



Portal Verbund Protokoll		Konvention
		PVP 1.9.2
		Empfehlung
Kurzbeschreibung	<p>Das Portalverbundsystem ermöglicht das Zusammenwirken von Stammportalen zur Registrierung von Benutzern mit ihren Zugriffsrechten einerseits und Anwendungsportalen zur Überprüfung des berechtigten Zuganges zu Anwendungen andererseits.</p> <p>Die Authentifizierung und Autorisierung kann delegiert werden.</p> <p>Der Aufwand für die Verwaltung der Benutzer wird reduziert und ein Single Sign-On unterstützt.</p>	
Autor(en):	Rainer Hörbe Peter Pfläging	Projektteam / Arbeitsgruppe AG-IZ Peter Pfläging
Beiträge von:	Harald Stradal, Peter Pichler, Michael Werzowa, Wolfgang Kremser, Joachim Minichhofer und etliche Andere	
Anhänge:	Die Datei pvp1.xsd ist normativer Bestandteil dieses Dokuments.	

Version 1.8.10:	20.05.2007	Fristablauf:	18.06.2007
Version 1.9.0:	20.05.2007	Fristablauf:	28.04.2009
Detailversion 1.9.1:	14.10.2009	Fristablauf:	22.10.2009
Detailversion 1.9.2:	20.01.2011	Fristablauf:	

Inhaltsverzeichnis

1.Zweck.....	3
2.Schreibweise.....	3
2.1. Begriffsbestimmung.....	3
2.2.Vergleiche zu anderen Nomenklaturen.....	4
2.3. Schreibweise der EBNF.....	4
3.Architektur.....	6
3.1. Entitäten der Authentifizierung und Autorisierung.....	6
3.2. Message Sequence.....	6
4.Datenmodell des PVP-Tokens.....	6
4.1. Hauptbestandteile des PVP-Tokens.....	6
4.2. Datenmodell in UML-Notation.....	8
4.3. Datenmodell in XML-Darstellung.....	9
4.4.Beschreibung der Parameter.....	10
4.5. Mapping der Attribute im PVP-Token.....	14
4.6. Syntaktische Definition des PVP-Tokens.....	16
5.Versionsabstimmung zwischen Client und Server.....	18
6.Namensräume für URLs.....	19
6.1. Betrieb von Anwendungsportalen als Reverse Proxy.....	19
6.2. Globale Namensräume für Anwendungen.....	19
7.Application Chaining.....	20
8.Zertifikate.....	20
9.Protokollbindung HTTP.....	21
9.1. Abbildung mittels HTTP Header Fields.....	21
9.2. Fehlermeldungen.....	21
9.3. Application Chaining.....	21
10. Protokollbindung SOAP.....	22
10.1. Namespaces.....	22
10.2. Security Element pvpToken.....	22
10.3. Zeichencodierung.....	23
10.4. Fehlermeldungen.....	23
11. HTTP Fehlermeldungen.....	24
Anhang A HTTP Beispiel-Request User Principal.....	25
Anhang B HTTP Beispiel-Request System Principal.....	26
Anhang C Beispiel mit Application Chaining.....	27
Anhang D SOAP Beispiel-Request	30
Anhang E Beispiele für Rechte und Rechteparameter.....	31
Anhang F Implementierungshinweise für Reverse Proxies.....	32
a.Problemstellung.....	32
b.Umschreiben von HTTP-Headern durch einen Reverse Proxy.....	32
Anhang G Debugschnittstelle für Clients.....	33
Anhang H Referenzen.....	34
Anhang I Änderungen von Version 1.9.1 zu 1.9.2.....	36
Anhang J Änderungen von Version 1.9.0 zu 1.9.1.....	36
Anhang K Änderungen von Version 1.8.10 zu 1.9.0.....	36

1. Zweck

Das Portalverbundsystem ermöglicht die Delegation von Benutzer-Identitäten und Berechtigungen [PV-Whitepaper]. Das Protokoll erweitert die Kommunikation zwischen Stamm- und Anwendungsportalen, indem vertrauenswürdige Aussagen über Authentizität, Autorisierung und Verrechnungsdaten von Benutzern kommuniziert werden.

Autorisierung bedeutet in diesem Zusammenhang, dass einem Benutzer für den Zugriff auf eine Ressource Rechte, Rechteparameter und eine Sicherheitsklasse zugewiesen werden.

Die Kommunikation zwischen den Portalen muss Integrität und Vertraulichkeit gewährleisten.

Der Portalverbund (PV) im Sinne von [PVV 1.0] ist zur Kommunikation zwischen Körperschaften öffentlichen Rechts vorgesehen. Das Protokoll kann aber in einem anderen rechtlichen Kontext für weitere Zwecke verwendet werden:

- Kommunikation zwischen Behörden und Nicht-Behörden auf Grund bilateraler Vereinbarungen
- Kommunikation zwischen internen Stamm- und Anwendungsportalen
- Kommunikation zwischen Anwendungsportalen und Anwendungen

2. Schreibweise

Beispiele und Fußnoten sind nicht Teil der Spezifikation.

2.1. Begriffsbestimmung

Die Begriffe sind in [PVV 1.0] definiert. Ergänzend dazu wird festgelegt:

Application Chaining

Das ist der Fall, wenn der Zugriff eines Benutzers auf eine Anwendung im Umweg über eine oder mehrere andere Anwendung(en) erfolgt.

Subteilnehmer

Bezeichnet eine Organisation, die mittels der Vereinbarung [PV-DASI] über einen Teilnehmer am Portalverbund teilnimmt.

Participant

Bezeichnet jene zugriffsberechtigte Stelle, die ein Teilnehmer oder ein Subteilnehmer ist. Ein Participant muss eine Rechtsperson sein, kann also keine Organisationseinheit sein.

Verrechnungsdaten

bestehen aus der Identifikation des Rechnungsempfängers, und der Liste der möglichen Kostenstellen und Gebührenstufen des Anwenders. Die Auswahl der konkreten Kostenstelle und Gebührenstufe einer Transaktion erfolgt in der Anwendung, nicht im PVP.

ZDA

Zertifizierungsdienstanbieter

2.2. Vergleiche zu anderen Nomenklaturen

<i>PVP</i>	<i>SAML</i>	<i>RFC 2904</i>	<i>eID (IDABC)</i>
Anwendung	Service Provider	Service Equipment	Service
Anwendungsportal	Service Provider	AAA-Server (Org1)	
Portalverbund	Circle of Trust	(agreement)	
Principal, Benutzer	Client	User	Entity
Stammportal	Identity Provider	AAA-Server (Org2)	Identity Provider
zugriffsberechtigte Stelle		User Home Organization	

2.3. Schreibweise der EBNF

Diese Spezifikation verwendet EBNF (erweiterte Backus-Naur Form). Mit EBNF wird über Produktionsregeln die Menge der möglichen Zeichenketten eine „Sprache“ definiert.

Diese Schreibweise ist vom RFC 822 abgeleitet und wird wie folgt definiert:

Name := Regel

Eine (Produktions-) Regel besteht aus einem oder mehreren Elementen. Ein Element ist Literal oder wiederum eine Regel. „:=“ heißt so viel wie „besteht aus“.

Regeln, die in [RFC2616] als „Basic Rules“ definiert sind, werden mit Großbuchstaben geschrieben, wie SPACE, CRLF, DIGIT, ALPHA, LWS.

Regeln können zur Klarstellung im Fließtext mit spitzen Klammern „<>“ geschrieben werden.

"Literal"

Literale sind durch Anführungszeichen markiert und bedeuten fixen Text, der nicht mehr weiter ersetzt wird. Der Text ist case-insensitive, wenn es nicht anders angegeben ist.

Kommentar

Text nach einem Semikolon bis zum Zeilenende wird als Kommentar betrachtet, z.B. ; Erklärung in die Spezifikation eingebettet

Regel-1 | Regel-2

Elemente, die durch einen vertikalen Strich ("|") getrennt sind, sind Alternativen, z.B. "JA" | "NEIN"

!Regel-1 Regel-2!

Elemente innerhalb von Rufzeichen werden als einzelnes Element betrachtet. Z.B. kann die Regel <"bearbeite " ! "alle " | "keine " ! "Anfragen"> zu "bearbeite alle Anfragen" und "bearbeite keine Anfragen" führen. Die übliche Schreibweise mit runden Klammern „()“ wird hier nicht verwendet, um die Lesbarkeit von Parameterlisten zu verbessern.

*Regel +Regel {N}Regel {N-M}Regel

Das Zeichen vor einem Element bedeutet die Anzahl der Wiederholungen des Elements:

*	0 oder mehr
+	1 oder mehr
{N}	N
{N-M}	größer gleich N und kleiner gleich M

[Regel]

Eckige Klammern bedeuten, dass das Element optional ist.

#;Regel

Das #-Zeichen ist eine Kurzschreibweise für eine Liste mit Semikolon als Literal für das Trennzeichen. Die Form <n>#<t> bedeutet mindestens <n> Elemente, die von einem oder mehreren Trennzeichen <t>, und optional LWS getrennt sind.

(*LWS element *(*LWS ";" *LWS element))

kann dargestellt werden durch

1#;element

Null-Elemente sind erlaubt, werden aber nicht gezählt. #Element bedeutet 0 bis unendlich viele Elemente, 1#element ein oder mehrere Elemente.

3. Architektur

3.1. Entitäten der Authentifizierung und Autorisierung

Im Kontext des Portalverbundprotokolls wird nur die Kommunikation zwischen Stamm- und Anwendungsportal definiert; darüber hinaus wird die Kommunikation zwischen Benutzer und Stammportal betrachtet.

Für das Protokoll sind folgende Entitäten erforderlich:

Principal (User- oder System Principal)

Stammportal

Anwendungsportal

3.2. Message Sequence

Der Portalverbund ist grundsätzlich neutral gegenüber dem Modell der Interaktionen zwischen Benutzern, Portalen und Anwendungen. Für PVP ist derzeit jedoch ein Reverse Proxy-Modell vorgesehen, wie es im RFC 2904 Abschnitt 3.1.1 als „agent sequence“ definiert ist.

4. Datenmodell des PVP-Tokens

Die im Protokoll übermittelten Parameter werden im PVP-Token zusammengefasst. In den folgenden Teilabschnitten wird dieses Datenmodell definiert.

4.1. Hauptbestandteile des PVP-Tokens

Die Information, die im PVP übermittelt werden, bestehen aus Attributen, die nach dem Schema [LDAP-gv.at] modelliert sind.

4.1.1. Metainformationen

Die Metainformation enthält die Versionsnummer. Diese wird aufgrund der aktuellen Implementierung des Clients gesetzt.

4.1.2. Authentifizierungsinformationen

Sind für die Nachvollziehbarkeit der Transaktionen in der Anwendung notwendig. Sie beinhalten Organisation und Organisationseinheit, sowie Identifikationsmerkmale des Benutzers. Informationen sind alle nach dem Schema [LDAP-gv.at] modelliert.

4.1.3. Autorisierungsinformationen

Sind jene Informationen, die benötigt werden, um dem Benutzer die Zugriffsberechtigung zur Anwendung zu erteilen. Die Interpretation der Rechteparameter ist in der Kompetenz der Anwendung.

4.1.4. Chained Token

Der Chained Token ist in einer Kette von Requests der PVP-Token des vorhergehenden Requests, wobei maximal zwei Rekursionsebenen erlaubt sind¹.

4.1.5. Verrechnungsinformationen

Dient der Verrechnung von Transaktionsgebühren. Da weder Kostenstelle noch Gebührenstufe für einen Benutzer immer feststeht, gibt es die Möglichkeit durch das Stammportal Werte vorzugeben.

4.1.6. PVP-Erweiterung

Adressierungsinformation

Unterstützt das Umschreiben von absoluten URLs im Content in Reverse Proxies².

Debughilfe

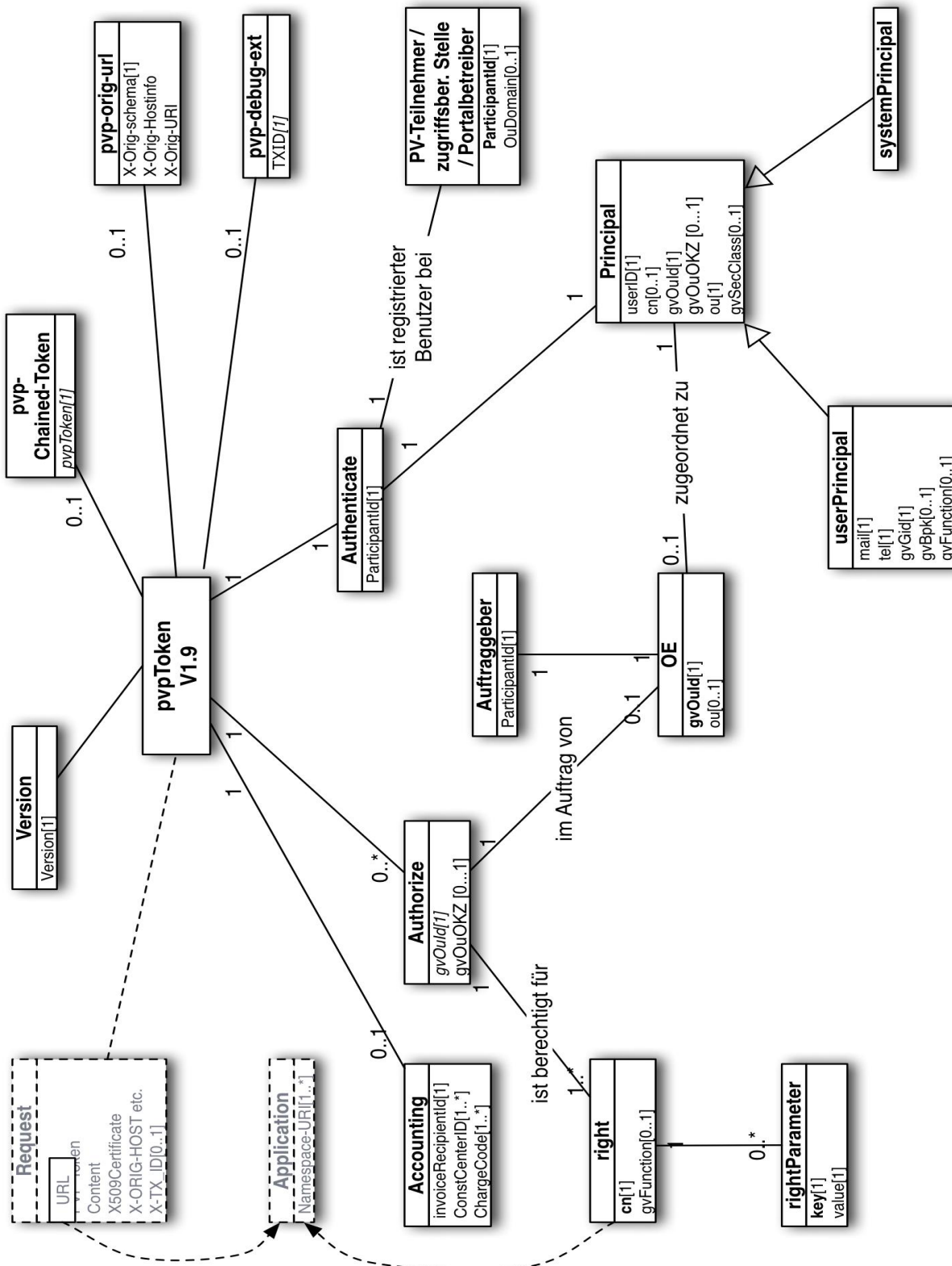
Bei größeren Anwendungsportalen ist das Transaktionsaufkommen so hoch, dass entweder nur sehr wenige Informationen in die Logdatei ausgegeben werden, oder die Suche nach Informationen in einer großen Logdatei aufwändig ist.

Durch eine durchgehende Kennzeichnung der Einträge von Client zu Anwendung wird eine einfache Auswertung auf Session-Ebene möglich.

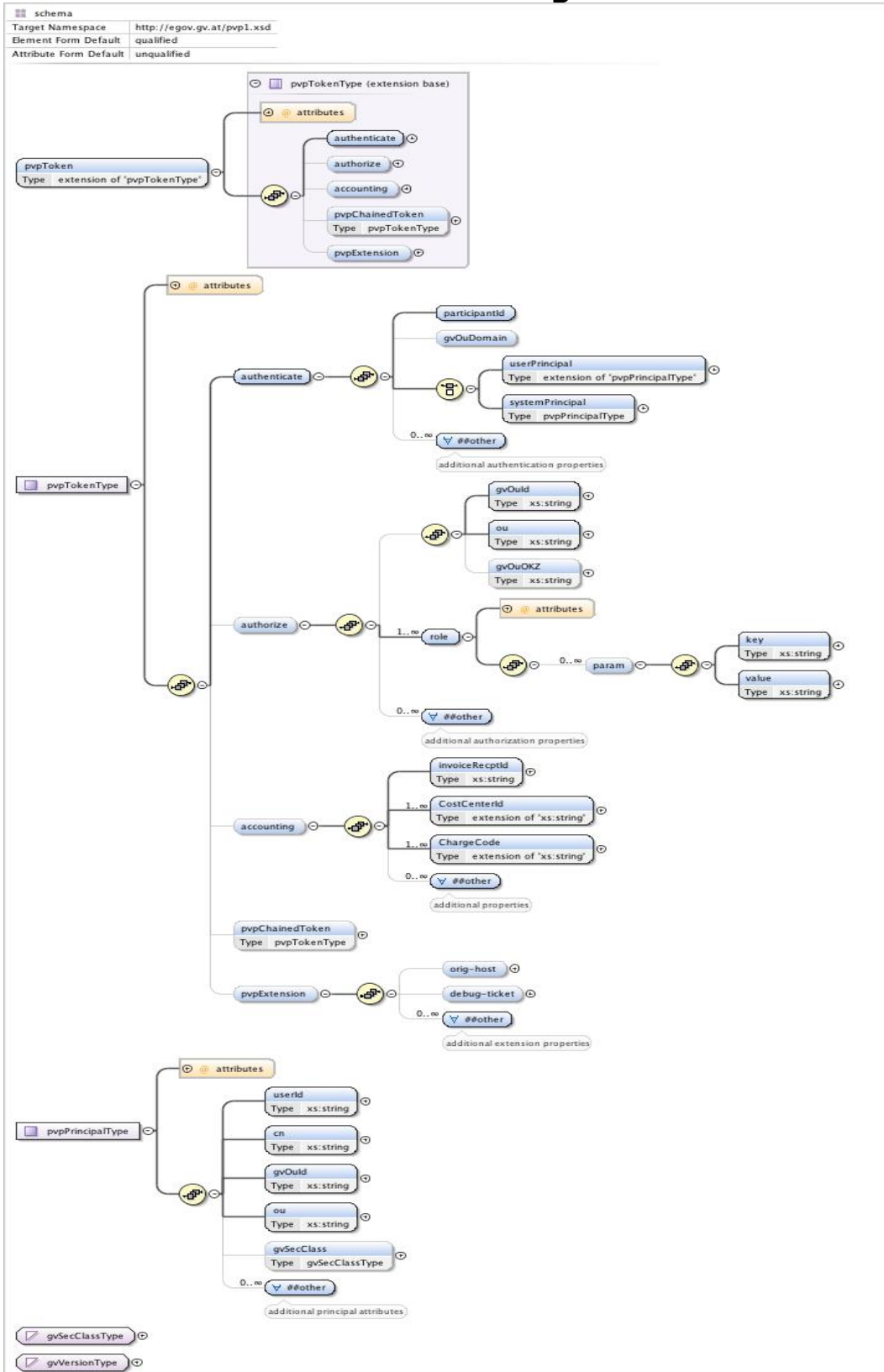
¹Der Chained Token ist als Teil der allgemeinen Spezifikation PVP-Token ausgeführt und daher unabhängig von der Bindung. Ein typischer Anwendungsfall ist, dass ein Benutzer einen HTTP-Request an eine Anwendung sendet, die wiederum einen SOAP-Request an ein Register ausführt.

²Anwendungsbeispiel: ZOPE-Anwendungsserver mit dem „Virtual Host Monster“

4.2. Datenmodell in UML-Notation



4.3. Datenmodell in XML-Darstellung



4.4. Beschreibung der Parameter		
Name	Wert	Länge
Version	die vom Client implementierte PVP-Version. (siehe Kapitel '5 Versionsabstimmung zwischen Client und Server')	4
AUTHENTICATE-...		
participantId	LDAP: gvOrgUnit/GvOuid des Participants, bei dem der Benutzer registriert ist. Siehe Spezifikation [LDAP.gv.at]	39
gvOuDomain	<i>veraltet! Ersetzt durch participantId</i> Organisationsdomäne des Benutzers. Entweder Internet-Domäne (z.B. magwien.gv.at) oder LDAP: Domain/dn bei System-Principals: Organisationsdomäne des Anwendungsverantwortlichen	255
UserID	UserID, mit der der Benutzer am Stammportal authentifiziert ist. (LDAP: gvOrgPerson/uid) oder abgekürzte Bezeichnung des System-Principals in der Form Anwendung.Subsystem.	128
cn	Name des Benutzers (LDAP: gvOrgPerson/cn) oder des System-Principals in der Form Anwendung.Subsystem	64
gvOuid	Stammdienststelle: Eindeutige Kennung (LDAP: gvOrgUnit/gvOuid) der Organisationseinheit des Benutzers, bei System-Principals des Anwendungsverantwortlichen. Die zugehörige Organisationseinheit ergibt sich entweder aus einer eindeutigen Zuordnung oder eine Auswahl des Benutzers, wenn mehrere definiert sind.	32
Ou	Kurzbezeichnung der Organisationseinheit	64
gvOuOKZ	Stammdienststelle: Verwaltungskennzeichen [VKZ] (LDAP: gvOrgUnit/gvOuVKZ) der Organisationseinheit des Benutzers, bei System-Principals des Anwendungsverantwortlichen.	32
gvSecClass	Sicherheitsstufe des Benutzers nach [SecClass] Fehlt dieser Header, wird die Sicherheitsklasse „1“ angenommen.	1

mail	E-Mail-Adresse des Benutzers. Hauptzweck ist die direkte Erreichbarkeit des Benutzers, aber auch die Verwendung in zukünftigen PV-Anwendungen. Die Darstellung ist ohne Display Name und ohne Quotes (also im Format name@domain).	128
tel	Telefonnummer des Benutzers (gvOrgperson/telephoneNumber)	32
gvGid	Global Identifier des Benutzers LDAP: gvOrgPerson/gvGid	128
gvFunction	Entspricht Funktion in gvPersonFunction. Verpflichtend, wenn für eine Person Funktionen definiert sind. LDAP: gvPersonFunction/gvFunction	32
gvBpk	bereichsspezifisches Personenkennzeichen mit einem Präfix bestehend aus „bPK:“, dem Kürzel laut Bereichsabgrenzungsverordnung, „:“ und dem bPK. Im Falle eines verschlüsselten bPK lautet das Präfix nur „vbPK:“ ohne Spezifikation des Bereiches. Beispiel: bPK:PV:NxdRQhp+tNyE9WhHdBSYuy3hA=	256
AUTHORIZE- ...		
gvOuId	Auftraggebende Dienststelle: Inhalt entsprechend AUTHENTICATE-gvOuId	32
Ou	Auftraggebende Dienststelle: Inhalt entsprechend AUTHENTICATE-Ou	64
gvOuOKZ	Auftraggebende Dienststelle: Inhalt entsprechend AUTHENTICATE-gvOuOKZ	32
roles	Anwendungsrechte, optional mit Rechte-Parametern. LDAP: gvApplicationRight (Rechte und Rechte-Parameter, z.B. GKZ, DST, BL, gvOuId) <i>Base64 codierte Binärinformationen als Rechteparameter zu verwenden ist technisch zwar möglich, widerspricht aber dem Konzept einer delegierten Benutzer- und Rechteverwaltung.</i>	32767 ³

³Die Längen der einzelnen Elemente wird im Dokument [AG_IZ_Rechtemodell] limitiert.

ACCOUNTING- ...		
InvoiceRecptId:	Org-ID des Rechnungsempfängers, zur Definition siehe [VKZ]	21
CostCenterId:	<p>Liste der für den Benutzer vorgegebenen Kostenstellencodes. Beispiele:</p> <p><default>ABC123,DEF456 Der Benutzer hat die Kostenstellen ABC123 und DEF456 zur Auswahl, wobei ABC123 der Vorgabewert ist.</p> <p>ABC123 Der Benutzer hat die Kostenstelle ABC123 fix vorgegeben.</p> <p><default>ABC123, DEF456,<user defined> Der Benutzer hat die Kostenstellen ABC123 und DEF456 zur Auswahl, wobei ABC123 der Defaultwert ist. Außerdem kann er weitere Kostestellen frei eingeben.</p>	32767
ChargeCode:	<p>Liste der für den Benutzer vorgegebenen Codes für Transaktionsgebühr, wobei 0 gebührenfrei bedeutet. Beispiele:</p> <p>1 Der Benutzer hat die Gebührenstufe in der Anwendung fix vorgegeben</p> <p>0<default>,1 Der Benutzer hat die Gebührenstufe 0 in der Anwendung vorgegeben, kann aber (über eine Auswahlliste) auch den Wert 1 eingeben.</p>	32
PVP-Extension		
TXID	<p>Um eine einheitliche Kennzeichnung von Einträgen in Stamm-, Anwendungsportal und Anwendung zu ermöglichen, SOLL das Anwendungsportal (oder die Client-Anwendung) pro Request eine eindeutige TransaktionsID erzeugen, und zwar im unten definierten Format. Bei Bedarf kann in der TransaktionsID eine SessionID inkludiert werden, wenn das für die Fehlersuche hilfreich ist.</p> <p>Das Format MUSS entsprechend einer RFC-2822 Message-ID <uniqueID>@<domain> aufgebaut sein, wobei <domain> der Fully-Qualified Host Name (FQHN) des Systems ist, das die TransaktionsID erzeugt. Die <uniqueID> SOLL wie folgt aufgebaut sein:</p> <p><hhmmss+zz>\$<uniquenessString>, wobei die einzelnen Teile wie folgt definiert sind:</p> <p><hhmmss+zz>: Systemzeit mit Zeitzone als Offset zu</p>	256

	UTC <uniquenessString>: String, der die TransaktionsID eindeutig macht (und optional eine bestimmte Sequenz von Transaktionen kennzeichnet) Z.B. „ 235933+01\$1@portal-test.bmi.gv.at“. Die TransaktionsID SOLL kürzer als 40 Zeichen sein, um die Lesbarkeit von Logfiles zu fördern.	
ORIG-...		
SCHEME	Protokollschema (http oder https)	8
HOST	FQHN inklusive :port, wenn nicht der Defaultport verwendet wird. RFC 2109: Fully-qualified host name means either the fully-qualified domain name (FQDN) of a host (i.e., a completely specified domain name ending in a top-level domain such as .com or .uk), or the numeric Internet Protocol (IP) address of a host. The fully qualified domain name is preferred; use of numeric IP addresses is strongly discouraged.	256
URI	Pfadteil des URL mit führenden "/" und ohne Query Parameter	2048

- Die mitgelieferten Benutzerdaten SOLLEN vom Anwendungsportal protokolliert werden, und es SOLL überprüft werden, ob die Rechte des Benutzers in der Menge der für die Organisation gültigen Rechte enthalten ist.

4.5. Mapping der Attribute im PVP-Token

Die Zuordnung der Attribute des PVP-Token für die entsprechende Protokollbindung erfolgt nach den Namen der Attribute, unabhängig von ihrem Pfad. Eine Ausnahme ist Auz-Roles, das bei der HTTP-Bindung in einem String, bei SOAP-Bindung in einer Liste von einzelnen <role>-Elementen mit abweichender Syntax abgebildet wird. Für die einfachere Referenz wird die Zuordnung hier wiedergegeben:

Definition	HTTP-Bindung	SOAP-Bindung (XPath)
Version	X-Version	/pvpToken[@version]
AUTHENTICATE-		
participantId	X-AUTHENTICATE-participantId	authenticate/participantId
gvOuDomain	X-AUTHENTICATE-gvOuDomain	authenticate/gvOuDomain
UserID	X-AUTHENTICATE-UserID	authenticate/*Principal/UserID
cn	X-AUTHENTICATE-cn	authenticate/*Principal/cn
gvOuId	X-AUTHENTICATE-gvOuId	authenticate/*Principal/gvOuId
Ou	X-AUTHENTICATE-Ou	authenticate/*Principal/Ou
gvOuOKZ	X-AUTHENTICATE-gvOuOKZ	authenticate/*Principal/gvOuOKZ
gvSecClass	X-AUTHENTICATE-gvSecClass	authenticate/*Principal/gvSecClass
mail	X-AUTHENTICATE-mail	authenticate/userPrincipal/mail
tel	X-AUTHENTICATE-tel	authenticate/userPrincipal/tel
gvGid	X-AUTHENTICATE-gvGid	authenticate/userPrincipal/gvGid
gvFunction	X-AUTHENTICATE-gvFunction	authenticate/userPrincipal/gvFunction
gvBpk	X-AUTHENTICATE-gvBpk	authenticate/userPrincipal/gvBpk
AUTHORIZE-		
gvOuId	X-AUTHORIZE-gvOuId	authorize/gvOuId
Ou	X-AUTHORIZE-Ou	authorize/Ou
gvOuOKZ	X-AUTHORIZE-gvOuOKZ	authorize/gvOuOKZ
roles	X-AUTHORIZE-roles	authorize/role
ACCOUNTING-		
InvoiceRecptId	X-ACCOUNTING-InvoiceRecptId	accounting/InvoiceRecptId

CostCenterId	X-ACCOUNTING-CostCenterId	accounting/CostCenterId
ChargeCode	X-ACCOUNTING-ChargeCode	accounting/ChargeCode
PVP-Extension		
TXID	X-PVP-TXID	pvpExtension/debug-ticket/txid
ORIG-		
SCHEME	X-ORIG-SCHEME	pvpExtension/orig-host/scheme
HOSTINFO	X-ORIG-HOSTINFO	pvpExtension/orig-host/hostinfo
URI	X-ORIG-URI	pvpExtension/orig-host/uri

4.6. Syntaktische Definition des PVP-Tokens

Diese Definition benutzt eine Produktionsregel in der oben definierten EBNF.

Zur Einschränkung der Zeichensätze werden folgende Basistoken werden definiert:

```

UACHAR := <druckbares US-ASCII Zeichen ohne CRLF (dezimal 33-126)
ILCHAR := <druckbares ISO-Latin Zeichen nach ISO8859-15; decimal 32-126 und
160-255>

pvp-token := pvp-Version pvp-Authentication [pvp-Authorization] [pvp-
Accounting] [pvp-Chained-Token] [pvp-extension]
pvp-Version:= "Version: " pvp-Version-Number
pvp-Version-Number := "1.0" | "1.1" | "1.2" | "1.8" | "1.9"
pvp-Authentication := pvp-Participant pvp-Principal
pvp-Participant := Auth-Participant [Auth-OuDomain]
Auth-OuDomain := "AUTHENTICATE-gvOuDomain: " +UACHAR
Auth-Participant := "AUTHENTICATE-participantId: " +UACHAR
pvp-Principal := Auth-User-Principal | Auth-System-Principal
Auth-User-Principal := Auth-UserId Auth-Cn Auth-Gid Auth-OuId Auth-Ou
[Auth-OuOKZ][Auth-SecClass] Auth-Mail Auth-Tel [Auth-bPK] [Auth-
Function]
Auth-System-Principal := Auth-UserId Auth-Cn Auth-OuId Auth-Ou [Auth-OuOKZ]
[Auth-SecClass]
pvp-Authorization := [Auz-ActingOrg] Auz-Roles
pvp-Accounting := Acc-InvoiceRecptId Acc-CostCenterIdList Acc-ChargeCodeList
pvp-Chained-Token := pvp-token
Auth-UserId := "AUTHENTICATE-userId: " +UACHAR ;
Auth-Cn := "AUTHENTICATE-cn: " +ILCHAR
Auth-Gid := "AUTHENTICATE-gvGid: " +UACHAR
Auth-OuId := "AUTHENTICATE-gvOuId: " +ILCHAR
Auth-Ou := "AUTHENTICATE-ou: " +ILCHAR
Auth-OuOKZ := "AUTHENTICATE-gvOuOKZ: " +ILCHAR
Auth-Function := "AUTHENTICATE-gvFunction: " +ILCHAR
Auth-Mail := "AUTHENTICATE-mail: " +UACHAR
Auth-Tel := "AUTHENTICATE-tel: " +UACHAR
Auth-SecClass := "AUTHENTICATE-gvSecClass: " "0" | "1" | "2" | "3"
Auth-bPK := "AUTHENTICATE-gvBpk: " bPK | vbPK
bPK := "bPK:" Bereichskennzeichen ":" +UACHAR
vbPK := "vbPK:" +UACHAR
Bereichskennzeichen := +UACHAR
Auz-ActingOrg := Auz-OuId Auz-Ou [Auz-OuOKZ]

```

Auz-OuId := "AUTHORIZE-gvOuId: " +UACHAR
Auz-Ou := "AUTHORIZE-ou: " +ILCHAR
Auz-OuOKZ := "AUTHORIZE-gvOuOKZ: " +UACHAR
Auz-Roles⁴ := "AUTHORIZE-Roles: " 1#;Auz-Right["(#,Auz-RoleParameter)"] [;]
Auz-Right := +NAMECHAR
Auz-RoleParameter := Auz-Parameter"=" +VALUECHAR
Auz-Parameter := +NAMECHAR
NAMECHAR := ALPHA | DIGIT | "-" | "_"
VALUECHAR := ILCHAR⁵
Acc-InvoiceRecptId := "ACCOUNTING-InvoiceRecptId: " +UACHAR
Acc-CostCenterIdList := "ACCOUNTING-CostCenterId: " Acc-CostCenterIdItem
Acc-CostCenterIdItem := ["<default>"] Acc-CostCenterIdValue
[,#,Acc-CostCenterIdValue] [,"<user defined>"]
Acc-CostCenterIdValue := {1-25}!ALPHA | DIGIT | SPACE | "-" | "_" | "/"!
Acc-ChargeCodeList := "ACCOUNTING-ChargeCode: " Acc-ChargeCodeItem
Acc-ChargeCodeItem := ["<default>"] Acc-ChargeCodeValue
[,#,Acc-ChargeCodeValue]
Acc-ChargeCodeValue := DIGIT [DIGIT]
pvp-extension := [pvp-orig-url] [pvp-debug-ticket]
pvp-orig-url := X-ORIG-SCHEME X-ORIG-HOSTINFO X-ORIG-URI
X-ORIG-SCHEME := http | https
X-ORIG-HOSTINFO := FQHN [:PORT]
FQHN siehe RFC 2109
PORT: +DIGIT
X-ORIG-URI := +UACHAR
pvp-debug-ticket := PVP-TXID
PVP-TXID := {1-256}UACHAR

⁴Bei SOAP-Bindung wird das Schlüsselwort „roles“ in die Einzahl, also „role“ gesetzt.

⁵Zur Einschränkung der Zeichencodierung bei HTTP-Bindung siehe Abschnitt „Protokollbindung HTTP“

5. Versionsabstimmung zwischen Client und Server

PVP-Version		Dokumentversion
1	0	1.4 (BMI-Gateway-Protokoll)
1	1	1.5.3
1	2	1.6, 1.7
1	8	1.8
1	9	Dieses Dokument

Der Client MUSS die Protokoll-Version senden, die im Client implementiert ist. Der Server MUSS per Default den Fehler 511 setzen, wenn eine höhere Version im Request enthalten ist, als im Server implementiert ist. Wenn ältere Serverversionen in bestimmten Fällen dennoch die höhere Clientversion unterstützen, wird im Server manuell eine Ausnahmeregel für den Client definiert, um die höhere Version zu verarbeiten.

Die erste Stelle der Version gibt die Hauptversion an, nach dem Punkt folgt die Unterversion. Unterversionen des Protokolls sind aufwärtskompatibel.

6. Namensräume für URLs

6.1. Betrieb von Anwendungsportalen als Reverse Proxy

Ein Portal kann als Reverse Proxy mehrere Anwendungen an einem virtuellen Host adressieren. Aus dieser Architektur ergeben sich folgende Konsequenzen:

- Jeder Anwendung wird ein Namensraum innerhalb des virtuellen Hosts zugewiesen, sodass aus dem URL eindeutig die Anwendung abgeleitet werden kann.
- Anwendungen MÜSSEN beachten, dass im Content von HTTP-Responses interne Ressourcen nur über relative URLs ohne Hostnamen adressiert wird, weil sonst der Browser die Anwendung direkt und nicht über das Stammportal adressieren würde. Daher DÜRFEN weder Link noch Base-URL einen Hostnamen enthalten. Ob der Pfadteil des URL absolut (z.B. /at.gv.abcv/xyz/images) oder relativ (.././images) angegeben wird ist unerheblich, solange die Konvention für globale Namensräume eingehalten wird.

6.2. Globale Namensräume für Anwendungen

Um den Betrieb von Stammportalen als Reverse Proxy zu vereinfachen, SOLL das Umschreiben von Pfaden in Stammportalen vermieden werden, indem jede Anwendung einen global eindeutigen Namensraum erhält. Dazu SOLL ab PVP 1.9⁶ folgende Konvention in der Adressierung eingehalten werden:

Der erster Teilstring ohne / ist das Kennzeichen der Anwendung, das mit der Domäne des Anwendungsportals qualifiziert ist, um eindeutige Pfade innerhalb der Portalverbund-Domäne zu gewährleisten. Die Qualifikation erfolgt hierarchisch von links, also z.B. at.gv.xyz.

Die von der Betriebsumgebung (Produktion, Test, ...) und Version abhängige Instanz der Anwendung ist über die Anwendungsbezeichnung zu adressieren.

Beispiel:

<code>https://awp.org-a.gv.at/at.gv.org-a.xyz1-p/</code>	<code># xyz Version 1 Produktion</code>
<code>https://awp.org-a.gv.at/at.gv.org-a.xyz1-t/</code>	<code># xyz Version 1 Test</code>
<code>https://awp.org-a.gv.at/at.gv.org-a.xyz2-t/</code>	<code># xyz Version 2 Test</code>

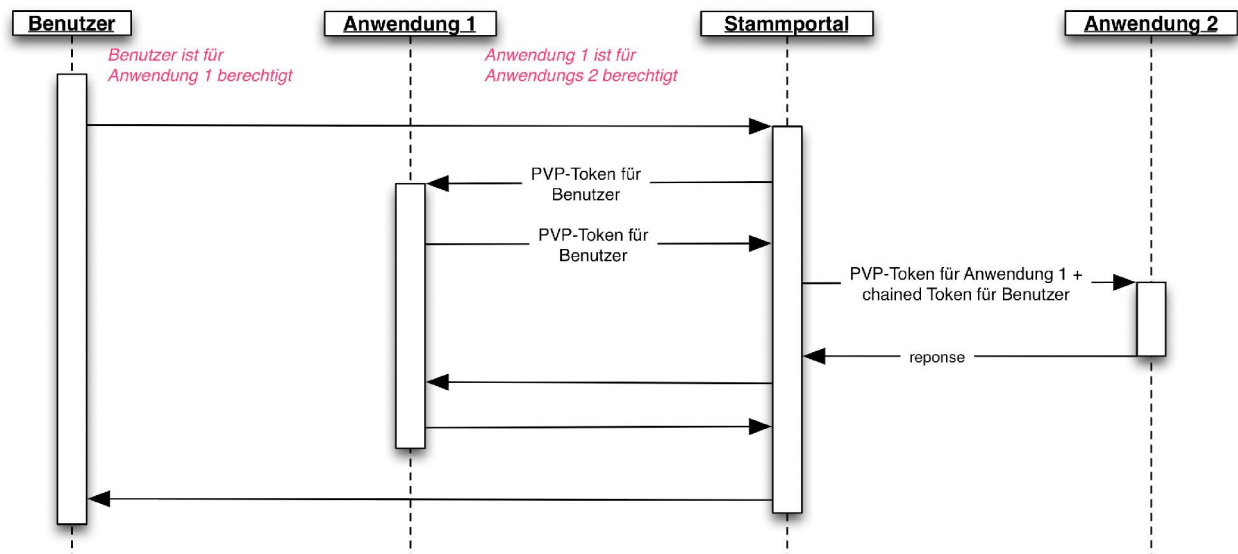
⁶ Davor war die Konvention, dass die Anwendung erst auf der 2. Ebene steht, was aber bei manchen Systemen zu Problemen mit dem Root Context führt. Bestehende Anwendungen brauchen nicht umgestellt werden, da sich die alten und neuen Namensräume nicht überschneiden.

7. Application Chaining

Um bei Zugriffen via Application Chaining die Daten des Endbenutzers protokollieren zu können, muss sein PVP-Token mit den implizit durchgeführten Zugriffen mitgeführt werden. Im Datenmodell ist dafür der Bereich pvpChainedToken vorgesehen.

Die Rechte im pvpChainedToken sollen nicht für die Berechtigungsprüfung herangezogen werden.

Ablauf für einstufiges Application Chaining:



8. Zertifikate

Im Portalverbundsystem sind Verwaltungszertifikate oder kommerzielle Zertifikate registrierter ZDAs zu verwenden. Das Zertifikat identifiziert den Stammportalbetreiber.

Ab PVP-Version 1.8 werden Teilnehmer durch den PVP-Parameter participantId gekennzeichnet. Anwendungsportale SOLLTEN jedoch Stammportale mit vorhergehenden PVP-Versionen unterstützen, wo der Teilnehmer identisch mit dem Stammportalbetreiber ist.

9. Protokollbindung HTTP

In diesem Abschnitt wird definiert, wie das PVP an das HTTP-Protokoll [RFC2616] gebunden wird.

9.1. Abbildung mittels HTTP Header Fields

- Das PVP-Token wird über benutzerdefinierte HTTP-Header mitgegeben.
- Die Namen der HTTP-Header werden mit dem Präfix X- (für benutzerdefinierte Header) versehen.
- In den Parametern von X-AUTHORIZE-roles (VALUECHAR aus EBNF) MÜSSEN folgende Zeichen vom Client codiert und vom Server decodiert werden:
 - , als \,
 - \ als \\
 -) als \)
- Vor und nach Trennzeichen ",;()=" in den Werten der HTTP-Header (z.B. X-AUTHORIZE-roles) SOLLTE Whitespace vermieden werden.
- Die Größe des gesamten HTTP-Headers MUSS kleiner als 64kB⁷ bleiben.
- Wenn die Anwendung Verrechnungsinformationen aus dem PVP-Token übernimmt, MUSS die Anwendung die vom Benutzer eingegebenen Werte als die Cookies `x-gvCostCenterId` und `x-gvChargeCode` übergeben, damit sie für die Portale zur Protokollierung lesbar sind. (Das wird über Scripting im Browser erreicht.)
- Jede HTTP-Transaktion wird für sich authentifiziert, da das HTTP-Protokoll stateless ist. Ein Session-Ticket Mechanismus wie bei Kerberos ist derzeit nicht vorgesehen.⁸

9.2. Fehlermeldungen

- Der Fehlercode wird als HTTP-Code samt zugehörigem Text zurück gegeben

9.3. Application Chaining

- Beim Application Chaining dient die Übergabe der Parameter nicht der Prüfung von Rechten während der Verarbeitung, sondern zur Protokollierung, um später die datenschutzrechtlich verantwortliche Person feststellen zu können.

Bei den vorangegangenen Stufen im Application Chaining reicht es, folgende Parameter zu übergeben:

- VERSION
- AUTHENTICATE-participantId (wenn innerhalb der Kette unterschiedlich)
- AUTHENTICATE-userId
- AUTHENTICATE-gvGid (bei User Principals)
- AUTHENTICATE-cn

⁷ Der Grund für diese Einschränkung ist die Notwendigkeit einen Grenzwert anzugeben, um die Abwehr von DOS-Angriffen auf Server zu unterstützen.

⁸ Um keinen Performance-Nachteil zu erhalten, wird serverseitig ein Caching der Authentifizierungs-Transaktion empfohlen.

- AUTHENTICATE-gvOuid
- AUTHENTICATE-gvOuOKZ
- AUTHENTICATE-ou
- AUTHORIZE-roles
- AUTHORIZE-cn
- AUTHORIZE-gvOuid
- AUTHORIZE-gvOuOKZ

Nach der Ausführung der Produktionsregeln gilt für die HTTP-Bindung:
Bei den Keys, die für das Application Chaining vom vorhergehenden Request übernommen werden, wird "nn-" vorangestellt. Dabei ist nn eine laufende Nummer für die Reihenfolge der Verarbeitung beginnend bei 01 für den Endbenutzer. Die Anzahl der Stufen im Chaining ist auf 2 beschränkt.
D.h., dass der letzte Principal in der Kette (der also die Operation am Anwendungsportal direkt ausführt) immer die Key-Value-Paare ohne laufende Nummer hat, und die zeitlich vorhergehenden Schritte beginnend beim Endbenutzer durchnummeriert werden.⁹

Im Anhang ist ein (nicht normnatives) Beispiel angegeben.

10. Protokollbindung SOAP

Die Protokollbindung für SOAP verwendet und erweitert die Spezifikation Web Services Security [WS-Security].

10.1. Namespaces

Folgende Namespaces werden in diesem Dokument verwendet:

Prefix	Namespace
P	http://egov.gv.at/pvp1.xsd
S	http://schemas.xmlsoap.org/soap/envelope/ Zur Rückwärtskompatibilität mit SOAP 1.1 muss auch folgender Namespace akzeptiert werden: http://www.w3.org/2001/12/soap-envelope
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wsswssecurity-secext-1.0.xsd Zur Rückwärtskompatibilität muss auch folgender Namespace akzeptiert werden: http://schemas.xmlsoap.org/ws/2002/04/secext

Es ist die Verantwortung des Anwendungsportals die Kompatibilität zur Anwendung herzustellen. Die Unterstützung von 2 WSSE-Namensräumen in PVP 1.9 kann z.B. durch Umschreiben des WSSE-Namensraums erfolgen.

⁹ Dadurch wird eine Rückwärtskompatibilität mit Anwendungsportalen erreicht, die die zusätzlichen Parameter des Application Chainings nicht protokollieren (PVP 1.5)

10.2. Security Element pvpToken

Das <wsse:security> Element ist im Header-Block, um sicherheitsrelevante Informationen zur Nachricht zu übergeben. Es wird zur Übergabe der PVP-Parameter an ein Anwendungsportal um das Element <P:pvpToken> erweitert. Die in 3. Architektur definierten Bereiche der Keys sind im Gegensatz zur HTTP-Bindung nicht im Namen jedes Elements enthalten, sondern werden zur hierarchischen Gliederung der Elemente in <authorize>, <authenticate> und <accounting> verwendet. Die Struktur ist im [PVP-Schema] beschrieben.

10.3. Zeichencodierung

Zeichenketten sind XML-konform zu codieren. Das Encoding SOLL mittels UTF-8 erfolgen. Das gilt auch für PVP-Token, die von der HTTP-Bindung auf SOAP-Bindung konvertiert werden; dabei müssen sämtliche Codierungen der HTTP-Bindung auf die XML-Bindung umgesetzt werden.

10.4. Fehlermeldungen

PVP-Fehlercodes MÜSSEN als SOAP Faults zurückgegeben werden.

Namespace= *urn:PVPservices*

Der Faultcode setzt sich aus den Zeichen "F" und dem Fehlercode (siehe Abschnitt 11 Fehlermeldungen) zusammen.

Zusätzlich werden für die SOAP-Bindung folgende Fehlermeldungen definiert:

Faultcode	Faultstring	Beschreibung
F480	Invalid SOAP Header	Kein gültiger SOAP-Header gefunden
F481	Missing Security (WS-Security) token in SOAP-Header	Die PVP-Parameter müssen in einem WSSE:Security-Token eingebettet sein
F482	Missing pvpToken	Kein pvpToken gefunden
F483	Invalid XML	Fehlerhaftes XML

11. HTTP Fehlermeldungen

Fehler-Code	Beschreibung
402	Für diese Funktion ist eine Verrechnung erforderlich, aber das Header-Feld XXXX fehlt (XXXX ist eines aus ACCOUNTING-gvInvoiceRecptId, ACCOUNTING-gvCostCenterId oder ACCOUNTING-gvChargeCode)
440	Mandatory PVP-Header XXXX fehlt
441	Werte in AUTHORIZE-roles haben ungültiges Format
442	Kein zulässiges Recht in AUTHORIZE-roles
443	Die UserId ist am Anwendungsportal gesperrt
444	Stammportal ist für Anfragen des angegebenen Participants nicht berechtigt
445	Participant am Anwendungsportal nicht registriert
450	ACCOUNTING-InvoiceRecptId: ungültiger Wert oder Verrechnungskonto gesperrt.
451	Ungültiger Wert für ACCOUNTING-ChargeCode.
462	Sicherheitsklasse (gvSecClass) muss mindestens 2 sein
463	Sicherheitsklasse (gvSecClass) muss 3 sein
482	PvpToken fehlt
490	Zertifikatsüberprüfung fehlgeschlagen. Grund: XXXXXXXXXXXXXXXXXXXX (z.B.: ungültige Root-CA, Zertifikat abgelaufen, Zertifikat nicht beim Portal registriert)
491	HTTP wird nicht unterstützt – es muss HTTPS verwendet werden
492	Keine Berechtigung für diese Anwendung im Anwendungsportal definiert
493	Keine Berechtigung für diese Anwendung im Stammportal
494	Die Authentifizierung des Stammportals ist fehlgeschlagen
496	Applikation ist nicht online
511	PVP-Version nicht unterstützt

Fehlerbedingungen sind im Text möglichst detailliert zu beschreiben, etwa durch die Referenz des betroffenen Headers und die Art der Bedingung (z.B. „Header `Version` fehlt“, „Wert für `gvSecClass` zu groß“)

Anhang A HTTP Beispiel-Request User Principal

[Beispiele sind nicht normativ. Bitte zur Entwicklung immer das Datenmodell benutzen.]

Beispiel für einen HTTP Header bei einem Request eines Stammportals ohne Verrechnungsdaten:

```
POST /abc.gv.at/anwendung1/servlet/ HTTP/1.1
Host: awp.abc.gv.at
Accept-Encoding: gzip, deflate
User-Agent: Mozilla (5.0 Linux)
X-Version: 1.9
X-AUTHENTICATE-participantId: AT:L6:1234789
X-AUTHENTICATE-UserId: mmustermann@kommunalnet.at
X-AUTHENTICATE-cn: Max Mustermann
X-AUTHENTICATE-gvOuId: AT:GGA-60420:0815
X-AUTHENTICATE-Ou: Meldeamt
X-AUTHENTICATE-gvOuOKZ: AT:GGA-60420-Abt13
X-AUTHENTICATE-gvSecClass: 2
X-AUTHENTICATE-gvGid: AT:B:0:LxXnvpcYZesiqVXsZG0bB==
X-AUTHENTICATE-mail: max.mustermann@hatzendorf.steiermark.at
X-AUTHENTICATE-tel: +43 3155 5153
X-AUTHENTICATE-gvFunction: SB
X-AUTHORIZE-roles: Beispielrolle(GKZ=60420,GKZ=62031,GKZ=62032,
                    GKZ=62010,GKZ=62008,GKZ=62023)
Content-Type: application/x-www-form-urlencoded
Content-Length: 788
```

In diesem Fall ist der Benutzer berechtigt, das Recht Beispielrolle mit den Parametern GKZ=60420, 62031, 62032, 62010, 62008 und 62023 auszuüben. Die Interpretation der Rollenparameter liegt bei der Anwendung. In diesem Beispiel wird unterstellt, dass die Anwendung den Benutzer die Funktionen der Beispielrolle auf die angeführten Gemeinden ausführen lässt.

Anhang B HTTP Beispiel-Request System Principal

[Beispiele sind nicht normativ. Bitte zur Entwicklung immer das Datenmodell benutzen.]

Die Organisation des Benutzers ist in dem Beispiel zwei Mal enthalten: Das Tupel `x-AUTHENTICATE-gvOuid/-ou` definiert die organisatorische (personelle) Zugehörigkeit des Benutzers. Im Falle eine System Principals ist das der datenschutzrechtliche Auftraggeber der Anwendung. `AUTHORIZE-gvOuid/-ou` geben an, im Auftrag welcher Organisation der Benutzer die Transaktion durchführt.

```
POST /abc.gv.at/anwendung2/xyz HTTP/1.1
Host: gondor.wien.gv.at
User-Agent: .JNET 1.1
X-Version: 1.9
X-AUTHENTICATE-participantId: AT:L9:MA2412
X-AUTHENTICATE-UserId: omr-appuser@wien.gv.at
X-AUTHENTICATE-cn: OMR
X-AUTHENTICATE-gvOuid: AT:L9:0014
X-AUTHENTICATE-Ou: MA14
X-AUTHENTICATE-gvOuidOKZ: AT:L9-MA14
X-AUTHENTICATE-gvSecClass: 2
X-AUTHORIZE-roles: Beispielrolle
X-AUTHORIZE-gvOuid: AT:L9:0018
X-AUTHORIZE-ou: Bürgerdienst
X-AUTHORIZE-gvOuidOKZ: AT:L9:MA55
Content-Type: text/xml
Content-Length: 788
```

Anhang C Beispiel mit Application Chaining

[Beispiele sind nicht normativ. Bitte zur Entwicklung immer das Datenmodell und XML-Schema in der Beilage benutzen.]

Request vom Browser zum Stammportal:

```
POST /abc.gv.at/anwendung1/servlet/ HTTP/1.1
Host: stp.intra.xyz.gv.at
Accept-Encoding: gzip, deflate
User-Agent: Mozilla (5.0 Linux)
Content-Type: application/x-www-form-urlencoded
Content-Length: 788
```

Request vom Stammportal zur Anwendung 1:

```
POST /abc.gv.at/anwendung1/servlet/ HTTP/1.1
Host: awp.abc.gv.at
Accept-Encoding: gzip, deflate
User-Agent: Mozilla (5.0 Linux)
X-Version: 1.9
X-AUTHENTICATE-participantId: AT:L6:1234789
X-AUTHENTICATE-UserId: mmustermann@kommunalnet.at
X-AUTHENTICATE-cn: Max Mustermann
X-AUTHENTICATE-gvGid: AT:B:0:LxXnvpcYZesiqVXsZG0bB==
X-AUTHENTICATE-gvOuId: AT:GGA-60420:0815
X-AUTHENTICATE-Ou: Meldeamt
X-AUTHENTICATE-gvOuOKZ: AT:GGA-60420-Abt13
X-AUTHENTICATE-mail: max.mustermann@hatzendorf.steiermark.at
X-AUTHENTICATE-tel: +43 3155 5153
X-AUTHENTICATE-gvSecClass: 2
X-AUTHORIZE-roles: Beispielrolle(GKZ=60420)
Content-Type: application/x-www-form-urlencoded
Content-Length: 788
```

Request von Anwendung 1 an das Stammportal:

Hierbei handelt es sich um das gleiche PVP-Token, das die Anwendung 1 vom Stammportal bekommen hat, konvertiert auf die SOAP-Bindung.

(Für die bessere Lesbarkeit mit Default-Namespace für PVP-Elemente)

```
<pvpToken version="1.9" xmlns="http://egov.gv.at/pvpl.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://egov.gv.at/pvpl.xsd pvpl.xsd">
  <authenticate>
    <participantId>AT:L6:1234789</participantId>
    <userPrincipal>
      <userId>mmustermann@kommunalnet.at</userId>
      <cn>Max Mustermann</cn>
      <gvOuid>AT:GGA-60420:0815</gvOuid>
      <ou>Meldeamt</ou>
      <gvOuOKZ>AT:GGA-60420-Abt13</gvOuOKZ>
      <gvSecClass>2</gvSecClass>
      <gvGid>AT:B:0:LxXnvpcYZesiqVXsZG0bB==</gvGid>
      <mail>max.mustermann@hatzendorf.steiermark.at</mail>
      <tel>+43 3155 5153</tel>
    </userPrincipal>
  </authenticate>
  <authorize>
    <role value="Beispielrolle">
      <param>
        <key>GKZ</key>
        <value>60420</value>
      </param>
    </role>
  </authorize>
</pvpToken>
```

Request vom Stammportal an Anwendung 2:

(Für die bessere Lesbarkeit mit Default-Namespace für PVP-Elemente)

```
<pvpToken version="1.9" xmlns="http://egov.gv.at/pvpl.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://egov.gv.at/pvpl.xsd pvpl.xsd">
  <authenticate>
    <participantId>AT:L6:1234789</participantId>
    <systemPrincipal>
      <userId>egovstar.appserv1.intra.xyz.gv.at</userId>
      <cn>Anwendung 1 Register-Interface</cn>
      <gvOuid>AT:L6:4711</gvOuid>
      <ou>Fachabteilung 1B Informationstechnik</ou>
      <gvOukZ>AT:L6-FA1B</gvOukZ>
      <gvSecClass>2</gvSecClass>
    </systemPrincipal>
  </authenticate>
  <authorize>
    <role value="Registerabfrage"></role>
  </authorize>
  <pvpChainedToken version="1.9">
    <authenticate>
      <participantId>AT:L6:1234789</participantId>
      <userPrincipal>
        <userId>mmustermann@kommunalnet.at</userId>
        <cn>Max Mustermann</cn>
        <gvOuid>AT:GGA-60420:0815</gvOuid>
        <ou>Meldeamt</ou>
        <gvOukZ>AT:GGA-60420-Abt13</gvOukZ>
        <gvSecClass>2</gvSecClass>
        <gvGid>AT:B:0:LxXnvpcYZesiqVXsZG0bB==</gvGid>
        <mail>max.mustermann@hatzendorf.steiermark.at</mail>
        <tel>+43 3155 5153</tel>
      </userPrincipal>
    </authenticate>
    <authorize>
      <role value="Beispielrolle">
        <param>
          <key>GKZ</key>
          <value>60420</value>
        </param>
      </role>
    </authorize>
  </pvpChainedToken>
</pvpToken>
```

Anhang D SOAP Beispiel-Request

[Beispiele sind nicht normativ. Bitte zur Entwicklung immer das XML-Schema in der Beilage benutzen.]

```
<S:Envelope
  xmlns:P="http://egov.gv.at/pvpl.xsd"
  xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wsswssecurity-secext-1.0.xs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://egov.gv.at/pvpl.xsd pvpl.xsd
    http://schemas.xmlsoap.org/soap/envelope/ soap-envelope.xsd">
  <S:Header>
    <wsse:Security>
      <P:pvptoken version="1.9">
        <P:authenticate>
          <P:participantId>AT:L6:994</P:participantId>
          <P:userPrincipal>
            <P:userId>fmeier@stmk.gv.at</P:userId>
            <P:cn>F. Meier</P:cn>
            <P:gvOuid>AT:L6:4711</P:gvOuid>
            <P:ou>Fachabteilung 1B Informationstechnik</P:ou>
            <P:gvOukZ>AT:L6-FA1B</P:gvOukZ>
            <P:gvSecClass>1</P:gvSecClass>
            <P:gvGid>AT:B:0:Uh05RG++klaOTsVY+CU=</P:gvGid>
            <P:mail>franz.meier@stmk.gv.at</P:mail>
            <P:tel>+43 316 8771234</P:tel>
          </P:userPrincipal>
        </P:authenticate>
        <P:authorize>
          <P:gvOuid>AT:L6:1010</P:gvOuid>
          <P:ou>Steirische Bezirkshaputmannschaften</P:ou>
          <P:gvOukZ>AT:L6BH</P:gvOukZ>
          <P:role value="ZMR-Fremdenbehoerdenanfrage">
            <P:param>
              <P:key>GKZ</P:key>
              <P:value>60000</P:value>
            </P:param>
          </P:role>
        </P:authorize>
      </P:pvptoken>
    </wsse:Security>
  </S:Header>
  <S:Body>
  </S:Body>
</S:Envelope>
```

Anhang E Beispiele für Rechte und Rechteparameter

Durch die zweidimensionale Darstellung von Rechten können die meisten Berechtigungssysteme mit angemessenem Aufwand abgebildet werden. Rechte an Anwendungen sind hier im Allgemeinen bereits aggregierte Einzelrechte, die in der Literatur oft als Rollen bezeichnet werden.

Einfaches Berechtigungsschema

Recht 1: Anwendungsadministrator

Recht 2: Sachbearbeiter

Recht 3: Abfrageberechtigter

Komplexeres Berechtigungsschema

Recht 1: Anwendungsadministrator

Recht 2: Sachbearbeiter (OE)

Recht 3: Abfrageberechtigter

In diesem Fall dürfen Sachbearbeiter nur für bestimmte Organisationseinheiten Geschäftsfälle erledigen. Dafür wird eine Liste von OE übergeben, für die der Benutzer die Rechte hat.

Das Modell ist auch für andere Einschränkungen oder explizite Berechtigungen einer Rolle anwendbar, etwa nach geografischen Gesichtspunkten.

Anhang F Implementierungshinweise für Reverse Proxies

a. Problemstellung

Ein Portal wird im Portalverbundprotokoll der Version 1.x im Allgemeinen als Gateway im Sinne der HTTP-Spezifikation [RFC 2616] implementiert, was auch als "Reverse Proxy" bezeichnet wird¹⁰. Aus der Sicht des HTTP-Clients ist ein Gateway keine Zwischenstation, sondern der endgültige Kommunikationspartner. Dem entsprechend sind Namensraum und Adressierung auf das Portal bezogen. Ein PV-Portal unterscheidet sich von einem gewöhnlichen Reverse Proxy durch zwei wesentliche Merkmale: Die Funktion als Authentifizierungs- und Autorisierungsproxy einerseits und das Mapping von URLs auf mehrere Anwendungsportale bzw. Anwendungen andererseits. URL-Mapping bedeutet, dass der Pfad-Teil des URLs entscheidet, zu welchem Server ein Request weiter geleitet wird. Dadurch entsteht am Reverse Proxy ein gemeinsamer Namensraum der Anwendungen. Diese Funktion ist verantwortlich für ein spezielles Problem bei der Verarbeitung von URLs, das hier besprochen werden soll.

b. Umschreiben von HTTP-Headern durch einen Reverse Proxy

- HTTP-Header, die URL-Teile enthalten müssen korrekt umgeschrieben werden:
 - SET-COOKIE muss umgeschrieben werden, wenn die Attribute PATH oder DOMAIN gesetzt sind.
 - HOST ist immer auf den Host umzuschreiben, auf den der Request weiter geleitet wird.
 - LOCATION ist auf den Hostnamen des Portals (oder Clients) zu setzen, das den zum Redirect-Response gehörigen Request erzeugt hat.
- Cookies aller Anwendungen des Stammportals haben einen gemeinsamen Namensraum für das Path-Attribut. Sollte das zu Problemen¹¹ führen, können zusätzliche Virtual Hosts die Namensräume trennen. Im Allgemeinen sind Cookies wie folgt umzuschreiben:

¹⁰ Die Implementierung als Reverse Proxy in Portalen ist allerdings nur eine Konvention. Eine Implementierung als Forward Proxy über das HTTP-Proxy Protokoll wäre eleganter und würde die Probleme der Namensräume lösen. Allerdings geht das im Falle von Stammportalen nur bei Open-Source Proxies (Squid, Delegated), und nicht für proprietäre Produkte von MS/Novell/Sun etc.

¹¹ Namensraumkonflikte können z.B. entstehen, wenn zwei Java-Anwendungen JSESSIONID Cookies verwenden, und in unterschiedlichen Application Servern ausgeführt werden, und dadurch innerhalb der selben Browser-Instanz die Cookies gegenseitig überschrieben werden.

- *Domain-Attribut*
Wird das Domain-Attribut im Set-Cookie Header nicht gesetzt, dann braucht es vom Reverse Proxy nicht umgeschrieben werden. Andernfalls müssen Cookie-Domains bei Responses so umgeschrieben werden, dass sie bei einem darauf folgenden Request einerseits vom Browser an das Portal übergeben werden und andererseits das Stammportal die Domäne wieder korrekt zurücksetzen kann. Dafür sind wiederum die Regeln des URL-Mappings anzuwenden.
- *Path-Attribut*
Das Path-Attribut ist analog zum Domain-Attribut zu verarbeiten.
- *Lokale Cookies im Stammportal*
Wenn im Stammportal ein Cookie erzeugt wird, etwa JSESSIONID zur Verwaltung der Benutzersession, und für den gleichen URL von der Anwendung ein Cookie des gleichen Namens erzeugt wird, müssen die Cookies durch unterschiedliche PATH-Attribut qualifiziert werden, etwa indem für das Stammportal-Cookie explizit PATH=/ gesetzt wird. Alternativ kann ein eindeutiger Name für das Cookie (z.B. XXX.GV.AT-SESSIONID) verwendet werden.

Anhang G Debugschnittstelle für Clients

Auf der Seite des Anwendungsportals soll das Logging nach folgenden Anforderungen gestaltet werden:

- a) Falls der Server auf mehrere Nodes verteilt ist, sollen diese Zusatzlogs auf einem Server konsolidiert werden (über syslog etc.). Die Netzwerk-Requests dafür können dabei geblockt werden, weil die exakte Reihenfolge der Cluster-Nodes nicht so wichtig ist. Für jeden Cluster in sich stimmen sie immer.
- b) Für jedes Ticket werden die Log-Einträge mit dem Ticket-Namen gekennzeichnet, z.B., indem eine eigene Datei erstellt wird.
- c) Die Logeinträge werden dem User an einem URL zugänglich gemacht, wobei der Ticketname gleichzeitig die Authentifizierung für den Zugriff beinhaltet. Das Verzeichnis mit der Logdatei darf dafür nicht aufgelistet werden können. Der User muss daher den Ticket-Namen wissen, um das Logfile zu sehen. Das Ticket soll die Form JJJJMMTTxxxxxxxxxxxxxxxxxxxx haben, wobei xxxxxxxxxxxxxxxxxxxx eine längere mindestens 10-stellige Zufallszahl, somit schwer zu erraten ist. Eine Authentifizierung mittels PVP erfolgt nicht.

Anhang H Referenzen

[AG_IZ_Rechtemodell]

Stradal, Freidl, Gritschenberger, Pichler, Reif: Rechtemodellierung für Portalverbundanwendungen

<http://www.ref.gv.at> -> Portalverbund

[LDAP.gv.at]

Spitzenberger, Martin: Spezifikation LDAP-gv.at V. 2.4.0

<http://www.ref.gv.at> -> Portalverbund

[PortalV-PKI]

<http://portal.bmi.gv.at/ref/> -> PKI

[PV-DASI]

Connert: Datensicherheitsmaßnahmen für Webanwendungen

<http://reference.e-government.gv.at>

[PV-Whitepaper]

Hörbe, Werzowa: Portal Verbund Whitepaper 2005-02-17

<http://reference.e-government.gv.at> -> Portalverbund

[PVP-Schema]

Hörbe, Liehmann: XML-Schema für den Portalverbund: pvp1.xsd (2005-02-01)

<http://reference.e-government.gv.at> -> Portalverbund

[PVV 1.0]

Connert, Grandits, Kotschy, Posch, Siegl: Vereinbarung über die einzuhaltenden Rahmenbedingungen bei der Einrichtung und Benützung eines E-Government Portalverbundsystems (21.11.2002)

<http://reference.e-government.gv.at> – Empfehlungen

[RFC2047]

RFC 2047 - MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text

<http://www.ietf.org/rfc/rfc2047.txt>

[RFC2616]

R. Fielding & al.: Hypertext Transfer Protocol -- HTTP/1.1

<http://www.ietf.org/rfc/rfc2616.txt>

[SecClass]

Hörbe: Sicherheitsklassen im Portalverbund-System

<http://reference.e-government.gv.at> – Empfehlungen

[VKZ]

Grandits: Verwaltungskennzeichen:

<http://reference.e-government.gv.at> Dokument VKZ 1.2.0

[WS-Security]

Specification: Web Services Security, Version 1.0 05 April 2002

<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

Anhang I Änderungen von Version 1.9.1 zu 1.9.2

- [4.2] Grafik aktualisiert
- [4.3] Grafik aktualisiert
- [4.4] VERSION: Fußnote gestrichen – Verweis auf Kapitel 5 eingefügt.
- [4.4] AUTHENTICATE-gvOuid geändert (wegen Abwärtskompatibilität)
- [4.4] Neu: AUTHENTICATE-gvOuOKZ, AUTHORIZE-gvOuOKZ.
- [4.4] Fehlender Prefix gv bei SecClass hinzugefügt.
- [4.5] Neu: AUTHENTICATE-gvOuOKZ, AUTHORIZE-gvOuOKZ
- [4.6] Neu: AUTHENTICATE-gvOuOKZ, AUTHORIZE-gvOuOKZ
- [9.1] Entfernen der Codierungsregeln für Space und Strichpunkt
- [9.3] Neu: AUTHENTICATE-gvOuOKZ, AUTHORIZE-gvOuOKZ
- [10.1]: Ergänzungen
- [Anhänge A-D]: Beispiele angepasst
- [Anhang G] Layout korrigiert

Anhang J Änderungen von Version 1.9.0 zu 1.9.1

- [4.1] Referenz auf LDAP-gv.at-Spezifikation eingefügt
- [4.1.3] Autorisierungsinformationen: ALLOW/DENY bei Rechteparametern entfernt, ist Anwendungskompetenz und nicht Aufgabe der Protokollspezifikation.
- [4.1.6, 4.4, 4.5, 4.6, Anhang G] Loglevel aus PVP-Token entfernt (nicht Teil der Protokollspezifikation, gehört zum Debug-Ticket)
- [4.2 und 7] Grafiken ohne Transparenz eingefügt, damit kein schwarzer Ausdruck entsteht
- [4.2] Inkonsistenz in UML-Darstellung: Attribute mail/tel als Pflichtfeld gesetzt
- [4.2] pvpExtAttr in UML-Darstellung entfernt (überflüssig)
- [4.2, 4.4] gvUserRestriction aus Diagramm und Tabelle entfernt (obsolet)
- [4.4] Rechteparameter: Charset ist auf ISO-Latin beschränkt
- [4.4] AUTHORIZE-gvOuid, -ou: Beschreibung des Inhalts nur als Referenz zu den entsprechenden Attributen in AUTHENTICATE-gvOuid, -ou
- [4.4] ParticipantID: Referenz nicht auf Kapitel 8, sondern auf die Spezifikation LDAP-gv.at
- [4.4, 4.5, 4.6] inkonsistente Spezifikation des Attributs gvBpk: fehlenden Präfix „gv“ ergänzt (PVP1.xsd ist korrekt und bleibt unverändert)

- [4.4] Hinweis auf die Beschränkungen der Längen der einzelnen Elemente von AUTHORIZE-roles in [AG_IZ_Rechtemodell]
- [4.6] Korrektur pvp-chained-token: Anpassung an XML-Schema
- [9.1] EBNF-Syntax: CR, LF aus Escape-Set entfernt (war nicht falsch, aber überflüssig, weil nicht im Wertevorrat enthalten)
- [Anhänge A-D] -gvOuID, -ou in den Beispielen konsistent zur Spezifikation gemacht.
- Anhang H: Referenz auf die LDAP-gv.at-Spezifikation aufgenommen

Anhang AA Änderungen von Version 1.8.10 zu 1.9.0

Einbeziehung der Technical Notes

- Application Chaining
- Debug-Schnittstelle
- Orig-Header
- Reverse-Proxy / cookies Rewrite

Korrekturen und Ergänzungen

- Begriffsbestimmungen für Subteilnehmer und Participant
- PVP-Token um PVP-Extensions (Debug, Orig-Header) erweitert
- Begriff PVP-Parameters durch PVP-Token und PVP-Chained-Token ersetzt
- Namensraum für SOAP-Request an SOAP 1.2 angepasst
- Anhang C Beispiel Application Chaining überarbeitet
- Anhang D SOAP Beispiel-Request: Namespaces „P“ und „wsse“ korrigiert
- Abbildungsregel für Chained Token von EBNF-Definition nach „HTTP-Bindung“ verschoben
- Zeichensätze auf „printable ohne CRLF“ eingeschränkt; OCTET wird als ISO8859-15 definiert, UTF-8 mittels MIME Message Header Extensions (RFC 2047). Die Auswahl dieser Codierungen, die jeder mit PVP 1.9 konforme Client verarbeiten können muss, verbessert die Interoperabilität bei gleichzeitiger Unterstützung von Zeichen, die nicht von ISO-8859-15 abgebildet werden.
- PVP-Token um Authenticate-gvBPK¹² erweitert
- Attribute um Längenangabe erweitert
- Mapping-Tabelle für Attribute (HTTP und SOAP) erstellt
- Globaler Namensraum für Anwendungen (im Root-Kontext)

¹²Mit PVP 1.9.1; in PVP 1.9.0 war die Spezifikation inkonsistent

- Der Satz „HTTP MUSS mit TLS oder SSL3.0 gesichert werden, wobei Client-Zertifikate verpflichtend sind“ wird gestrichen, weil die Sicherheitsanforderungen im Dokument SecClass spezifiziert werden.
- Fehlermeldungen 450, 451, 492 und 496 ergänzt.

Bedeutungsänderung im PVP-Token

Der Inhalt der Attribute gvOuid und ou wurde geändert. Die Änderung ist dann wirksam, wenn auf Grund dieser Attribute am Anwendungsportal Rechte geprüft werden.

Dokumentstruktur: Gegenüberstellung 1.8 zu 1.9

1. Zweck	1. Zweck
2. Begriffsbestimmung	2. Begriffsbestimmung 2.1. Vergleiche zu anderen Nomenklaturen 2.2. Schreibweise der EBNF
	3. Architektur 3.1. Entitäten der Authentifizierung und Autorisierung 3.2. Message Sequence
3. Grundbestandteile des PVP 3.1. Metainformationen 3.2. Authentifizierungsinformationen 3.3. Autorisierungsinformationen 3.4. Verrechnungsinformationen	4. Datenmodell des PVP-Tokens 4.1. Hauptbestandteile des PVP-Tokens 4.2. Datenmodell in UML-Notation 4.3. Datenmodell in XML-Darstellung 4.4. Beschreibung der Parameter
4. Schreibweise (-> neu in 2.2.)	4.5. Mapping der Attribute des PVP-Token
5. Grammatik des Portalverbund-Protokolls	4.6. Definition des PVP-Tokens
6. Beschreibung der Parameter	
6.1. Versionsabstimmung zwischen Client und Server	5. Versionsabstimmung zwischen Client und Server
	6. Namensräume für URLs 6.1. Betrieb von Anwendungsportalen als Reverse Proxy 6.2. Globale Namensräume für Anwendungen
7. Protokollbindung HTTP	8. Protokollbindung HTTP
8. Protokoll-Bindung PVP – SOAP 8.1. Namespaces 8.2. Security Element pvpToken 8.3. Fehlermeldungen	9. Protokollbindung SOAP 9.1. Namespaces 9.2. Security Element pvpToken 9.3. Fehlermeldungen
9. Zertifikate	10. Zertifikate

10. Fehlermeldungen	11. Fehlermeldungen
Anhang A HTTP Beispiel-Request User Principal	Anhang A HTTP Beispiel-Request User Principal
Anhang B HTTP Beispiel-Request System Principal	Anhang B HTTP Beispiel-Request System Principal
Anhang C.	Anhang C Beispiel mit Application Chaining
Anhang D SOAP Beispiel-Request	Anhang D SOAP Beispiel-Request
Anhang E Beispiele für Rechte und Rechteparameter	Anhang E Beispiele für Rechte und Rechteparameter
Anhang E Beispiele für Rechte und Rechteparameter Anhang F Implementierungshinweise a. Betrieb von Anwendungsportalen als Reverse Proxy b. Betrieb von Stammportalen als Reverse Proxy c. Verifikation von Client-Zertifikaten (Anhang F Implementierungshinweise für Reverse Proxies a. Problemstellung b. Umschreiben von HTTP-Headern durch einen Reverse Proxy
Anhang G Referenzen	Anhang G Debugschnittstelle für Clients
Anhang H Funktionelle Änderungen von Version 1.7 zu 1.8	Anhang H Referenzen
	Anhang I Änderungen von Version 1.8.10 zu 1.9.0