusiveNamespaces PrefixList="#default saml ds xs xsi"
                                    xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                                    </ds:Transform>
                            </ds:Transforms>
                            <ds:DigestMethod
                                    Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
                            sha256"/>
                            <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValu>
                            </ds:Reference>
                    </ds:SignedInfo>
            <ds:SignatureValue>
                    x/GyPbzmFEe85pGD3c1aXG4Vspb9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+5
                    EIYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m
                    3D
                    w6vKhaqledl0BYyrIzb4KkHO4ahNyBVXbJwqv5pUaE4=
            </ds:SignatureValue>
            <ds:KeyInfo>
                    <ds:X509Data>
                    <ds:X509Certificate>
...
                    </ds:X509Certificate>
                    </ds:X509Data>
            </ds:KeyInfo>
</ds:Signature>
<Status>
<StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</Status>


<saml:Assertion ID="_3c39bc0fe7b13769cab2f6f45eba801b1245264310738"
  IssueInstant="2012-03-17T18:45:10.738Z" Version="2.0">
   <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
          urn:eck:kennisnet:acctsvc:federatiehub
</saml:Issuer>

<saml:Subject>
            <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
              Some-UserId@Some-UserAdministrationId
          </saml:NameID>
          <saml:SubjectConfirmation
                    Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                    <saml:SubjectConfirmationData
                              Recipient="urn:eck:abc:ecsvc:dienst"
                              NotOnOrAfter="2013-12-31T12:00:00"
                              InResponseTo= Authn_request_identifier_1234567 >
                    </saml:SubjectConfirmationData>
          </saml:SubjectConfirmation>

</saml:Subject>

<saml:Conditions NotBefore="2012-03-17T18:45:10.738Z"
  NotOnOrAfter="2012-03-17T18:50:10.738Z">

            <saml:AudienceRestriction>
                    <saml:Audience> urn:eck:abc:ecsvc:dienst</saml:Audience>
            </saml:AudienceRestriction>
</saml:Conditions>

<saml:AuthnStatement
          AuthnInstant="2013-03-17T18:45:10.738Z"
          SessionIndex="2c81606885bf79c716eb2c082a2249a5" >

          <saml:SubjectLocality Address="87.213.96.194"/>

  <saml:AuthnContext>
```

```
        <saml:AuthnContextClassRef>
                urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        </saml:AuthnContextClassRef>
  </saml:AuthnContext>
</saml:AuthnStatement>

<saml:AttributeStatement>
        <saml:Attribute NameFormat="DTDL">Name="ProfileServiceId"
                <saml:AttributeValue> 99ZZ99-1</saml:AttributeValue>
        </saml:Attribute>
</saml:AttributeStatement>

</saml:Assertion>
</samlp:Response>
</samlp:ArtifactResponse>
</soap11:Body>
</soap11:Envelope>
```

4.10.5 *Voorbeeld LogoutRequest*

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                ID="_12344ea37e28763e351189529639b9c2b150ff37e5"
                Version="2.0"
                IssueInstant="2012-03-17T18:47:15.455Z"
                Destination="https://dienst.eck.nl/profiles/saml2/logout">
        <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
                urn:eck:abc:ecsvc:dienst
        </saml:Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
        <ds:Reference URI=" ">
        <ds:Transforms>
        <ds:Transform
        Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
        <ds:DigestValue>
        </ds:DigestValue>
        </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>
        </ds:SignatureValue>
        <ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
        …
</ds:X509Certificate>
</ds:X509Data>
        </ds:KeyInfo>
        </ds:Signature>

        <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
        format:unspecified">
                Some-UserId@Some-UserAdministrationId

        </saml:NameID>

        <samlp:SessionIndex>2c81606885bf79c716eb2c082a2249a5</samlp:SessionIndex>

</samlp:LogoutRequest>
```

4.10.6 *Voorbeeld LogoutResponse*

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<samlp:LogoutResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
                ID="_56784ea37e28763e351189529639b9c2b150ff37e5"
                Version="2.0"
                IssueInstant="2008-06-03T12:59:57Z"
                Destination="https://accountservice.eck.nl/profiles/saml2/sp/slo"
                InResponseTo="_12344ea37e28763e351189529639b9c2b150ff37e5">
        <saml:Issuer>https://openidp.feide.no</saml:Issuer>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
        <ds:Reference URI=" ">
        <ds:Transforms>
        <ds:Transform
        Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
        <ds:DigestValue>
        </ds:DigestValue>
        </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>
        </ds:SignatureValue>
        <ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
        …
</ds:X509Certificate>
</ds:X509Data>
        </ds:KeyInfo>
        </ds:Signature>
        <samlp:Status>
                <samlp:StatusCode
        Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
                <samlp:StatusMessage>Logout success</samlp:StatusMessage>
        </samlp:Status>
</samlp:LogoutResponse>
```

4.10.7 *Voorbeeld metadata AccountService*

Hieronder volgt ter info een voorbeeld van de SAML2 Metadata van een AccountService.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
entityID="urn:eck:kennisnet:acctsvc:federatiehub">
…
<ds:Signature>...</ds:Signature>
…

<IDPSSODescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
WantAuthnRequestsSigned="true">
<SingleSignOnService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://accountservice.domein/service/saml20_sso"/>
<ArtifactResolutionService isDefault="true" index="0"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https:// accountservice. Domein AccountService /SAML/Artifact"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https:// accountservice. Domein AccountService /SAML/SLO/SOAP"/>
<SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https:// accountservice. Domein AccountService /SAML/SLO/Browser"
ResponseLocation="https:// domein /SAML/SLO/Response"/>
```

```
…
<KeyDescriptor use="signing">
<ds:KeyInfo>
        <ds:KeyName>accountservice.domein SSO Key</ds:KeyName>
        <ds:X509Data>
                <ds:X509Certificate>
        ...
                </ds:X509Certificate>
        </ds:X509Data>
</ds:KeyInfo>
</KeyDescriptor>
</IDPSSODescriptor>
...
</EntityDescriptor>
```

4.10.8 Voorbeeld metadata Dienstverlener

Hieronder volgt ter info een voorbeeld van de SAML2 Metadata van een Dienstverlener.

<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
entityID="PartyId Dienstaanbieder">
…
<ds:Signature>...</ds:Signature>
…

<**SPSSODescriptor** AuthnRequestsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<**SingleLogoutService**
Binding="urn:oasis:names:tc:SAML:2.0:bindings:**SOAP**"
Location="https:// domein dienstaanbieder /SAML/SLO/SOAP"/>
<**SingleLogoutService**
Binding="urn:oasis:names:tc:SAML:2.0:bindings:**HTTP-Redirect**"
Location=" https:// domein dienstaanbieder /SAML/SLO/Browser"
ResponseLocation="https:// domein /SAML/SLO/Response"/>
<**AssertionConsumerService** isDefault="true" **index**="0"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:**HTTP-Artifact**"
Location="https:// domein dienstaanbieder /SAML/SSO/Artifact"/>
<**AssertionConsumerService index**="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:**HTTP-POST**"
Location="https:// domein dienstaanbieder /SAML/SSO/POST"/>

…
<**KeyDescriptor** use="signing">
<ds:KeyInfo>
<ds:KeyName>Domein SSO Key</ds:KeyName>
</ds:KeyInfo>
</KeyDescriptor>
…
</SPSSODescriptor>
…
</EntityDescriptor>

5. Foutafhandeling

Bij berichtenuitwisseling tussen partijen kunnen fouten optreden. Er zijn drie verschillende fouttypen te onderkennen:

1. Technische fouten.
2. Functionele fouten
3. Protocol- en transportfouten (dus TLS/SSLv3 of HTTP fouten).

5.1 Technische fouten

Deze foutmeldingen hebben betrekking op situaties waarbij het request-bericht door de web service is ontvangen, maar het om een technische redenen niet goed kan verwerken en er een foutmelding wordt teruggestuurd. Technische fouten worden gecodeerd met een negatieve integer. Een overzicht van technische foutmeldingen is in bijlage A opgenomen.

5.2 Functionele fouten

De functionele fouten worden gecodeerd met een positieve integer en gelden per web service. Functionele fouten worden per functie gespecificeerd in het Functioneel Ontwerp document.

5.3 Protocol- en transportfouten

Deze fouten vinden niet op basis van de DTDL-applicatieprotocollen plaats. Hierdoor zijn hier geen standaard berichten voor gedefinieerd en is men afhankelijk van de onderliggende protocollen zoals TLS en/of HTTP.

5.4 Foutmeldingsbericht

Alle web services implementeren het standaard foutmeldingsbericht. Hier wordt de fout die is opgetreden beschreven en de code die hiermee samenhangt. Het type dat gebruikt wordt om een fout te beschrijven, is het FaultMessageType, en is opgenomen als een van de generieke types in de CommonTypes definities.

Noot: bij elke foutmelding wordt altijd een FaultMessage gebruikt. Het Confirmation bericht heeft een soortgelijke structuur, maar dient alleen gebruikt te worden bij geen foutmeldingsbericht.

6. Identifiers

Identifiers die in de keten gedeeld worden, moeten binnen de hele keten uniek zijn. Elke Identifier wordt uniek gemaakt, door deze betekenis te geven binnen de context van een administratie (AdministrationId). In de XML schema's worden zowel lokale identifiers (zoals bijvoorbeeld UserId, OrderId, etc.) als administraties (AdministrationId) gespecificeerd als IdentifierType (xsd:string).

6.1 IdentifierType

Een identifier, waarmee bijvoorbeeld een gebruiker of een administratie wordt aangeduid, is altijd van het type IdentifierType. Het IdentifierType wordt gespecificeerd als een xsd:string.

6.2 AdministrationId

De AdministrationId is de identifier van een administratie waarmee lokale Identifiers binnen de keten uniek kunnen worden gemaakt. Hiermee wordt binnen de keten uniciteit gegarandeerd. De AdministrationId wordt in de XML schema's gedefinieerd door een xsd:string.
Een AdministrationId mag bestaan uit alle printbare karakters, minus het '@'-karakter.

6.3 Impliciete identifier normalisatie

Er zijn situaties waarin een globaal unieke identifier als een waarde moet worden gebruikt (bijvoorbeeld voor een user identifier). In dit geval kan de lokale identifier 'LocalId' in combinatie met de AdministrationId worden samengevoegd als: *LocalId@AdministrationId*.

Voorbeelden:
1. Een gebruiker met UserId 'u18211' die binnen administratie met AdministrationId 'basis.school.domein' wordt geidentificeerd, kan volgens bovenstaande afspraak worden genormaliseerd tot 'u18211@basis.school.domein'
2. Een betaling met PaymentId 'p012345' die binnen administratie met PaymentAdministrationId 'store.distributeur.domein' is geregistreerd , kan volgens bovenstaande afspraak worden genormaliseerd tot 'p12345@store.distributeur.domein'

6.4 ContentId

De alContentId is de identificatie voor een leermiddel binnen de keten. Er zijn momenteel meerdere specificaties voor dit gegeven zoals EAN, ISBN[3] en UPI[4]. De ContentId is in de XML schema's gedefinieerd door een xsd:string.

[3] http://nl.wikipedia.org/wiki/Internationaal_Standaard_Boeknummer

[4] http://www.edustandaard.nl/afspraken/upi

7. Services

7.1 Service beschrijving

In de onderstaande tabel wordt per Web Service aangegeven in welk bestand de WSDL respectievelijk de XSD specificatie wordt gepubliceerd. Daarbij geldt dat voor "vx.y" de versie moet worden ingevuld. Bijvoorbeeld: een WSDL-specificatie van versie 1.6 van de AccountService, wordt onder de naam ECK-DTDL-AccountService-v1.6.wsdl gepubliceerd.

Service	**WSDL, XSD**
AccountService	ECK-DTDL-AccountService-vx.y.wsdl ECK-DTDL-AccountService-vx.y.xsd
ProfileService	ECK-DTDL-ProfileService-vx.y.wsdl ECK-DTDL-ProfileService-vx.y.xsd
ContentListService	ECK-DTDL-ContentListService-vx.y.wsdl ECK-DTDL-ContentListService -vx.y.xsd
OrderListService	ECK-DTDL-OrderListService-vx.y.wsdl ECK-DTDL-OrderListService-vx.y.xsd
OrderService	ECK-DTDL-OrderService-vx.y.wsdl ECK-DTDL-OrderService -vx.y.xsd
PaymentService	ECK-DTDL-PaymentService-vx.y.wsdl, ECK-DTDL-PaymentService-vx.y.xsd
DistributionService	ECK-DTDL-DistributionService-vx.y.wsdl ECK-DTDL-DistributionServiceSchema-vx.y.xsd
LicenseService	ECK-DTDL-LicenseService-vx.y.wsdl ECK-DTDL-LicenseService-vx.y.xsd
ContentLocationService	ECK-DTDL-ContentLocationService-vx.y.wsdl ECK-DTDL-ContentLocationService -vx.y.xsd

7.2 Service beveiliging

De onderstaande tabel geeft van alle Web Services aan of ze beveiligd moeten worden.

Gegevensuitwisseling	**Beveiliging**
AccountService	Beveiligd transportkanaal
ProfileService	Beveiligd transportkanaal
ContentListService	Onbeveiligd transportkanaal
OrderListService	Beveiligd transportkanaal
OrderService	Beveiligd transportkanaal
PaymentService	Beveiligd transportkanaal
DistributionService	Beveiligd transportkanaal
LicenseService	Beveiligd transportkanaal
ContentLocationService	Beveiligd transportkanaal

8. Bijlage A: Foutmeldingen

In de onderstaande tabel zijn de foutmeldingen opgenomen die de web services minimaal ondersteunen. Hierbij worden een aantal nummer reeksen gereserveerd. Codes vanaf -1 zijn strikt technische aard. Vanaf 0 en hoger worden functionele foutmeldingen opgenomen. Deze zullen vaak specifiek voor een bepaalde service gelden en bij de betreffende beschrijving opgenomen zijn.

Code	**Melding**
-1	Algemene fout
-2	Authenticatie fout
-101	Leermiddel Identifier bestaat niet
-102	Rechtspersoon Identifier (SchoolId) ontbreekt/incorrect
-103	Aantal niet juist (>=1)
-108	Identifier ontbreekt/niet gevonden
-200	Bericht is niet conform het contract (WSDL en/of XSD schema)
-201	Een verplicht gegeven ontbreekt (0)
-300	Afnemer kan niet geïdentificeerd worden
-301	Afnemer is niet geautoriseerd

9. Afkortingen / verklarende woordenlijst

Term/afkorting	Definitie
DEFLATE encoding	Een data compressie algoritme waarbij geen data verlies optreedt. (Zie http://www.ietf.org/rfc/rfc1951.txt en http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
CA	Een certificaatautoriteit (CA, certificate authority of ook wel certification authority) is in de cryptografie een entiteit die digitale certificaten verleent aan andere partijen. De bedoeling is dat het digitale certificaat bewijst dat de eigenaar daadwerkelijk degene is die hij beweert te zijn. (zie http://nl.wikipedia.org/wiki/Certificaatautoriteit)
CSP	Certificate Service Provider (zie CA)
CRL	Een certificate revocation list (CRL) is een lijst van digitale identiteitscertificaten die vervallen of ongeldig zijn. Een CRL wordt bijgehouden door een certificate authority (CA). (Zie http://nl.wikipedia.org/wiki/Certificate_revocation_list)
HTTP 1.1	Hypertext Transfer Protocol (http://www.w3.org/Protocols/rfc2616/rfc2616.html)
Internet domeinnaam	Domeinnaam van een internet dienst (zie http://tools.ietf.org/html/rfc1034)
IP	Internet Protocol (http://www.ietf.org/rfc/rfc791.txt versie 4, http://www.faqs.org/rfcs/rfc2460.html versie 6)
PKI	Public Key Infrstructure (zie http://nl.wikipedia.org/wiki/Public_key_infrastructure)
SAML2	OASIS Security Assertion Markup Language V2.0 (zie http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
SSO	Bij Single Sign On krijgt een gebruiker na één keer succesvol geauthenticeerd te zijn gedurende dezelfde sessie ook toegang tot andere diensten, zonder zichzelf opnieuw te hoeven authenticeren. Het betreft Cross Domain Single Sign On (CDSSO) indien dit over verschillende organisatiedomeinen plaatsvindt.
SSL	Secure Socket Layer (zie http://nl.wikipedia.org/wiki/Secure_Sockets_Layer)
SOAP	Simple Object Access Protocol, XML berichtstandaard (http://www.w3.org/TR/2000/NOTE-SOAP-20000508/)
TLS	Transport Layer Security (zie http://datatracker.ietf.org/wg/tls/charter/)
URL	Uniform Resource Location (zie http://www.ietf.org/rfc/rfc1738.txt)
UUID	Universally Unique Identifier (zie ISO/IEC 9834-8:2008)
Whitelist	Een lijst welke aangeeft welke partijen gemachtigd zijn bepaalde diensten af te nemen. (zie http://en.wikipedia.org/wiki/Whitelist)
WS-I Basic Profile	WS-I is een consortium dat zich richt op het verbeteren van web service interoperabiliteit. Basic Profile versie 1.2 is meest recente versie waarin voorwaarden voor interoperabele gegevevensuitwisseling opgenomen zijn. (zie http://www.ws-i.org/)
WSDL	Web Service Discription Language (http://www.w3.org/TR/wsdl)
X509 Certificaat	Zie http://en.wikipedia.org/wiki/X.509

10. Referenties

[1] Educatieve contentketen, Distributie en toegang van digitale leermiddelen – Definities en procesmodel, versie 1.6 (concept), 15 oktober 2013
[2] Educatieve contentketen, Distributie en toegang van digitale leermiddelen – Functioneel model, versie 1.6 (concept), 15 oktober 2013