Portalverbundprotokoll Version 2 **R-Profil**		**Konvention** **PVP2-R-Profil 2.1.2** **Empfehlung**
Kurzbeschreibung	Das R-Profil beschreibt die von PVP Version 1.x übernommenen Protokollbindungen für HTTP und SOAP in einer Reverse-Proxy-Architektur.	
Autor(en):	Peter Pfläging (Wien) Rainer Hörbe (Wien)	Projektteam / Arbeitsgruppe **AG Integration und Zugänge (AG-IZ)** AG-Leiter: Michael Pellmann (Wien) Stellvertreter: Dominik Klauser (BKA)
Beiträge von:	Harald Stradal, Peter Pichler, Michael Werzowa, Wolfgang Kremser, Joachim Minichshofer und anderen	

Version 2.1.2 : **1.6.2015**	Angenommen: **14.12.2015 VSt-1712/531**
Version 2.1.1: **20.4.2015**	Angenommen: **10.11.2015 VSt-1712/526**
Version 2.1.0: **26.9.2013**	Angenommen: **21.11.2013 VSt-1712/488**
Version 2.0.0.a : **31.8.2011**	Angenommen: **14.10.2011 VST-1712/455**

Inhaltsverzeichnis

1 Zertifikate

Im Portalverbundsystem identifizieren Zertifikate Stammportalbetreiber. Die jeweilige Stammorganisation[1] der Benutzer wird durch den PVP-Parameter participantId gekennzeichnet.

[1] Die Stelle die für die Verwaltung des Benutzers zuständig ist. Kann von der zugriffsberechtigten Stelle abweichen, wenn der Benutzer im Auftrag einer anderen Organisation handelt.

2 Protokollbindung HTTP

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

2.1 Abbildung mittels HTTP Header Fields

- Der *PVP-eGovToken* wird über benutzerdefinierte HTTP-Header mitgegeben.
- Die Größe des gesamten HTTP-Headers MUSS kleiner als 64kB[2] bleiben.
- Wenn die Anwendung Verrechnungsinformationen aus dem *PVP-eGovToken* übernimmt, MUSS die Anwendung die vom Benutzer eingegebenen Werte als die Cookies `X-PVP-COST-CENTER-ID` und `X-PVP-CHARGE-CODE` 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.3

2.2 Fehlermeldungen

Der Fehlercode wird als HTTP-Code samt zugehörigem Text zurückgegeben.

2.3 Reverse Proxy: Anforderungen an PVP-Anwendungen

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. (siehe unten)
- Anwendungen MÜSSEN beachten, dass im Content von HTTP-Responses interne Ressourcen nur über relative URLs ohne Hostnamen adressiert werden, 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) angegebenen wird ist unerheblich, solange die Konvention für globale Namensräume eingehalten wird.
- Nicht-ASCII-Zeichen und das kaufmännische „Und" (&) werden mithilfe von SGML Numeric Character References dargestellt (Numeric Character References können auch in HTML und XML verwendet werden).

2.4 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

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

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

Anwendung einen global eindeutigen Namensraum erhält. Dazu SOLL ab PVP 1.9[4] 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:
https://awp.org-a.gv.at/at.gv.org-a.xyz1-p/https://awp.org-a.gv.at/at.gv.org-a.xyz1-p/

xyz Version 1 Produktion
https://awp.org-a.gv.at/at.gv.org-a.xyz1-t/https://awp.org-a.gv.at/at.gv.org-a.xyz1-thttps://awp.org-a.gv.at/at.gv.org-a.xyz1-t//

xyz Version 1 Test
https://awp.org-a.gv.at/at.gv.org-a.xyz2-t/https://awp.org-a.gv.at/at.gv.org-a.xyz2-t/

xyz Version 2 Test

Einschränkung im Namensraum des Cookiepfads
Die Anwendung soll die Cookie Domain nicht setzen. Wenn der Cookie Path gesetzt wird, SOLL dieser nicht außerhalb vom definierten Namensraum (nicht im „*Context Root*“) liegen.

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

3 Protokollbindung SOAP

In PVP 1.x gab es einen eigenen PVP Dialekt für SOAP Webservice Anfragen.
Mit PVP 2 wurde diese PVP Variante gestrichen und festgelegt, dass für SOAP Anfragen das HTTP-R Profil verwendet werden soll.
Gründe: Mit dem R-Profil können alle Webserice Varianten unterstützt werden (z.B. auch REST). Im Portalverbund werden SOAP Anfragen ausschließlich über HTTPS verschickt. Sofern eine SOAP Anwendung technische Probleme damit hat HTTP Header auszulesen (Hauptargument für die Einführung des SOAP Bindings in PVP 1.x), soll das in der Sphäre des Anwendungsverantwortlichen gelöst werden. (am Anwendungsportal oder in der Anwendung selbst)).

4 Fehlermeldungen

Allgemeine PVP Fehlermeldungen für HTTP- und SOAP-Bindung:

Fehler-Code	Beschreibung
402	Für diese Funktion ist eine Verrechnung erforderlich, aber das Header-Feld XXXX fehlt (XXXX ist eines aus X-PVP-INVOICE-RECPT-ID, X-PVP-COST-CENTER-ID oder X-PVP-CHARGE-CODE)
440	Mandatory PVP-Header XXXX fehlt
441	Werte in X-PVP-ROLES haben ungültiges Format
442	Kein zulässiges Recht in X-PVP-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	X-PVP-INVOICE-RECPT-ID: ungültiger Wert oder Verrechnungskonto gesperrt.
451	Ungültiger Wert für X-PVP-CHARGE-CODE.
462	Sicherheitsklasse muss mindestens 2 sein
463	Sicherheitsklasse muss 3 sein
482	PVP-eGovToken fehlt
483	Falsche Protokollbindung
490	Zertifikatsüberprüfung fehlgeschlagen. Grund: XXXXXXXXXXXXXXXX (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-EgovToken-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 X-PVP-EGOVTOKEN-VERSION fehlt", „Wert für SecClass zu groß")

5 Beispiele

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

5.1 HTTP Beispiel-Request User Principal

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-PVP-EGOVTOKEN-VERSION: 2.0
X-PVP-BINDING: http
X-PVP-PARTICIPANT-ID: AT:L6:1234789
X-PVP-USERID: mmustermann@kommunalnet.at
X-PVP-PRINCIPALNAME: Mustermann
X-PVP-GIVENNAME: Max
X-PVP-OU-GV-OU-ID: AT:GGA-60420:0815
X-PVP-OU: Gemeinde Musterdorf
X-PVP-OU-OKZ: AT:GGA-60420-Abt13
X-PVP-SECCLASS: 2
X-PVP-GID: AT:B:0:LxXnvpcYZesiqVXsZG0bB==
X-PVP-MAIL: max.mustermann@musterdorf.gv.at
X-PVP-tel: +43 3155 5153
X-PVP-FUNCTION: SB
X-PVP-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.

5.2 HTTP Beispiel-Request System Principal

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

```
POST /abc.gv.at/anwendung2/xyz HTTP/1.1
Host: host.wien.gv.at
User-Agent: .JNET 1.1
X-PVP-EGOVTOKEN-VERSION: 2.0
X-PVP-BINDING: http
X-PVP-PARTICIPANT-ID: AT:L9:MA2412
X-PVP-USERID: omr-appuser@wien.gv.at
X-PVP-PRINCIPALNAME: OMR
X-PVP-OU-GV-OU-ID: AT:L9:0014
X-PVP-OU: MA14
X-PVP-OU-OKZ: AT:L9-MA14
X-PVP-SECCLASS: 2
X-PVP-ROLES: Beispielrolle
Content-Type: text/xml
Content-Length: 788
```

6 Implementierungshinweise für Reverse Proxies

6.1 Problemstellung

Ein Portal wird im PVP2 R-Profil im Allgemeinen als Gateway im Sinne der HTTP-Spezifikation [RFC 2616] implementiert, was auch als "Reverse Proxy" bezeichnet wird. 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.

6.2 Umschreiben von HTTP-Headern durch einen Reverse Proxy

HTTP-Header, die URL-Teile enthalten müssen korrekt umgeschrieben werden, wenn die dazugehörige Abbildungsregel für URLs umkehrbar ist:

a) SET-COOKIE muss umgeschrieben werden, wenn die Attribute PATH oder DOMAIN gesetzt sind.
b) HOST ist immer auf den Host umzuschreiben, auf den der Request weiter geleitet wird.
c) 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 Problemen[5] führen, wird der Einsatz des S-Profils empfohlen. Im Allgemeinen sind Cookies wie folgt umzuschreiben:

a) 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.
b) Path-Attribut
   Das Path-Attribut ist nach den Abbildungsregeln für URLs umzuschreiben.

6.3 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=/

[5] 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.

gesetzt wird. Alternativ sollte ein eindeutiger Name für das Cookie (z.B. XXX.GV.AT-SESSIONID) verwendet werden.

6.4 Einschränkung im Namensraum des Cookiepfads

Die Anwendung soll die Cookie Domain nicht setzen. Wenn der Cookie Path gesetzt wird SOLL dieser nicht außerhalb vom definierten Namensraum (nicht im Context Root) liegen. Andernfalls SOLL ein eindeutiger Name für das Cookie (z.B. XXX.GV.AT-SESSIONID) verwendet werden.

Änderungshistorie

Version 2.1.1:

- Die V1 Definition für PVP SOAP Webservices wurde entfernt. Im Zuge dessen wurde auch Beispiele, die sich auf PVP SOAP bezogen haben und der für PVP SOAP eingeführte http Header x-pvp-binding wieder entfernt.

Version 2.1.2:

- Synchronisierung der Versionsnummer mit dem Dokumentenset PVP 2.1.2
- Keine Änderung des Inhalts.