Errata der Spezifikationen zur österreichischen Bürgerkarte		Konvention - Empfehlung
Kurzbeschreibung	Das vorliegende Dokument ist eine Sammlung von bekannten Errata in den Spezifikationen zur österreichischen Bürgerkarte ab der Version 1.1.0.	
Autoren:	Arno Hollosi Gregor Karlinger Thomas Rössler Martin Centner et al.	Projektteam/Arbeitsgruppe AG Bürgerkarte
Datum:	1.3.2005	

Inhaltsverzeichnis

1. Einleitung

Vorbemerkung: Zur besseren Lesbarkeit wurde in diesem Dokument auf geschlechtsneutrale Formulierungen verzichtet. Die verwendeten Formulierungen richten sich jedoch ausdrücklich an beide Geschlechter.

Dieses Dokument listet die bekannten Errata in den Spezifikationen zur österreichischen Bürgerkarte ab der Version 1.1.0.

1.1. Geltung dieses Dokuments

Die in diesem Dokument gelisteten Errata beziehen sich auf sämtliche Spezifikationsdokumente zur österreichischen Bürgerkarte, im Einzelnen sind das:

- Einführung
- Applikationsschnittstelle Security-Layer
- Transportprotokolle Security-Layer
- Minimale Umsetzung des Security-Layers
- Standardisierte Key- und Infoboxen
- Fehlercodes Security-Layer
- Zugriffschutz
- Anforderungen an die Benutzer-Schnittstelle
- Standard-Anzeigeformat

Mit dem Erscheinen eines der Spezifikationsdokumente mit einer höheren Versionsnummer werden die bis zu diesem Erscheinungsdatum gelisteten Errata

in die aktualisierte Spezifikation eingearbeitet, sämtliche Errata bleiben jedoch in diesem Dokument weiter gelistet.

Sobald ein Erratum in dieses Dokument eingetragen wurde, gilt er - falls anwendbar - im Sinne der im Eintrag angeführten Korrektur als behoben.

1.2. Struktur eines Erratum-Eintrags

Jeder Eintrag in Abschnitt 2 dieses Dokuments enthält die folgenden Informationen:

- Spezifikationsdokument, Versionsnummer und Ort des Auftretens innerhalb des Dokuments;
- eine eindeutige Referenznummer des Eintrags;
- das Datum der Aufnahme des Erratums in dieses Dokument;
- den Berichter des Erratums;
- eine Klassifikation des Erratums:
  - Fehler (falsche oder in sich widersprüchliche Aussage);
  - Unklarheit (missverständliche Aussage oder Aussage mit Interpretationsspielraum);
  - Editorialer Fehler (z. B. Tippfehler, Broken Link, ...);
- eine Beschreibung sowie ggf. eine Korrektur des Erratums;
- die Versionsnummer des betroffenen Spezifikationsdokuments ab welcher der Erratum korrigiert ist.

1.3. Kontakt für Erratum-Bericht

Wenn Sie einen Erratum in einem der Spezifikationsdokumente entdecken, schicken Sie bitte einen Bericht per Email an die in der Dokumenteninformation genannten Autoren.

2. Indizes der Errata

2.1. Nach Berichtszeitpunkt

Dieser Abschnitt listet die Errata nach dem Zeitpunkt ihrer Aufnahme in dieses Dokument.

2.2. 2.2 Nach zugehörigem Spezifikationsdokument

Dieser Abschnitt listet die Errata nach den zugehörigen Spezifikationsdokumenten.

- Erratum 2 (21. 02. 2003)
- Erratum 4 (21. 02. 2003)
- Erratum 5 (21. 02. 2003)
- Erratum 6 (21. 02. 2003)
- Erratum 7 (21. 02. 2003)
- Erratum 8 (05. 03. 2003)
- Erratum 9 (05. 03. 2003)
- Erratum 10 (05. 03. 2003)
- Erratum 11 (24. 03. 2003)
- Erratum 12 (24. 03. 2003)
- Erratum 13 (24. 03. 2003)
- Erratum 14 (24. 03. 2003)
- Erratum 15 (24. 03. 2003)
- Erratum 17 (07. 05. 2003)
- Erratum 18 (02. 09. 2003)
- Erratum 20 (06. 10. 2003)
- Erratum 21 (29. 06. 2004)
- Erratum 22 (29. 06. 2004)
- Erratum 23 (30. 07. 2004)
- Erratum 26 (08. 02. 2005)
- Erratum 27 (08. 02. 2005)
- Erratum 29 (08. 02. 2005)
- Erratum 32 (01. 03. 2005)

- Transportprotokolle Security-Layer
  - Erratum 16 (24. 03. 2003)
  - Erratum 19 (04. 09. 2003)
  - Erratum 31 (01. 03. 2005)
- Minimale Umsetzung des Security-Layers
  - Erratum 3 (21. 02. 2003)
- Standardisierte Key- und Infoboxen
-
  - Erratum 25 (11. 02. 2005)
  - Erratum 30 (01. 03. 2005)
- Fehlercodes
- Zugriffsschutz
- Anforderungen an die Benutzer-Schnittstelle
-
  - Erratum 28 (11. 02. 2005)
- Standard-Anzeigeformat
  - Erratum 24 (16. 09. 2004)

3. Liste der Errata

3.1. Errata in "Einführung"

Derzeit sind sind keine Fehler bekannt.

3.2. Errata in "Applikationsschnittstelle Security-Layer"

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.1, Informationen zum Datenobjekt, Datenobjekt, Absatz n der Tabelle**
Bericht	**Referenznummer**	1	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Das Inhaltsmodell von `sl10:XMLContent` ist so definiert, dass es eine beliebige Mischung aus Text und XML-Markup erlaubt. Das schließt ausdrücklich auch reinen Text mit ein. Eine gültige Instanz von `sl10:XMLContent` ist also beispielsweise auch `<sl10:XMLContent>Text</sl10:XMLContent>`.			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.1, Informationen zum Datenobjekt, Datenobjekt, Tabelle, F A und B**
Bericht	**Fehlernummer**	2	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			

Beschreibung	Die Einbindung eines im Fall A oder B übergebenen Datenobjekts in die XML-Signatur bzw. die Referenzierung auf dieses in die Sig eingebundene Datenobjekt aus dem zugehörigen `dsig:Reference` Element der XML-Signatur muss so erfolgen, dass ausschließlich die im Re in `sl10:DataObject` übergebenen Daten signiert werden. Wird beispielsweise das Datenobjekt als Inhalt eines `dsig:Object` Elements in die XML-Signatur eingebunden, darf dieses `dsig:Ob` Containerelement nicht mitsigniert werden, sondern lediglich sein Inhalt.

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.2, Implizite Transformationsparameter, 4. Absatz**
Bericht	**Fehlernummer**	4	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Stefan Grill, für die BRZG
Klassifikation	Unklarheit			
Beschreibung	Der letzte Satz im vierten Satz lässt nicht erkennen, dass sich die darin erwähnte Auflösung der Referenz auf den impliziten Transformationspara auf den Fall der Signaturprüfung bezieht. Er sollte also verbessert lauten: "Das Attribut URI eines Referenzobjekts enthält dabei die Referenz auf den impliziten Transformationsparameter, und zwar in exakt gleicher W wie sie von der Bürgerkarten-Umgebung im Falle der Signaturprüfung zur Auflösung verwendet werden würde."			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.1, Informationen zum Datenobjekt, Datenobjekt, Tabelle**
Bericht	**Fehlernummer**	5	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Tabelle zeigt nur die gültigen Kombinationen aus dem Wert des Attributs `Structure`, dem Wert des Attributs `Reference`, sowie dem Inha Element `sl10:DataObject`. Alle anderen Kombinationen sind ungültig.			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.2, Signaturattribute**
Bericht	**Fehlernummer**	6	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Um den Vorgaben des XML-Schemas für die ETSI-Attribute hinsichtlich des Elements etsi:SignedSignatureProperties zu genügen, müssen nebe eigentlich geforderten Attributen für die signierten Metainformationen sowie für die signierte Zertifikatsreferenz zwei weitere Attribute in die Sig integriert werden: • `etsi:SigningTime`: Dieses Attribut enthält den Zeitpunkt der Signaturerstellung. • `etsi:SignaturePolicyIdentifier`: Dieses Attribut enthält Angaben über die Signatur-Policy, die der Signatur zu Grunde liegt. Es empfohlen, als Inhalt dieses Attributs das Element `etsi:SignaturePolicyImplied` zu verwenden, um anzuzeigen, dass die unterzeich Daten die Signatur-Policy implizieren.			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.2, Implizite Transformationsparameter, 3. Absatz**
Bericht	**Fehlernummer**	7	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Fehler			
Beschreibung	Die Definition eines implizten Transformationsparameter lautet laut Absatz 2: "Ein impliziter Transformationsparameter ist in diesem Zusammer ein Datenobjekt, das von der Bürgerkarten-Umgebung zur Berechnung der Transformationen verwendet wurde, jedoch nicht explizit als Paramet entsprechenden Transformationsobjekt (dsig:Transform) aufscheint." Der erste Satz des Absatzes 3 ist deshalb als falsch einzustufen. Die Referenzeingangsdaten bilden die Eingangsdaten zur Berechnun Transformationen, sind aber keine Parameter. Die Referenzeingangsdaten brauchen daher nicht in das Signaturmanifest inkludiert zu werden.			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.1, Informationen zum Datenobjekt, Datenobjekt, Tabell Fall A**
Bericht	**Fehlernummer**	8	**behoben ab Version**	1.2.0
	am	05. 03. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Es wird nicht spezifiziert, wie das Datenobjekt in die Signatur eingebunden werden soll, wenn es als `sl10:Base64Content` vorliegt. Gerad diesen Fall ist aber eine genaue Vorgabe angebracht, da es beim Einbinden des dekodierten Inhalts von `sl10:Base64Content` zu Fehlern kor kann. Es wird also folgende Ergänzung vorgenommen: "Ist das Datenobjekt Base64-kodiert (Verwendung des Elements `sl10:Base64Content` in `sl10:DataObject`), MUSS es in dieser Ba kodierten Form in die Signaturstruktur eingebunden werden. Signiert werden MUSS jedoch das Base64-dekodierte Datenobjekt; es ist daher in d das eingebundene Datenobjekt verweisenden dsig:Reference eine Base64 Transformation zu verwenden."			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.1, Informationen zum Datenobjekt, Datenobjekt, Tabell Fall B**
Bericht	**Fehlernummer**	9	**behoben ab Version**	1.2.0
	am	05. 03. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Es wird nicht spezifiziert, wie das Datenobjekt in die Signatur eingebunden werden soll, wenn es sich beim referenzierten Datenobjekt nicht um X Daten handelt. Gerade für diesen Fall ist aber eine genaue Vorgabe angebracht, da es beim Einbinden von beliebigen Stream-Daten in die Signat Fehlern kommen kann. Es wird also folgende Ergänzung vorgenommen: "Handelt es sich beim referenzierten Datenobjekt nicht um XML-Daten, oder wird das Datenobjekt entgegen der Empfehlung nicht auf Vorliege XML-Daten geprüft, MUSS es in Base64-kodierter Form in die Signaturstruktur eingebunden werden. Signiert werden MUSS jedoch ursprüngliche Datenobjekt; es ist daher in der auf das eingebundene Datenobjekt verweisenden dsig:Reference eine Base64 Transformatio verwenden."			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.2, Implizite Transformationsparameter**
Bericht	**Fehlernummer**	10	**behoben ab Version**	1.2.0
	am	05. 03. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Durch die mittels Errata 7 eingeführte Änderung kann es nun vorkommen, dass keine impliziten Transformationsparameter vorliegen. In so Fällen ist überhaupt von der Erstellung eines Signaturmanifests Abstand zu nehmen. Der Abschnitt "Implizite Transformationsparameter" wird um folgenden Satz ergänzt: "Liegen keine impliziten Transformationsparameter vor, DARF das Signaturmanifest NICHT erstellt werden."			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2.2.2, Implizite Transformationsparameter**
Bericht	**Fehlernummer**	11	**behoben ab Version**	1.2.0
	am	24. 03. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Spezifikation lässt offen, was genau die Bürgerkarten-Umgebung als Eingangsdaten für die Berechnung des Hash-Wertes über einen impl Transformationsparameter verwenden muss. Der Abschnitt "Implizite Transformationsparameter" wird daher um folgenden Absatz ergänzt: "Die Eingangsdaten für die Berechnung des Hash-Wertes über einen implizten Transformationsparameter ergeben sich wie folgt: • Wird der implizite Transformationsparameter als Referenz angegeben, die von der Bürgerkarten-Umgebung selbst aufzulösten ist, ist zwi einer externen und einer internen Referenz zu unterscheiden: o Die Auflösung einer externen Referenz MUSS einen Byte-Stream liefern. Dieser Byte-Stream bildet die Eingangsdaten für die Berechnung. o Die Auflösung einer internen Referenz MUSS eine XPath-Knotenmenge liefern (vgl. [XMLDSIG, Abschnitt 4.3.3.3]). Knotenmenge ist nach [C14N] zu kanonisieren, um einen eindeutigen Byte-Stream zu erhalten. Dieser Byte-Stream bildet dan Eingangsdaten für die Hash-Berechnung. • Wird der implizite Transformationsparameter als Referenz angegeben, die von der Bürgerkarten-Umgebung nicht selbst aufzulösen ist, w der Anfrage ein entsprechendes Ergänzungsobjekt angegeben wurde, ist wiederum zwischen einer externen und einer internen Refere unterscheiden: o Enthält das Ergänzungsobjekt Daten für eine externe Referenz, MÜSSEN diese Daten als Base64 (`sl10:Supplement/sl10:Content/sl10:Base64Content`) vorliegen. Die Base64-dekodierten Daten bilden dann die Eingangs für die Hash-Berechnung. o Enthält das Ergänzungsobjekt Daten für eine interne Referenz, MÜSSEN diese Daten als XML (`sl10:Supplement/sl10:Content/sl10:XMLContent`) vorliegen. Aus diesen Daten ist entsprechend [XMLDSIG, Abschnitt 4. eine XPath-Knotenmenge zu erzeugen. Diese Knotenmenge ist nach [C14N] zu kanonisieren, um einen eindeutigen Byte-Strea erhalten. Dieser Byte-Stream bildet dann die Eingangsdaten für die Hash-Berechnung." In Abschnitt 9 der Spezifikation ist weiters folgende Referenz aufzunehmen: "C14N Boyer, John: Canonical XML. W3C Recommendation, März 2001. Abgerufen aus dem World Wide Web am 31. August 2002 http://www.w3.org/TR/2001/REC-xml-c14n-20010315."			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 3.1.2, Prüfung der Signaturprüfdaten**
Bericht	**Fehlernummer**	12	**behoben ab Version**	1.2.0
	am	24. 03. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Spezifikation lässt offen, ob die Prüfung der Signaturprüfdaten durchgeführt werden muss, wenn die bei der Prüfung der Gültigkeit der Sig ein Fehler aufgetreten ist. Der erste Absatz dieses Abschnittes wird daher um folgenden Satz ergänzt: "Die Prüfung der Signaturprüfdaten DARF unterbleiben, wenn bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist." Weiters wird die Tabelle in diesem Abschnitt um folgenden Eintrag erweitert: " **99** **Die Prüfung der Signaturprüfdaten wurde nicht durchgeführt, da bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten** "			

Ort	Version	1.1.0	Auftreten im Dokument	Abschnitt 3.2.2, Prüfung des Signaturmanifests
Bericht	Fehlernummer	13	behoben ab Version	1.2.0
	am	24. 03. 2003	von	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Spezifikation lässt offen, ob die Prüfung des Signaturmanifests durchgeführt werden muss, wenn die bei der Prüfung der Gültigkeit der Sig ein Fehler aufgetreten ist. Der erste Absatz dieses Abschnittes wird daher um folgenden Satz ergänzt: "Die Prüfung des Signaturmanifests DARF unterbleiben, wenn bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist." Weiters wird die Tabelle in diesem Abschnitt um folgenden Eintrag erweitert: " **99 Die Prüfung des Signaturmanifests wurde nicht durchgeführt, da bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten** "			

Ort	Version	1.1.0	Auftreten im Dokument	Abschnitt 3.2.2, Prüfung weiterer Manifeste
Bericht	Fehlernummer	14	behoben ab Version	1.2.0
	am	24. 03. 2003	von	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Spezifikation lässt offen, ob die Prüfung weiterer Manifeste durchgeführt werden muss, wenn die bei der Prüfung der Gültigkeit der Signatu Fehler aufgetreten ist. Der erste Absatz dieses Abschnittes wird daher um folgenden Satz ergänzt: "Die Prüfung weiterer Manifeste DARF unterbleiben, wenn bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist." Weiters wird die Tabelle in diesem Abschnitt um folgenden Eintrag erweitert: " **99 Die Prüfung weiterer Manifeste wurde nicht durchgeführt, da bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist.** "			

Ort	Version	1.1.0	Auftreten im Dokument	Abschnitt 2.2.1, Informationen zum Datenobjekt, Ergänzungsobjekte
Bericht	Fehlernummer	15	behoben ab Version	1.2.0
	am	24. 03. 2003	von	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Spezifikation macht zu wenig deutlich, dass das Datenobjekt selbst nicht als Ergänzungsobjekt übergeben werden darf, sondern dafü Container `sl10:DataObject` zu verwenden ist. Der erste Absatz dieses Abschnittes wird daher um folgenden Satz ergänzt: "Das Datenobjekt selbst DARF jedoch NICHT als Ergänzungsobjekt übergeben werden."			

Ort	Version	1.1.0	Auftreten im Dokument	Abschnitt 7, Beispiel
Bericht	Fehlernummer	17	behoben ab Version	1.2.0
	am	07. 05. 2003	von	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Editorialer Fehler			

Beschreibung	Das angeführte Beispiel für eine ErrorResponse entspricht nicht dem XML-Schema. Das Element, das als Inhalt den Fehlercode aufweist, wi Beispiel irrtümlich mit <Code> bezeichnet, laut Schema müsste es aber <ErrorCode> heißen. Korrekter Weise sieht das Beispiel also wie folgt au
	"
	`<sl10:ErrorResponse>`
	`  <sl10:ErrorCode>1152</sl10:ErrorCode>`
	`  <sl10:Info>Read Access denied.</sl10:Info>`
	`</sl10:ErrorResponse>`
	Beispiel: Fehler-Antwort auf die Anfrage zum Lesen des Inhalts einer Infobox
	"			

Ort	Version	1.1.0	Auftreten im Dokument	Abschnitt 3.2.2, Beispiel
Bericht	Fehlernummer	18	behoben ab Version	1.2.0
	am	02. 09. 2003	von	Markus Punz, BDC
Klassifikation	Editorialer Fehler			
Beschreibung	Das angeführte Beispiel für eine `VerifyXMLSignatureResponse` entspricht nicht dem XML-Schema. Für das Element `CertificateCheck` v mit `sl10` irrtümlich der falsche Namenraum-Präfix verwendet. Der korrekte Namenraum-Präfix lautet `sl11`.			

Ort	Version	1.1.0	Auftreten im Dokument	Abschnitt 3.2.2, Prüfung des Signaturmanifests
Bericht	Fehlernummer	20	behoben ab Version	1.2.0

	am	06. 10. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle

Klassifikation	Editorialer Fehler

Beschreibung

Durch die mittels Errata 7 eingeführte Änderung kann es nun vorkommen, dass keine impliziten Transformationsparameter vorliegen, und somi Signatur, die dieser Spezifikation genügt, kein Signaturmanifest beinhalten muss. Die Tabelle, welche die Bedeutung der Prüfcodes festschreibt, daher wie folgt geändert:

`sl11:Code`	**Bedeutung**
0	Dieser Code hat eine der folgenden Bedeutungen: • Für diese Signatur ist kein Signaturmanifest notwendig. • Die Signatur enthält eine Referenz auf das notwendige Signaturmanifest. Das Signaturmanifest entspricht vom Umfang her Anforderungen dieser Spezifikation. Für jede `dsig:Reference` des Signaturmanifests konnte der Hash-Wert erfolgr überprüft werden.
1	Die Signatur enthält keine Referenz auf das notwendige Signaturmanifest.
2	Die Signatur enthält zwar eine Referenz auf das notwendige Signaturmanifest, dieses entspricht vom Umfang her jedoch nicht den Anforderungen dieser Spezifikation. Die Hash-Werte der im Signaturmanifest vorhandenen `dsig:Reference`-Elemente wurden ni überprüft.
3	Die Signatur enthält eine Referenz auf das notwendige Signaturmanifest. Das Signaturmanifest entspricht vom Umfang her den Anforderungen dieser Spezifikation. Bei der überprüfung des Hash-Werts zumindest einer `dsig:Reference` des Signaturmanifests jedoch ein Fehler aufgetreten.
99	Die Prüfung eines gegebenenfalls notwendigen Signaturmanifests wurde nicht durchgeführt, da bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist.

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**XML-Schema Core-1.2.xsd**

Bericht	**Fehlernummer**	21	**behoben ab Version**	1.2.1
	am	29. 06. 2004	**von**	Patrick Peck, Anecon

Klassifikation	Editorialer Fehler

Beschreibung

Die Schema-Definition für das Element `sl:FriendlyName` im Datentyp `sl:VerificationResultType` hat keinen Datentyp spezifiziert. D wird nun mit `xsd:string` festgelegt. Somit lautet die korrigierte Schema-Definition:

```
<xsd:element name="FriendlyName" type="xsd:string" minOccurs="0"/>
```

Analoges gilt für das Element `sl:FriendlyName` im Datentyp `sl:VerifyHashInfoRequestType`. Dafür lautet die korrigierte Schema-Defi somit:

```
<xsd:element name="FriendlyName" type="xsd:string" minOccurs="0"/>
```

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**XML-Schema Core-1.2.xsd; Abschnitt 1.2, Tabelle**

Bericht	**Fehlernummer**	22	**behoben ab Version**	1.2.1
	am	29. 06. 2004	**von**	Patrick Peck, Anecon

Klassifikation	Fehler

Beschreibung

Der mit der Version 1.2.0 eingeführte Versionierungsmechanismus über das Attribut ProtocolVersion in den einzelnen Befehlsanfragen ansta bisherigen Versionierung über den XML-Namespace führt in der Praxis zu unnötig komplexen Abläufen beim Parsen der Befehlsanfragen.

Aus diesem Grund wird zum bisher verwendeten Versionierungsmechanismus über den XML-Namespace zurückgekehrt:

- Der XML-Namespace für die Schnittstellenbefehle wird auf folgenden geändert:`http://www.buergerkarte.at/namespaces/securitylayer/1.2#`.
- Das in der Schnittstellenspezifikation verwendete Namenraum-Präfix `sl` steht somit für den Namen `http://www.buergerkarte.at/namespaces/securitylayer/1.2#`.
- Das Attribut `ProtocolVersion` wird aus sämtlichen Befehlsanfragen sowie aus der Befehlsantwort `sl:GetPropertiesResponse` entfe

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**XML-Schema Core.20020831.xsd**

Bericht	**Fehlernummer**	23	**behoben ab Version**	1.2.0
	am	30. 07. 2004	**von**	Gregor Karlinger, IKT-Stabsstelle

Klassifikation	Fehler

Beschreibung

Die Schema-Definitionen der Typen `sl11:ReferencesCheckResultType`, `sl11:ReferencesCheckResultInfo` `sl11:ManifestRefsCheckResultType`, `sl11:ManifestRefsCheckResultInfoType` sind fehlerhaft. Sie sind durch folgende Definition ersetzen:

```
<xsd:complexType name="ReferencesCheckResultType">
  <xsd:sequence>
    <xsd:element name="Code" type="xsd:nonNegativeInteger"/>
    <xsd:element name="Info" type="ReferencesCheckResultInfoType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ReferencesCheckResultInfoType" mixed="true">
  <xsd:sequence>
    <xsd:element name="FailedReference" type="xsd:positiveInteger" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
```

```
  </xsd:complexType>
  <xsd:complexType name="ManifestRefsCheckResultType">
    <xsd:sequence>
      <xsd:element name="Code" type="xsd:nonNegativeInteger"/>
      <xsd:element name="Info" type="ManifestRefsCheckResultInfoType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ManifestRefsCheckResultInfoType" mixed="true">
    <xsd:sequence>
      <xsd:element name="FailedReference" type="xsd:positiveInteger" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="ReferringSigReference" type="xsd:positiveInteger"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
```

Ort	Version	1.2.1	Auftreten im Dokument	XML-Schema Core-1.2.xsd
Bericht	Fehlernummer	26	behoben ab Version	1.2.2
	am	11. 02. 2005	von	Gernot Egger/Siemens
Klassifikation	Fehler			
Beschreibung	Die Schema-Definition für das Element `sl:GetPropertiesResponse` schreibt vor, dass das Element `sl:KeyboxIdentifier` zumindest ei vorkommen muss. Nachdem die Semantik der Elemente diesen Namens in `sl:GetPropertiesResponse` jedoch jene ist, dass damit angegeben wird, w Schlüsselpaare zum Zeitpunkt der Anfrage für Signatur bzw. Ver-/Entschlüsselung zur Verfügung stehen, muss es auch möglich sein, dass das Ele `sl:KeyboxIdentifier` in `sl:GetPropertiesResponse` gar nicht vorkommt (z.B. wenn sich keine Karte im Kartenleser befindet). Die entsprechende Definition des Datentyps für `sl:GetPropertiesResponse` wird daher wie folgt neu festgelegt:			

```
<xsd:complexType name="GetPropertiesResponseType">
  <xsd:sequence>
    <xsd:element name="ViewerMediaType" type="MimeTypeType" maxOccurs="unbounded"/>
    <xsd:element name="XMLSignatureTransform" type="xsd:anyURI" maxOccurs="unbounded"/>
    <xsd:element name="KeyboxIdentifier" type="QualifiedBoxIdentifierType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Binding" type="BindingType" maxOccurs="unbounded"/>
    <xsd:element name="ProtocolVersion" type="xsd:token" maxOccurs="unbounded"/>
    <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Ort	Version	1.2.1	Auftreten im Dokument	Abschnitt 8.1.2, Aufzählung, 3. Punkt
Bericht	Fehlernummer	27	behoben ab Version	1.2.2
	am	11. 02. 2005	von	Gernot Egger/Siemens
Klassifikation	Editorialer Fehler			
Beschreibung	Aus der Beschreibung des Elements `sl:KeyboxIdentifier` geht nicht klar genug hervor, dass die Semantik des Elements jene ist, dass angegeben wird, welche Schlüsselpaare zum Zeitpunkt der Anfrage für Signatur bzw. Ver-/Entschlüsselung zur Verfügung stehen. Es wird daher im ersten Satz des Aufzählungspunktes nach dem Wort "Bürgerkarten-Umgebung" die Wortfolge "zum Zeitpunkt der Anf eingefügt.			

Ort	Version	1.2.1	Auftreten im Dokument	Abschnitt 8
Bericht	Fehlernummer	29	behoben ab Version	1.2.2
	am	11. 02. 2005	von	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Durch die Streichung der Beispiele seit Version 1.2.0 des Dokuments werden für die Befehle von Kapitel 8 die Namen der zugehörigen X Elemente nicht mehr erwähnt. Es werden daher folgende Änderungen durchgeführt: • Der erste Satz in Abschnitt 8.1.1 lautet nun neu: "Die Anfrage besteht aus dem leeren Element `sl:GetPropertiesRequest`." • Der erste Satz in Abschnitt 8.1.2 lautet nun neu: "Die Antwort besteht aus dem Element `sl:GetPropertiesResponse` und enthä Kindelemente folgende Eigenschaften der *Bürgerkarten-Umgebung*:" • In den ersten Satz in Abschnitt 8.2.1 wird nach dem Wort Befehl die Wortfolge ", bestehend aus dem Element `sl:GetStatusRequ` eingefügt. • Der erste Satz in Abschnitt 8.2.2 lautet nun neu: "Die Antwort besteht aus dem Element `sl:GetStatusResponse` und enthält als Tex Kindelements `sl:TokenStatus` den aktuellen Status des Tokens (entweder `ready` oder `removed`)."			

Ort	Version	1.2.1	Auftreten im Dokument	XML-Schema Core-1.2.xsd
Bericht	Fehlernummer	32	behoben ab Version	1.2.2
	am	01. 03. 2005	von	Rainer Gundacker/IT-Solution
Klassifikation	Fehler			
	In der Schema-Definition für das Element `sl:DecryptedBinaryResponse` fehlen die im Spezifikationstext erwähnten Attribute `MimeType` `Encoding`.			

Beschreibung	Die entsprechende Definition des Datentyps für `sl:DecryptXMLResponse` wird daher wie folgt neu festgelegt:
	`<xsd:complexType name="DecryptXMLResponseType">`
	`<xsd:sequence minOccurs="0">`
	`<xsd:element name="CandidateDocument" type="XMLContentType"/>`
	`<xsd:element name="DecryptedBinaryData" minOccurs="0" maxOccurs="unbounded">`
	`<xsd:complexType>`
	`<xsd:simpleContent>`
	`<xsd:extension base="xsd:base64Binary">`
	`<xsd:attribute name="EncrElemSelector" type="xsd:string" use="required"/>`
	`<xsd:attribute name="MimeType" type="xsd:string" use="optional"/>`
	`<xsd:attribute name="Encoding" type="xsd:anyURI" use="optional"/>`
	`</xsd:extension>`
	`</xsd:simpleContent>`
	`</xsd:complexType>`
	`</xsd:element>`
	`</xsd:sequence>`
	`</xsd:complexType>`

3.3. Errata in "Transportprotokolle Security-Layer"

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 3.3.2**
Bericht	**Fehlernummer**	16	**behoben ab Version**	1.2.0
	am	24. 03. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Der Punkt 5e im Ablauf ist missverständlich formuliert. Er wird daher durch folgende Formulierung ersetzt: • "HTTP-Code 200, wobei die Kombination aus Content-Type und Inhalt nicht unter einen der Punkte 5a, 5b oder 5d fällt; HTTP-Code wobei Content-type nicht unter Punkt 5c fällt; HTTP-Code 301, 302, 303: Die Daten werden als Response unverändert an die Bro Verbindung weitergeleitet, die Browser-Verbindung wird geschlossen und die Verarbeitung beendet."			

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 3.3.2**
Bericht	**Fehlernummer**	19	**behoben ab Version**	1.2.0
	am	04. 09. 2003	**von**	Udo Linauer, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Der Punkt 5b, 5c und 5d sind hinsichtlich der weiteren Verwendung von Weitergabe-Feldern missverständlich formuliert. Sie werden daher folgende Formulierungen ersetzt: • b) HTTP-Code 200, Content-type: text/xml: Die Daten werden als XML-Request ausgewertet und dem Formular-Parameter `XMLReq` zugewiesen. Die anderen Formular-Parameter (`StylesheetURL`, `RedirectURL` und `DataURL`) sowie die Weitergabe-Felder bl unverändert. Die Verarbeitung setzt in Schritt 4 fort. • c) HTTP-Code 307, Content-type: text/xml: Die Daten werden als XML-Request ausgewertet und dem Formular-Parameter `XMLReq` zugewiesen. Die `DataURL` wird für die folgende Verarbeitung auf die im `Location` HTTP-Header enthaltene URL gesetzt. Die an Formular-Parameter (`StylesheetURL` und `RedirectURL`) sowie die Weitergabe-Felder bleiben unverändert. Die Verarbeitung setzt in S 4 fort. • d) HTTP-Code 200, Content-type: application/x-www-form-urlencoded oder multipart/form-data: Die Daten werden als HTTP-Re ausgewertet. Die bisherigen Formular-Parameter sowie Weitergabe-Felder werden verworfen. Ggf. werden die im neuen Request vorhanc Formular-Parameter und Weitergabe-Felder verwendet. Die Verarbeitung setzt in Schritt 3 fort.			

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**Abschnitt 3.3.2.2**
Bericht	**Fehlernummer**	31	**behoben ab Version**	1.2.1
	am	01.03.2005	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Unklarheit			
Beschreibung	Die Erläuterungen zur Kodierung der Formularfelder im HTTP-Request an den DataURL-Server sind nicht schlüssig: 1. Es ist nicht festgelegt, dass die Kodierung immer `multipart/mime` sein muss. Die Erläuterungen für XMLResponse und BinaryResp gehen jedoch davon aus. 2. In den Erläuterungen für XMLResponse und BinaryResponse entsteht der Eindruck, dass `Content-Type` ein Feld von `Cont Disposition` sei. Tatsächlich sind beides jedoch unabhängige Header. Es werden daher folgende Änderungen am Spezifikationstext durchgeführt: 1. Der erste Absatz des Unterkapitels *Kodierung des HTTP-Requests* lautet nun wie folgt: "Der HTTP-Request an den mittels `Dat` bezeichneten Server MUSS nach der Methode `POST` (vgl. [HTTP 1.1], Abschnitt 9.5) erfolgen, als `multipart/form-data` kodiert sein folgende Formularfelder enthalten:". 2. Der zweite Satz des Unterkapitels *Kodierung des HTTP-Requests, Weitergabe-Parameter* wird gestrichen. 3. Der zweite Satz des Unterkapitels *Kodierung des HTTP-Requests, Formular-Parameter XMLResponse* lautet nun wie folgt: "Der H `Content-Type` MUSS für diesen *Mime Part* verwendet werden und ist fix mit dem Wert `text/xml` zu belegen (siehe auch Anmerkung Abschnitt 3.2.3).". 4. Der zweite Satz des Unterkapitels *Kodierung des HTTP-Requests, Formular-Parameter BinaryResponse* lautet nun wie folgt: "Der H `Content-Type` MUSS für diesen *Mime Part* verwendet werden, um den Content-Type der Binär-Response zu bezeichnen. Ist der Content- nicht bekannt, so MUSS der Feldwert mit `application/octet-stream` angegeben werden (siehe auch Anmerkung 2 in Abschnitt 3.2.3)			

3.4. Errata in "Minimale Umsetzung des Security-Layers"

Ort	**Version**	**1.1.0**	**Auftreten im Dokument**	**Abschnitt 2, Tabelle**
Bericht	**Fehlernummer**	3	**behoben ab Version**	1.2.0
	am	21. 02. 2003	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Editorialer Fehler			
Beschreibung	Der Befehl zur Erzeugung eines Sitzungszertifikats wurde mit der Version 1.1.0 abgeschafft. In dieser Tabelle existiert aber noch ein Eintrag diesen Befehl als optional klassifiziert. Dieser Tabelleneintrag ist also nicht mehr als Teil der Spezifikation anzusehen.			

3.5. Errata in "Standardisierte Key- und Infoboxen"

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**Abschnitt 1.1, Tabelle**
Bericht	**Fehlernummer**	25	**behoben ab Version**	1.2.1
	am	11.02.2005	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Editorialer Fehler			
Beschreibung	Der Namenraum der Elemente der Schnittstellenspezifikation korrekterweise `http://www.buergerkarte.at/namespaces/securitylayer/1.2#` anstatt `http://www.buergerkarte.at/namespaces/securitylayer#`.			

3.6. Errata in "Fehlercodes"

Derzeit sind sind keine Fehler bekannt.

3.7. Errata in "Zugriffsschutz"

Derzeit sind sind keine Fehler bekannt.

3.8. Errata in "Anforderungen an die Benutzer-Schnittstelle"

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**Abschnitt 2.1, Authentisierungsklassen, *certifiedGovAgency***
Bericht	**Referenznummer**	28	**behoben ab Version**	1.2.1
	am	11.02.2005	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Fehler			
Beschreibung	Aus den Erläuterungen zur Authentisierungsklasse *certifiedGovAgency* geht nicht hervor, dass es sich beim Ursprung bzw. Ziel anstatt um Behörde auch um einen Dienstleister der Behörde handeln darf. Der Erläuterungstext wird daher wie folgt neu festgesetzt: "Die *Bürgerkarten-Umgebung* hat gesicherte (zertifikatsbasierende) Informationen über den Ursprung der Befehlsanfrage und/oder das Zie Befehlsantwort. Aus diesen Informationen geht hervor, dass es sich bei Ursprung bzw. Ziel um eine Behörde oder einen Dienstleister der Bel handelt, d.h. der Domainname, auf den das Zertifikat ausgestellt ist, matched das Pattern `*.gv.at`, oder das Zertifikat enthält entwede Zertifikatserweiterung *Verwaltungseigenschaft* oder die Zertifikatserweiterung *Dienstleistereigenschaft* [VerwEig]."			

3.9. Errata in "Standard-Anzeigeformat"

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**Abschnitt 2.1.7, drittletzter Absatz**
Bericht	**Referenznummer**	24	**behoben ab Version**	1.2.1
	am	16. 09. 2004	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Fehler			
Beschreibung	Der drittletzten Absatz definiert jene Marker, die in einem anzuzeigenden JPEG JFIF Bild nicht vorkommen dürfen. Unter anderem wird dort auc Marker `APPn` ausgeschlossen. Für das Format JPEG JFIF ist jedoch die Verwendung des Markers `APP0` verpflichtend vorgeschrieben. Der letzte dieses Absatzes wird daher wie folgt geändert: "Enthält eine [JPEG]-Datei jedoch Marker vom Typ `TEM`, `JPG`, `JPGn` (n >= 0), `RSTn` (n >= 0) oder `APPn` (n > 0), muss das Instanzdokument vo Bürgerkarten-Umgebung zurückgewiesen werden."			

Ort	**Version**	**1.2.0**	**Auftreten im Dokument**	**Abschnitt 2.1.2.2, 3. Absatz**
Bericht	**Referenznummer**	30	**behoben ab Version**	1.2.1
	am	01.03.2005	**von**	Gregor Karlinger, IKT-Stabsstelle
Klassifikation	Editorialer Fehler			
	Das Inhaltsmodell für den Content Set `Inline` lautet korrekterweise `br|cite|code|em|span|strong` und nicht wie fälschlich angeg			

Beschreibung	abbr|acronym|br|cite|code|em|span|strong.