

MOA-Amtssignatur – MOA-AS

Spezifikation

Version 1.0.1, 11.02.2008

DI Arne Tauber – arne.tauber@egiz.gv.at

Zusammenfassung:

Das Modul MOA-AS soll dem Umstand des einfachen Aufbringens einer Amtssignatur auf Bescheide oder Erledigungen einer Behörde Rechnung tragen. Das MOA-Servermodul MOA-SS wurde speziell für das Signieren von XML-Dokumenten entwickelt, welches sich nicht direkt für das Aufbringen einer Amtssignatur eignet und nur durch entsprechenden technischen Mehraufwand umsetzbar ist.

MOA-AS soll Applikationen ermöglichen, gängige lesbare Dokumentenformate wie bspw. PDF (Portable Document Format), Microsoft Word oder ODF (Open Document Format) Dokumente mit einer Amtssignatur zu versehen. Das signierte Dokument enthält nach Aufbringen der Amtssignatur einen entsprechenden Signaturblock inkl. Bildmarke der Behörde und den essentiellen zu visualisierenden Daten laut § 19 E-GovG.

Dieses Dokument spezifiziert das Modul MOA-AS.

Inhaltsverzeichnis:

1	Motivation.....	6
1.1	Schlüsselwörter	6
1.2	Geschlechtsspezifische Bezeichnungen	6
1.3	Einleitung	6
„	Erledigungen.....	6
1.4	Amtssignatur	6
	Besonderheiten elektronischer Aktenführung	7
	Amtssignatur	7
	Beweiskraft von Ausdrucken	7
1.4.1	Worum geht es.....	7
1.4.2	Wie sieht die Amtssignatur aus?	7
1.4.3	Nutzen für die Bürger	8
1.5	MOA-AS	8
2	Allgemeine Anforderungen	9
2.1	Unterstützte Plattformen	9

2.2	TLS Authentisierung	9
2.3	Skalierbarkeit und Verfügbarkeit	9
2.4	Namespace	9
2.5	API (Application Program Interface)	9
3	Modell.....	10
3.1	Prozessbeschreibung	11
3.2	Kommunikation	11
3.2.1	Transportprotokoll	11
3.2.2	SOAP (SwA)	11
3.2.3	Webservices	12
4	Signaturerstellung.....	13
4.1	Anfrage	13
4.1.1	SignatureInfo.....	13
4.1.2	DataObject.....	13
4.1.3	Signaturparameter	14
4.1.4	Fehlermeldungen	14
4.2	Antwort	15
4.2.1	SignatureEnvironment.....	15
4.2.2	ErrorResponse.....	15
4.3	PDF Modul	15
4.3.1	Überblick PDF-AS Signaturerstellung.....	16
4.3.2	Mime-Typ	16
4.3.3	PDF-AS Spezifikation.....	16
4.3.4	Signaturparameter	16
4.3.4.1	Typ der PDF Signatur	17
4.3.4.2	Schlüsselname.....	17
4.3.4.3	Signaturblock	17
4.3.4.3.1	Felder des Signaturblocks.....	18
4.3.4.3.1.1	Feldname	18
4.3.4.3.1.2	Standardwert für einen Feldnamen.....	19
4.3.4.4	Platzhalter-Länge.....	19
4.3.4.4.1	Table.....	19
4.3.4.4.1.1	Tabellenreihen.....	20
4.3.4.4.1.2	Tabellenspalten	20
4.3.4.4.1.2.1	Image.....	20
4.3.4.4.1.2.2	Caption	20
4.3.4.4.1.2.3	Value.....	21
4.3.4.4.1.2.4	Width.....	21
4.3.4.4.1.2.5	Inhalt des Td Elements	21
4.3.4.4.2	Styles.....	21
4.3.4.4.2.1	Bildmarke	22
4.3.4.4.2.2	Hintergrundfarbe.....	22
4.3.4.4.2.3	Innenabstand.....	23
4.3.4.4.2.4	Horizontale Ausrichtung.....	23
4.3.4.4.2.5	Vertikale Ausrichtung.....	23
4.3.4.4.2.6	Schriftart.....	23
4.3.4.4.2.6.1	Schriftarten	23
4.3.4.4.2.6.2	Schriftgröße	23
4.3.4.4.2.6.3	Gewichtung.....	23
4.3.4.4.3	Position.....	24
4.3.4.4.3.1	Horizontale Positionierung.....	24
4.3.4.4.3.2	Vertikale Positionierung	24
4.3.4.4.4	Automatische Positionierung.....	25
4.3.4.4.5	Manuelle Positionierung.....	25
4.3.5	Antwort.....	25

5	Signaturprüfung.....	26
5.1	Anfrage.....	26
5.1.1	SignatureInfo.....	26
5.1.2	Angabe des Signaturzeitpunkt.....	26
5.1.3	DataObject.....	26
5.1.4	Signierte Daten.....	27
5.1.5	Prüfparameter.....	27
5.1.6	Fehlermeldungen.....	27
5.2	Antwort.....	28
5.2.1	Prüfergebnis.....	28
5.2.1.1	Prüfung der Gültigkeit der Signatur.....	29
5.2.1.2	Prüfung des Signaturmanifests.....	29
5.2.1.3	Prüfung der Signaturprüfdaten.....	30
5.2.2	ErrorResponse.....	31
5.3	PDF Modul.....	31
5.3.1	Mime-Typ.....	31
5.3.2	PDF-AS Spezifikation.....	31
5.3.3	Prüfparameter.....	31
5.3.3.1	Vertrauensprofil.....	32
5.3.3.2	Angabe der Prüfung einer speziellen Signatur.....	32
5.3.3.3	Angabe des Signaturzertifikats.....	32
6	Anhang A.....	33
6.1	PDF User-Space Einheiten.....	33
6.2	Beispiele zu Ausführungen des Td Elements.....	33
6.2.1	Bildmarke in Tabellenspalte anzeigen.....	33
6.2.2	Bezeichner des Signaturzeitpunkts anzeigen.....	33
6.2.3	Wert des Signaturzeitpunkts anzeigen.....	33
6.2.4	Anfrage für die Erstellung einer Amtssignatur.....	33
6.2.4.1	PDF Dokument direkt eingebettet.....	33
6.2.4.2	PDF Dokument über LocRefContent referenziert.....	33
6.2.5	Antwort auf die Erstellung einer PDF Amtssignatur.....	34
6.2.5.1	PDF Dokument direkt eingebettet.....	34
6.2.5.2	PDF Dokument über LocRefContent referenziert.....	34
6.2.6	Anfrage für die Prüfung einer Amtssignatur.....	35
6.2.6.1	PDF Dokument direkt eingebettet.....	35
6.2.6.2	PDF Dokument über LocRefContent referenziert.....	35
6.2.7	Antwort auf die Prüfung einer PDF Amtssignatur.....	36
6.2.7.1	PDF Dokument direkt eingebettet.....	36
6.2.7.2	PDF Dokument über LocRefContent referenziert.....	36
7	Anhang B.....	38
7.1	Amtssignaturblock für die öffentliche Verwaltung.....	38
7.1.1	Deutsch.....	38
7.1.2	Englisch.....	39
8	Referenzen.....	42

Abbildungen

Abbildung 1: Modell des Moduls MOA-AS	10
Abbildung 2: CreateSignatureRequest Element	13
Abbildung 3: SignatureInfo Element	13
Abbildung 4: CreateSignatureResponse Element.....	15
Abbildung 5: ErrorResponse	15
Abbildung 6: PDF-Signaturparameter	17
Abbildung 7: PDF-Signaturblock.....	18
Abbildung 8: Feld im Signaturblock einer PDF Amtssignatur.....	18
Abbildung 9: Table Element (Design des Signaturblocks)	20
Abbildung 10: Style Definition.....	22
Abbildung 11: Schriftart	23
Abbildung 12: Positionierung des Signaturblocks	24
Abbildung 13: Horizontale Positionierung des Signaturblocks	24
Abbildung 14: Vertikale Positionierung des Signaturblocks	25
Abbildung 15: VerifySignatureRequest Element	26
Abbildung 16: SignatureInfo Element zur Prüfung einer Amtssignatur.....	26
Abbildung 17: VerifySignatureResponse Element	28
Abbildung 18: ErrorResponse.....	31
Abbildung 19: PDF-VerifySignaturparameters	31
Abbildung 20: Amtssignaturblock Deutsch.....	38
Abbildung 21: Amtssignaturblock Englisch	40

Revision History

Version	Datum	Autor(en)	
0.0.1	30.07.2007	Arne Tauber (EGIZ)	Erstversion
0.0.2	13.08.2007	Arne Tauber (EGIZ)	E-GovG Novellierung Signaturprüfung
0.0.3	14.08.2007	Arne Tauber (EGIZ)	Korrekturen
0.1.0	21.08.2007	Arne Tauber (EGIZ)	Änderungsvorschläge C. Herwig
0.2.0	18.09.2007	Arne Tauber (EGIZ)	Entfernung EGovG Novelle
1.0.0	23.01.2008	Arne Tauber (EGIZ)	Anpassungen an Schema 1.0.0
1.0.1	11.02.2008	Arne Tauber (EGIZ)	Anpassung EGovG Novelle

1 Motivation

1.1 Schlüsselwörter

Dieses Dokument verwendet die Schlüsselwörter MUSS, DARF NICHT, ERFORDERLICH, SOLLTE, SOLLTE NICHT, EMPFOHLEN, DARF, und OPTIONAL zur Kategorisierung der Anforderungen. Diese Schlüsselwörter sind analog zu ihren englischsprachigen Entsprechungen MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, und OPTIONAL zu handhaben, deren Interpretation in [KEYWORDS] festgelegt ist.

1.2 Geschlechtsspezifische Bezeichnungen

Alle Personenbezeichnungen, die in diesem Dokument in der männlichen Form verwendet werden, gelten sinngemäß auch für die weibliche Form.

1.3 Einleitung

Die elektronische Kommunikation zwischen zwei Parteien bzw. zwischen Behörde und Partei stellt einen wesentlichen Aspekt jeder E-Government-Anwendung dar. Bisher wurde aus Gründen der Nachweisbarkeit von Anträgen, Bescheiden udgl. und auf deren automatisierten Weiterverarbeitung in Fachapplikationen stets auf signierte XML-Dateien zurückgegriffen.

Der Nachteil dieses Formats, das eigentlich zur elektronischen Verarbeitung von Daten entwickelt wurde, ist die fehlende Lesbarkeit und Rekonstruierbarkeit der übermittelten Daten für den Bürger. Um einer Partei zusätzlich eine "les- und druckbare" Variante zu bieten war es bislang notwendig ein zusätzliches Dokument (meist HTML oder PDF) beizulegen.

Solche schriftliche Erledigungen sind laut § 18 AVG [AVG] mit einer Unterschrift vom Genehmigungsberechtigten zu genehmigen.

Auszug aus dem AVG:

„Erledigungen

§ 18. [...]

(3) Schriftliche Erledigungen sind vom Genehmigungsberechtigten mit seiner Unterschrift zu genehmigen; wurde die Erledigung elektronisch erstellt, kann an die Stelle dieser Unterschrift ein Verfahren zum Nachweis der Identität (§ 2 Z 1 E-GovG) des Genehmigenden und der Authentizität (§ 2 Z 5 E-GovG) der Erledigung treten.

(4) Jede schriftliche Ausfertigung hat die Bezeichnung der Behörde, das Datum der Genehmigung und den Namen des Genehmigenden zu enthalten. Ausfertigungen in Form von elektronischen Dokumenten müssen mit einer Amtssignatur (§ 19 E-GovG) versehen sein; Ausfertigungen in Form von Ausdrucken von mit einer Amtssignatur versehenen elektronischen Dokumenten oder von Kopien solcher Ausdrücke brauchen keine weiteren Voraussetzungen zu erfüllen. Sonstige Ausfertigungen haben die Unterschrift des Genehmigenden zu enthalten; an die Stelle dieser Unterschrift kann die Beglaubigung der Kanzlei treten, dass die Ausfertigung mit der Erledigung übereinstimmt und die Erledigung gemäß Abs. 3 genehmigt worden ist. Das Nähere über die Beglaubigung wird durch Verordnung geregelt.

1.4 Amtssignatur

Für die digitale Unterschrift von Erledigungen seitens der Behörde wurde im E-Government Gesetz [EGovG] eine spezielle Unterschrift – die Amtssignatur – definiert.

Die hier vorgestellten Anforderungen basieren auf dem zum Zeitpunkt der Spezifikationserstellung anwendbaren E-Government Gesetz [EGovG], das mit 01.01.2008 novelliert wurde.

Auszug aus dem E-Government Gesetz:

Besonderheiten elektronischer Aktenführung

Amtssignatur

§ 19. (1) Die Amtssignatur ist eine fortgeschrittene elektronische Signatur im Sinne des Signaturgesetzes, deren Besonderheit durch ein entsprechendes Attribut im Signaturzertifikat ausgewiesen wird.

(2) Die Amtssignatur dient der erleichterten Erkennbarkeit der Herkunft eines Dokuments von einem Auftraggeber des öffentlichen Bereichs. Sie darf daher ausschließlich von diesen unter den näheren Bedingungen des Abs. 3 bei der elektronischen Unterzeichnung und bei der Ausfertigung der von ihnen erzeugten Dokumente verwendet werden.

(3) Die Amtssignatur ist im Dokument durch eine Bildmarke, die der Auftraggeber des öffentlichen Bereichs im Internet als die seine gesichert veröffentlicht hat, sowie durch einen Hinweis im Dokument, dass dieses amtssigniert wurde, darzustellen. Die Informationen zur Prüfung der elektronischen Signatur sind vom Auftraggeber des öffentlichen Bereichs bereitzustellen.

Beweiskraft von Ausdrucken

§ 20. Ein auf Papier ausgedrucktes elektronisches Dokument einer Behörde hat die Beweiskraft einer öffentlichen Urkunde (§ 292 der Zivilprozessordnung – ZPO, RGBI. Nr. 113/1895), wenn das elektronische Dokument mit einer Amtssignatur versehen wurde. Die Amtssignatur muss durch Rückführung des Dokuments aus der ausgedruckten in die elektronische Form prüfbar oder das Dokument muss durch andere Vorkehrungen der Behörde verifizierbar sein. Das Dokument hat einen Hinweis auf die Fundstelle im Internet, wo das Verfahren der Rückführung des Ausdrucks in das elektronische Dokument und die anwendbaren Prüfmechanismen enthalten sind, oder einen Hinweis auf das Verfahren der Verifizierung zu enthalten.

1.4.1 Worum geht es

Die Amtssignatur ist die elektronische Unterschrift einer natürlichen Person, die namens einer **Behörde** handelt. Sie wird auf Bescheide und andere Erledigungen seitens einer Behörde aufgebracht und macht damit kenntlich, dass es sich um ein amtliches Schriftstück handelt. Dies wird im Zertifikat der Signatur durch ein spezielles Attribut (dem Object Identifier der Behörde) ausgedrückt und durch die **Bildmarke** der Behörde visualisiert. Technisch kann die Amtssignatur eine sichere Signatur, eine Verwaltungssignatur oder eine gewöhnliche Signatur sein. Im Zuge der Novellierung entspricht die Amtssignatur einer fortgeschrittenen Signatur im Sinne des Signaturgesetzes.

1.4.2 Wie sieht die Amtssignatur aus?

Die Auswahl und Darstellung bestimmter Merkmale der Amtssignatur **gewährleisten die Sicherheit** der Signatur und damit die Gültigkeit des Dokumentes auch bei einem Ausdruck auf Papier. Für das Aussehen der Amtssignatur gibt es keine verbindliche Regelung. Das E-Government Gesetz [EGovG] hält im §19 lediglich fest, dass die Amtssignatur im Dokument durch eine Bildmarke, die der Auftraggeber des öffentlichen Bereichs im Internet als die seine gesichert veröffentlicht hat, sowie durch einen Hinweis, dass das Dokument amtssigniert wurde, dargestellt werden muss.

Für eine Rekonstruktion von Papier muss neben der Bildmarke in der Ansicht zumindest die Seriennummer sowie der Name und das Herkunftsland des ZDA und der eigentliche Signaturwert angegeben werden. Um eine hohe Akzeptanz zu erreichen und zur besseren Erkennbarkeit einer Amtssignatur wird in der Visualisierung ein möglichst einheitliches Erscheinungsbild der Amtssignatur empfohlen. Die Spezifikation Layout-Amtssignatur [Layout-AS] definiert ein einheitliches standardisiertes Aussehen des Signaturblocks in deutscher sowie englischer Ausprägung. Ist das Dokument gemäß § 20 durch andere Vorkehrungen der Behörde verifizierbar, kann auch nur die Bildmarke des Auftraggebers des öffentlichen Bereichs auf dem Dokument dargestellt werden.

2 Allgemeine Anforderungen

Dieser Abschnitt beschreibt die allgemeinen Anforderungen an das Modul MOA-AS.

2.1 Unterstützte Plattformen

Es muss Java Runtime Environment ab Version 1.5 unterstützt werden.

2.2 TLS Authentisierung

Die Schnittstellen zu allen Services müssen über TLS verfügbar sein. Schwächere SSL Protokolle dürfen aus sicherheitstechnischen Gründen nicht unterstützt werden.

Der Zugriff auf MOA-AS über die Webservice Schnittstelle darf standardmäßig nur über HTTPs (via TLS) erfolgen. Der Zugriff über HTTP kann jedoch explizit in der Konfiguration von MOA-AS freigeschaltet werden.

2.3 Skalierbarkeit und Verfügbarkeit

MOA-AS muss skalierbar und auf einen 7x24h Betrieb ausgelegt sein.

2.4 Namespace

Alle in dieser Spezifikation definierten XML-Elemente sind dem Namespace <http://reference.e-government.gv.at/namespace/moa/20070611#> zugrundegelegt.

2.5 API (Application Program Interface)

Die Funktionalitäten der Signaturerstellung bzw. Signaturprüfung der Webservices müssen auch über ein Java-API zur Verfügung stehen.

3 Modell

Das strukturelle Modell von MOA-AS ist in folgender Abbildung skizziert:

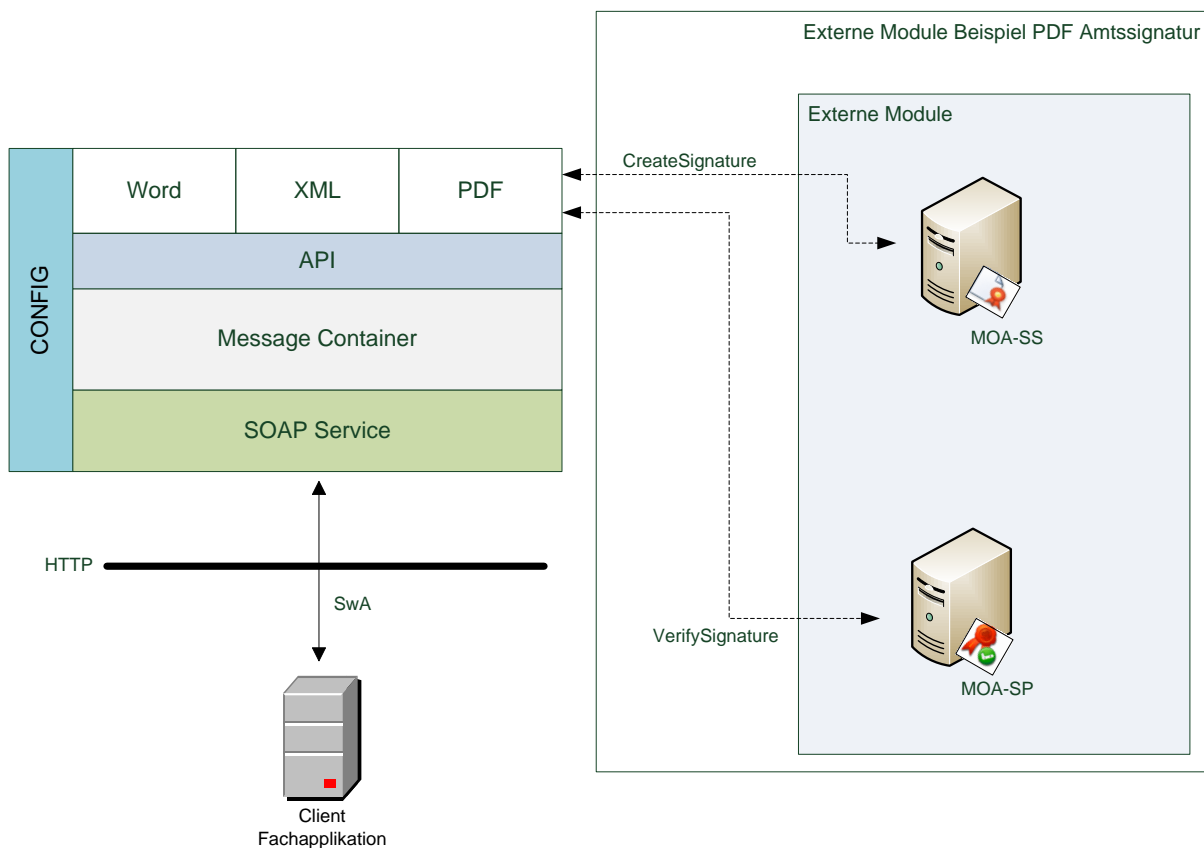


Abbildung 1: Modell des Moduls MOA-AS

Das Kernmodul von MOA-AS ist ein mehrschichtiges abstraktes System aus Komponenten, die alle auf eine gemeinsame Konfiguration zugreifen können. Als Schnittstelle nach außen (zu den Clients) fungiert ein SOAP Service, welches über HTTP bzw. HTTPS (via TLS) angesprochen werden kann und die XML-Nachrichten (Request/Responses) in SOAP-Container ver- bzw. entpackt. Diese SOAP-Container werden im Modell als sog. „Message-Container“ bezeichnet. Die höchste Schicht im Komponentensystem bildet eine API (Application Program Interface), welche vom SOAP-Service die Message-Container von Requests der Clients zur Verfügung gestellt bekommt, bzw. welche Message-Container dem SOAP-Service als Antwort für die Clients zur Verfügung stellt. All diese Komponenten können auf eine gemeinsame Konfiguration des MOA-AS Moduls zurückgreifen.

Die API bildet die gemeinsame Schnittstelle für die Erstellung bzw. Verifikation von Amtssignaturen. Alle Module, die diese Schnittstelle implementieren, können für die Erstellung von Amtssignaturen in MOA-AS verwendet werden. Als Beispiele im Modell sind hier Module für die Erstellung von Amtssignaturen für Microsoft Word Dokumente, XML Dokumente bzw. PDF Dokumente angeführt. Die Art und Weise der Erstellung der Signatur ist ganz allein den einzelnen Modulen überlassen. Beispielsweise signiert und verifiziert das PDF-Amtssignatur Modul die Signaturen mit Hilfe der MOA-Basismodule MOA-SS und MOA-SP [MOA-SP/SS] und lagert somit die Kernfunktionalität auf externe Komponenten aus.

Welches Modul für die Erstellung bzw. Verifikation der Signatur herangezogen wird (bspw. PDF, DOC, ODF), hängt vom Mime-Typ des Dokuments ab. Jedes Modul muss eine Liste von unterstützten Mime-Typen zur Verfügung stellen.

3.1 Prozessbeschreibung

Der Prozess für die Verifikation einer Signatur ist ident mit dem Prozess für die Erstellung einer Signatur. Daher wird nur der Prozess der Signaturerstellung mittels MOA-AS beschrieben. Dieser ist wie folgt:

1. Der Client (z.B. Fachapplikation) stellt eine Verbindung zum Signaturerstellungsservice von MOA-AS über HTTP oder HTTPS her
2. Über diese Verbindung sendet der Client einen XML-Request (Anfrage) zur Signaturerstellung. Dieser Request ist in einem SwA (SOAP with Attachments) Protokoll [SwA] gekapselt. Allfällige Beilagen, die für die Erstellung der Signatur benötigt werden, können innerhalb dieses SwA Request einfach mitgeliefert werden.
3. Das SOAP Service extrahiert die einzelnen Message-Container (XML-Request + allfällige Beilagen) aus dem Container und stellt diese über eine API dem betroffenen Signaturerstellungsmodul zur Verfügung. Welches Signaturmodul aufgerufen wird, hängt vom Mime-Typ des zu signierenden Dokuments ab.
4. Das Signaturerstellungsmodul erstellt die Signatur und liefert das Ergebnis über die API an das SOAP Service von MOA-AS zurück.
5. Dieses SOAP Service kapselt die Antwort innerhalb einer SOAP with Attachments [SwA] Antwort und sendet diese an den Client zurück.

3.2 Kommunikation

Die Kommunikation mit MOA-AS erfolgt über ein Webservice, das via HTTP oder HTTPS angesprochen werden kann. MOA-AS stellt zwei Webservices zur Verfügung: ein Webservice dient zum Erstellen einer Amtssignatur, das zweite Webservice dient zur Prüfung einer solchen. Das verwendete Protokoll, mittels welchem die XML-basierten Nachrichten zur Erstellung bzw. Verifikation einer Amtssignatur zwischen der Clientapplikation und dem MOA-AS Webservice ausgetauscht werden, ist SOAP with Attachments [SwA]. SwA definiert einen Mechanismus, um SOAP [SOAP] Nachrichten in MIME multipart/related [MIME] Container zu kapseln.

3.2.1 Transportprotokoll

Als Transportprotokolle für die Kommunikation zwischen Clients und den MOA-AS Webservices müssen HTTP und HTTPS unterstützt werden. Die HTTPS Kommunikation muss über TLS erfolgen. SSL Protokolle mit geringerer Sicherheitsstufe werden nicht unterstützt.

3.2.2 SOAP (SwA)

Der Austausch der XML-Nachrichten zwischen Clients und den MOA-AS Webservices muss über SwA (SOAP Messages with Attachments) [SwA] erfolgen. Der XML-Request für die Signaturerstellung bzw. Verifikation einer Signatur bzw. dessen Response muss dabei als primäre SOAP-Nachricht vorhanden sein.

3.2.3 Webservices

MOA-AS verfügt über zwei für Clients zugängliche SOAP Webservices. Diese Services müssen über folgende URLs zugänglich sein (relativ auf den Kontext der Webapplikation bezogen):

1. **Signaturerstellung:** `/services/SignatureCreation`
2. **Signaturprüfung:** `/services/SignatureVerification`

4 Signaturerstellung

Dieser Abschnitt beschreibt die Struktur der XML-Nachrichten (Request/Response) für die Erstellung einer Amtssignatur mittels MOA-AS.

4.1 Anfrage

Der Request für die Erstellung einer Amtssignatur erfolgt über das Element `CreateSignatureRequest`.

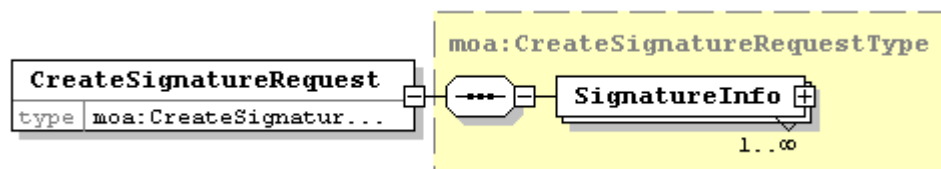


Abbildung 2: CreateSignatureRequest Element

Dieses Element dient als Container für die Erstellung einer beliebigen Anzahl von Amtssignaturen. Jede einzelne zu erstellende Amtssignatur wird über das Element `SignatureInfo` definiert.

4.1.1 SignatureInfo

Das Element `SignatureInfo` definiert eine einzelne zu erstellende Amtssignatur.

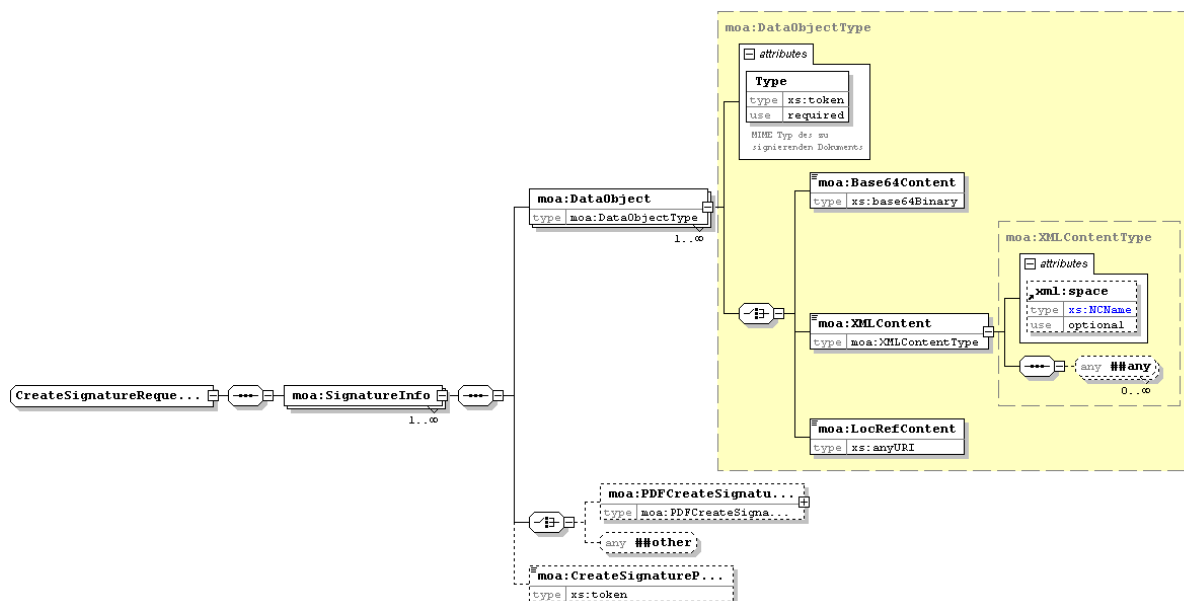


Abbildung 3: SignatureInfo Element

Innerhalb des `SignatureInfo` Elements kann eine beliebige Anzahl von Datenobjekten (`DataObject`) definiert werden.

4.1.2 DataObject

Das erste Datenobjekt in der Liste muss das zu signierende Dokument sein. Allfällige Beilagen und Dokumente, die für den Signaturprozess benötigt werden, können als weitere Datenobjekte angeführt werden. Für jedes Datenobjekt muss explizit über das `Type` Attribut der Mime-Type des enthaltenen Objekts angeführt werden.

Abhängig vom Mime-Typ des ersten Dokuments – welches das zu signierende Dokument darstellt – wird das entsprechende in MOA-AS für diesen Mime-Typ registrierte Modul zur Erstellung der Amtssignatur aufgerufen.

Es gibt drei Möglichkeiten den Inhalt des Datenobjekts zu referenzieren, von denen genau ein Mechanismus verwendet werden muss:

1. Der Inhalt des Datenobjekts ist ein XML Dokument und kann somit direkt innerhalb des `XMLContent` Elements angegeben werden.

Namespace Definitionen für darüber liegende Elemente (z.B. `DataObject`, `SignatureInfo`, `CreateSignatureRequest`) müssen beim Extrahieren des Inhalts aus dem `XMLContent` Element entfernt werden.

2. Der Inhalt des Datenobjekts kann binär als Base64-kodierter Wert innerhalb des `Base64Content` Elements angegeben werden.
3. Über das `LocRefContent` Element des Elements `DataObject`. Der Wert dieses Elements kann eine beliebige URI sein, die jedoch von MOA-AS aus zugänglich sein muss. Referenzen müssen auch für alle Attachments in der SOAP-Nachricht (SwA) aufgelöst werden können.

Für größere Dokumente wird empfohlen, diese als SOAP Attachment zu definieren und das Datenobjekt mittels dem `LocRefContent` Element zu referenzieren.

4.1.3 Signaturparameter

Die Parameter für die Erstellung der Signatur (z.B. Name des Signaturschlüssels usw.) können entweder direkt über die Signaturparameter (z.B. `PDFCreateSignatureParameters`) oder über ein Profil (Element `CreateSignatureProfile`) angegeben werden. Falls ein Signaturprofil verwendet wird, muss dieses in der Konfiguration von MOA-AS definiert werden können.

Die genaue Form der Signaturparameter ist Sache des jeweiligen Signaturmoduls (z.B. PDF) und wird durch dieses Modul definiert. Ein Beispiel für die Ausführung von Signaturparametern ist im Abschnitt der PDF-Amtssignatur näher beschrieben.

Da die Signaturparameter eine beliebige Ausführung haben können, werden diese von der API unverändert dem jeweiligen Signaturmodul übergeben. Für die Überprüfung der Korrektheit dieser Parameter ist somit das jeweilige Signaturmodul verantwortlich.

4.1.4 Fehlermeldungen

Folgende Liste zeigt die standardisierten allgemeinen Fehlermeldungen, die während der Erstellung einer Signatur mittels MOA-AS auftreten können:

Kategorie	Beschreibung
1xxx	
2xxx	
3xxx	
4xxx	
5xxx	Fehler des Signaturmoduls

Fehlernummer	Beschreibung

4.2 Antwort

Als Antwort auf die Anfrage für die Erstellung einer Amtssignatur liefert MOA-AS das `CreateSignatureResponse` Element zurück.

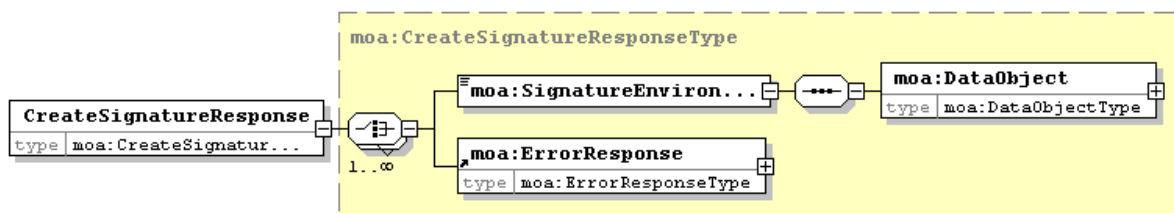


Abbildung 4: CreateSignatureResponse Element

Die Antwort kann entweder die erstellte Signatur (`SignatureEnvironment`) im Erfolgsfall oder im Fehlerfall eine entsprechende Meldung enthalten.

4.2.1 SignatureEnvironment

Das Element `SignatureEnvironment` enthält im Erfolgsfall die erstellte Signatur bzw. eine Referenz (Element `LocRefContent`) auf diese.

Wie der Inhalt dieses Elements strukturiert ist, hängt vom verwendeten Signaturmodul ab. Daher ist der Inhalt für jedes Modul eigens zu spezifizieren.

Die Referenz auf ein Dokument mittels `LocRefContent` muss auch für Anhänge innerhalb der SwA Response auflösbar sein.

4.2.2 ErrorResponse

Im Fehlerfall wird eine entsprechende Meldung mittels dem `ErrorResponse` Element zurückgeliefert.

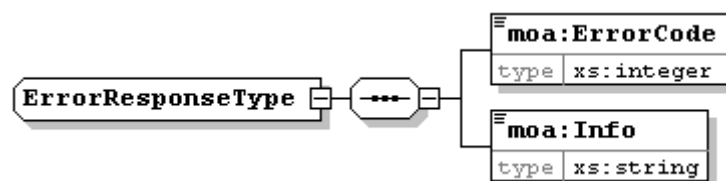


Abbildung 5: ErrorResponse

Die Antwort enthält einen Fehlercode (`ErrorCode` Element) und eine Fehlermeldung (`Info` Element).

4.3 PDF Modul

Das PDF-AS Modul ist ein integraler Bestandteil von MOA-AS und kann zum Aufbringen von Amtssignaturen auf PDF Dokumente verwendet werden.

4.3.1 Überblick PDF-AS Signaturerstellung

Dokumente im PDF-Format sind breit in Verwendung und im Online-Verkehr besonders etabliert - mehr als 200 Millionen PDF-Dokumente im Internet zeugen davon. Um auch im E-Government auf dieses beliebte Dokumentenformat zurückgreifen zu können - bspw. zur Kommunikation von der Behörde hin zum Bürger - müssen PDF-Dokumente auch mit einer elektronischen Signatur versehen werden können. Gerade im Falle von offiziellen Dokumenten der Behörde - wie etwa Bescheiden - werden durch das E-Government Gesetz (E-GovG, §§ 19-21) der auf die Dokumente aufzubringenden (Amts-)Signatur besondere Formvorschriften auferlegt.

Mit MOA-AS können PDF-Dokumente mit einer elektronischen Signatur versehen werden, die bei Bedarf selbst vom Papierausdruck rekonstruiert und validiert werden kann. Zum Aufbringen der Signatur wird dabei das serverseitige Signaturmodul (MOA-SS) verwendet.

Es werden zwei Arten von PDF-Signaturen unterstützt:

- textuelle PDF-Signatur
- binäre PDF-Signatur

Die textuelle Signatur extrahiert nur den Text aus einem gegebenen PDF-Dokument, ignoriert jedoch Bilder und andere nicht textuelle Elemente, und signiert diesen Text in einer normalisierten Weise. So ist gewährleistet, dass textuell signierte PDF-Dokumente jederzeit auch auf Basis eines Papierausdruckes rekonstruiert und letztlich auch deren Signatur geprüft werden kann. Dieses Verfahren eignet sich besonders zur sicheren Signatur rein textueller PDF-Dokumente ohne grafische oder bildhafte Komponenten.

Ergänzend dazu kann eine binäre PDF-Signatur erstellt werden, die zwar das gesamte PDF-Dokument mit allen darin enthaltenen Elementen signiert, deren Signatur aber letztlich nicht mehr von einem Ausdruck rekonstruiert werden kann.

4.3.2 Mime-Typ

Der Mime-Typ, der die Verwendung des PDF-Moduls zur Erstellung einer Amtssignatur kennzeichnet ist „application/pdf“.

4.3.3 PDF-AS Spezifikation

Das Modul MOA-AS unterstützt die Erstellung von PDF Amtssignaturen nach der Spezifikation 1.0.0 [PDF-AS].

4.3.4 Signaturparameter

Dieser Abschnitt beschreibt die Parameter für die Erstellung einer PDF Amtssignatur mittels MOA-AS. Diese Parameter können sowohl im Request enthalten sein bzw. können auch als Profil in der MOA-AS Konfiguration definiert werden.

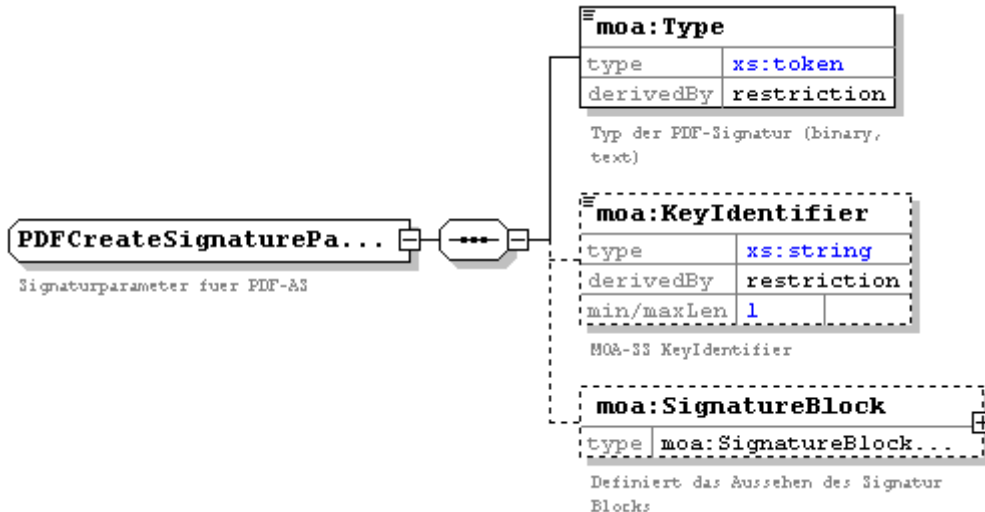


Abbildung 6: PDF-Signaturparameter

Für die Erstellung einer PDF-Amtssignatur müssen drei obligatorische Parameter angegeben werden.

4.3.4.1 Typ der PDF Signatur

Das Element `Type` gibt die Art der PDF Signatur an. Diese kann entweder vom Typ `binary` oder vom Typ `textual` sein. Andere Werte werden nicht unterstützt.

4.3.4.2 Schlüsselname

Die Erstellung der PDF-Amtssignatur erfolgt mit Hilfe des Basismoduls MOA-SS. Für dieses Modul muss ein spezieller Schlüsselname (`KeyIdentifier`) angegeben werden, mit Hilfe dessen die Signatur erstellt werden kann.

4.3.4.3 Signaturblock

Das Element `SignatureBlock` definiert das Aussehen des Signaturblocks.

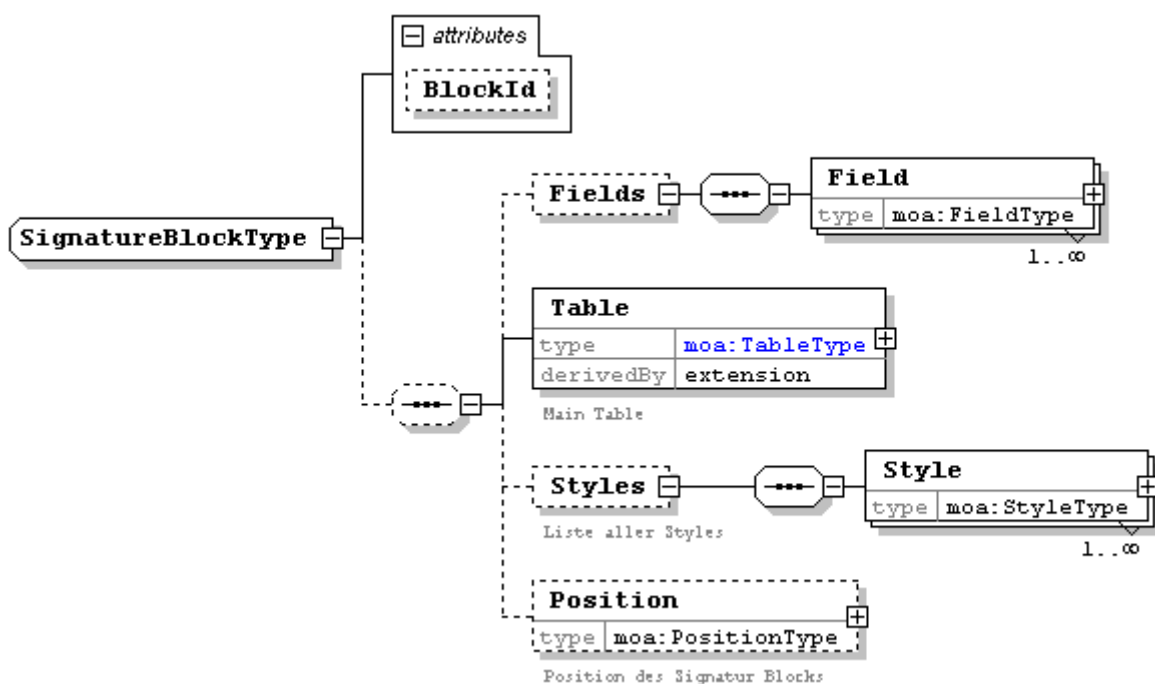


Abbildung 7: PDF-Signaturblock

Das Schema erlaubt ein flexibles und benutzerdefiniertes Design des Signaturblocks.

Der Signaturblock muss nicht zwingend definiert werden, es ist auch möglich einen in der Konfiguration bereits vordefinierten Signaturblock über das `BlockId` Attribut zu referenzieren.

4.3.4.3.1 Felder des Signaturblocks

Das `Field` (Feld) Element definiert die Basiseigenschaften eines Felds, das Teil des Signaturblocks ist. Felder sind z.B. der Signaturzeitpunkt, der Name des Signators, Zertifikatseigenschaften usw.

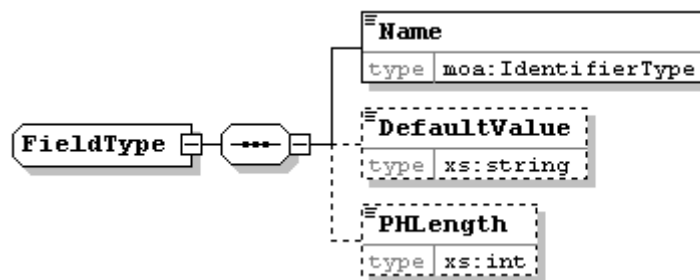


Abbildung 8: Feld im Signaturblock einer PDF Amtssignatur

4.3.4.3.1.1 Feldname

Das Element `Name` gibt den standardisierten Feldnamen an, für welchen die Basiseigenschaften definiert werden sollen. Folgende Felder stehen zur Verfügung:

Feldname	Beschreibung	Variabel	Verpflichtend
SIG_LABEL	Bildmarke der Behörde	nein	ja
SIG_VALUE	Signaturwert	ja	ja
SIG_META	Hinweis, z.B. zur Gültigkeit der Signatur (bspw. nur in elektronischer Form) oder zur Rekonstruktion und Prüfung	nein	nein
SIG_NAME	Inhaber des Zertifikats oder Fertigungsklausel	ja	nein
SIG_ISSUER	Aussteller des Signaturzertifikats	ja	ja
SIG_TYPE	Verfahrenskennung	nein	nein
SIG_NUMBER	Seriennummer des Zertifikats	ja	ja
SIG_KZ	Verwendete Algorithmus	nein	ja
SIG_DATE	Signaturzeitpunkt	ja	ja

4.3.4.3.1.2 Standardwert für einen Feldnamen

Das Element `DefaultValue` gibt den vordefinierten Wert für einen Feldnamen an. Bis auf den Feldnamen `SIG_LABEL`, welcher keine textuelle Darstellung hat, kann der Wert aller Feldnamen angegeben werden.

Beispiel:

Falls das signierte PDF Dokument ein Bescheid ist und von Papier rekonstruierbar bzw. prüfbar ist, könnte im Signaturblock ein Hinweis über den Feldnamen `SIG_META` definiert werden, der wie folgt lauten könnte:

„Dieser Bescheid ist gemäß §20 E-Government-Gesetz (E-GovG) rückführbar. Das Service für die Rückführung und Signaturprüfung kann unter folgender URL abgerufen werden: <http://MeineBehoerde.gv.at/services/SignatureVerification>“

4.3.4.4 Platzhalter-Länge

Das Element `PHLength` darf nur im Falle der Erstellung einer binären PDF Signatur verwendet werden. Bei der binären Signatur wird das gesamte Dokument inklusive Signaturblock signiert. Da zum Signaturzeitpunkt allerdings noch nicht alle Werte verfügbar, also variabel sind, werden für binäre Signaturen im Signaturblock (und im signierten Dokument) Platzhalter freigelassen (bspw. für Signaturwert, Aussteller, etc.).

Textsignaturen benötigen diesen Mechanismus nicht. Die `PHLength`-Werte werden bei Textsignaturen daher nicht verwendet.

Die Standardlängen (in Bytes) für variable Felder müssen für alle verpflichtenden Felder (siehe Tabelle oben) angegeben werden. Ein Wert kleiner gleich 0 gibt an, dass das entsprechende Feld nicht variabel ist, sondern ein statischer Inhalt angezeigt werden soll. Sollte der statische Inhalt (`DefaultValue`) des entsprechenden Feldes leer oder nicht definiert sein, so wird das Feld als Ganzes nicht angezeigt.

Es ist nicht sinnvoll für zur Prüfung verwendete variable Felder statische Werte anzugeben! Die für eine Prüfung erforderlichen Werte lauten: `SIG_DATE`, `SIG_NUMBER`, `SIG_ISSUER`, sowie `SIG_VALUE`.

Einen Sonderfall stellt das `SIG_KZ` Feld dar. Der Wert dieses Feldes wird zum Signaturzeitpunkt vom verwendeten Algorithmus eingesetzt und repräsentiert die URN des Algorithmus laut Spezifikation.

4.3.4.4.1 Table

Mit Hilfe des `Table` Elements kann das Design des Signaturblocks festgelegt werden.

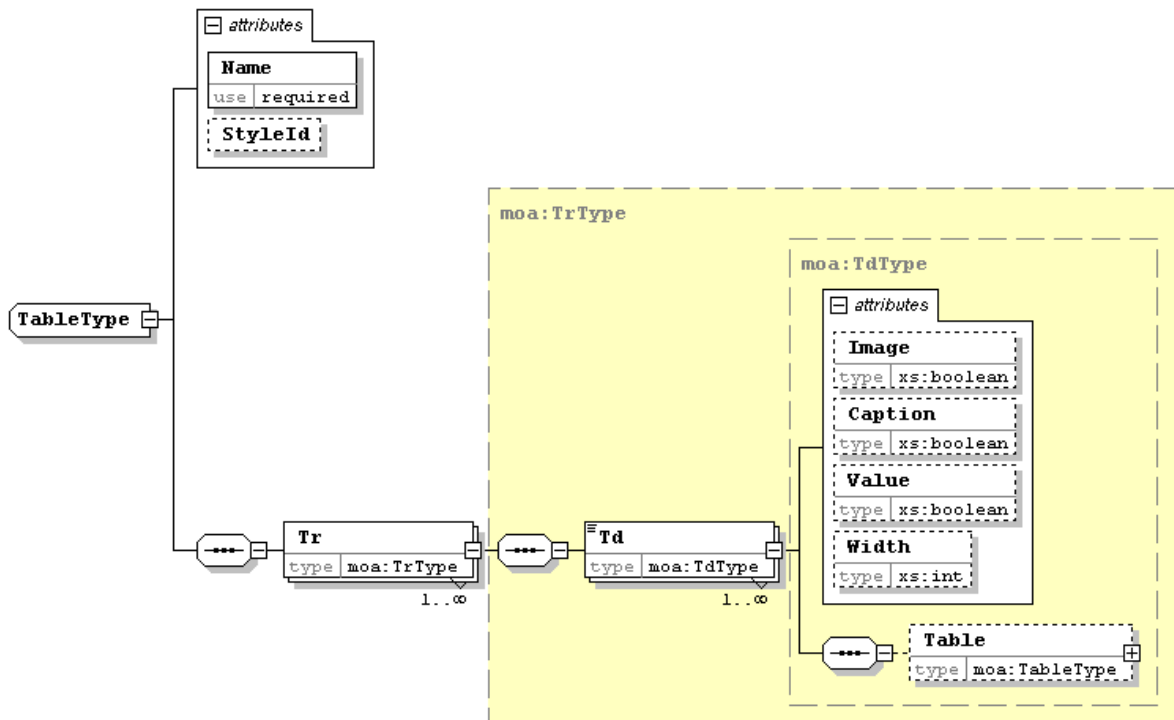


Abbildung 9: Table Element (Design des Signaturblocks)

Das Design des Signaturblocks erfolgt mittels (verschachtelten) Tabellen. Ein Signaturblock besteht aus mindestens einer Tabelle. Der Name der jeweiligen Tabelle muss als `Name` Attribut des `Table` Elements angegeben werden.

Das Aussehen der jeweiligen Tabelle kann über benutzerdefinierte Styles definiert werden, welche im Abschnitt Styles spezifiziert sind. Das optionale Attribut `styleId` muss den Namen eines im Request definierten Styles enthalten. Ist kein Style mit der angegebenen ID vorhanden, wird dieses Attribut ignoriert und das Standarddesign verwendet.

4.3.4.4.1.1 Tabellenreihen

Das Design der Tabellenstruktur erfolgt ähnlich dem Design einer Tabelle in HTML und kann daher analog erfolgen. Eine Tabelle kann eine beliebige Anzahl an Reihen besitzen. Zumindest eine Reihe muss definiert werden. Die Angabe einer Tabellenreihe erfolgt über das `Tr` Element.

4.3.4.4.1.2 Tabellenspalten

Für jede Reihe kann eine beliebige Anzahl an Spalten definiert werden. Zumindest eine Spalte muss definiert werden. Die Angabe einer Tabellenspalte erfolgt über das `Td` Element. Der Inhalt und das Aussehen einer Tabellenspalte wird über die Attribute des `Td` Element und dessen Inhalt definiert.

4.3.4.4.1.2.1 Image

Ist dieses Attribut vorhanden und auf `true` gesetzt, wird die Bildmarke (falls keine angegeben, die Standardbildmarke – siehe Anhang A) in die Tabellenspalte eingefügt.

Alternativ dazu kann als Inhalt des `Td` Elements der Feldname `SIG_LABEL` angeführt werden, was denselben Effekt hat.

4.3.4.4.1.2.2 Caption

Ist dieses Attribut vorhanden und auf `true` gesetzt, wird der Bezeichner des im `Td` Element angeführten Feldnamens eingefügt.

4.3.4.4.1.2.3 Value

Ist dieses Attribut vorhanden und auf `true` gesetzt, wird der Wert des im `Td` Element angeführten Feldnamens eingefügt.

4.3.4.4.1.2.4 Width

Ist dieses Attribut vorhanden, wird die Breite der Tabellenspalte auf den angegebenen Wert festgelegt.

4.3.4.4.1.2.5 Inhalt des `Td` Elements

Der Inhalt des `Td` Elements bestimmt, ob die Bildmarke, der Bezeichner oder Wert eines bestimmten Feldes oder eine verschachtelte Tabelle in die Tabellenspalte eingefügt wird.

Falls der Bezeichner oder der Wert eines bestimmten Feldes in die Spalte eingetragen werden soll, so muss zusätzlich zum entsprechenden Attribut (`Caption` oder `Value`) in das `Td` Element der Feldbezeichner eingetragen werden. Z.B. `SIG_VALUE`.

Für verschachtelte Tabellen kann als Unterelement des `Td` Elements ein `Table` Element eingetragen werden.

Beispiele für unterschiedliche Ausprägungen des `Td` Elements sind in Anhang A angeführt.

4.3.4.4.2 Styles

Das Aussehen einer Tabelle und somit bestimmend für das Aussehen des Signaturblocks wird über `Styles` definiert. Im Rahmen einer Definition für einen Signaturblock können beliebig viele `Styles` definiert werden. Dies erfolgt über das `Styles` bzw. `Style` Element.

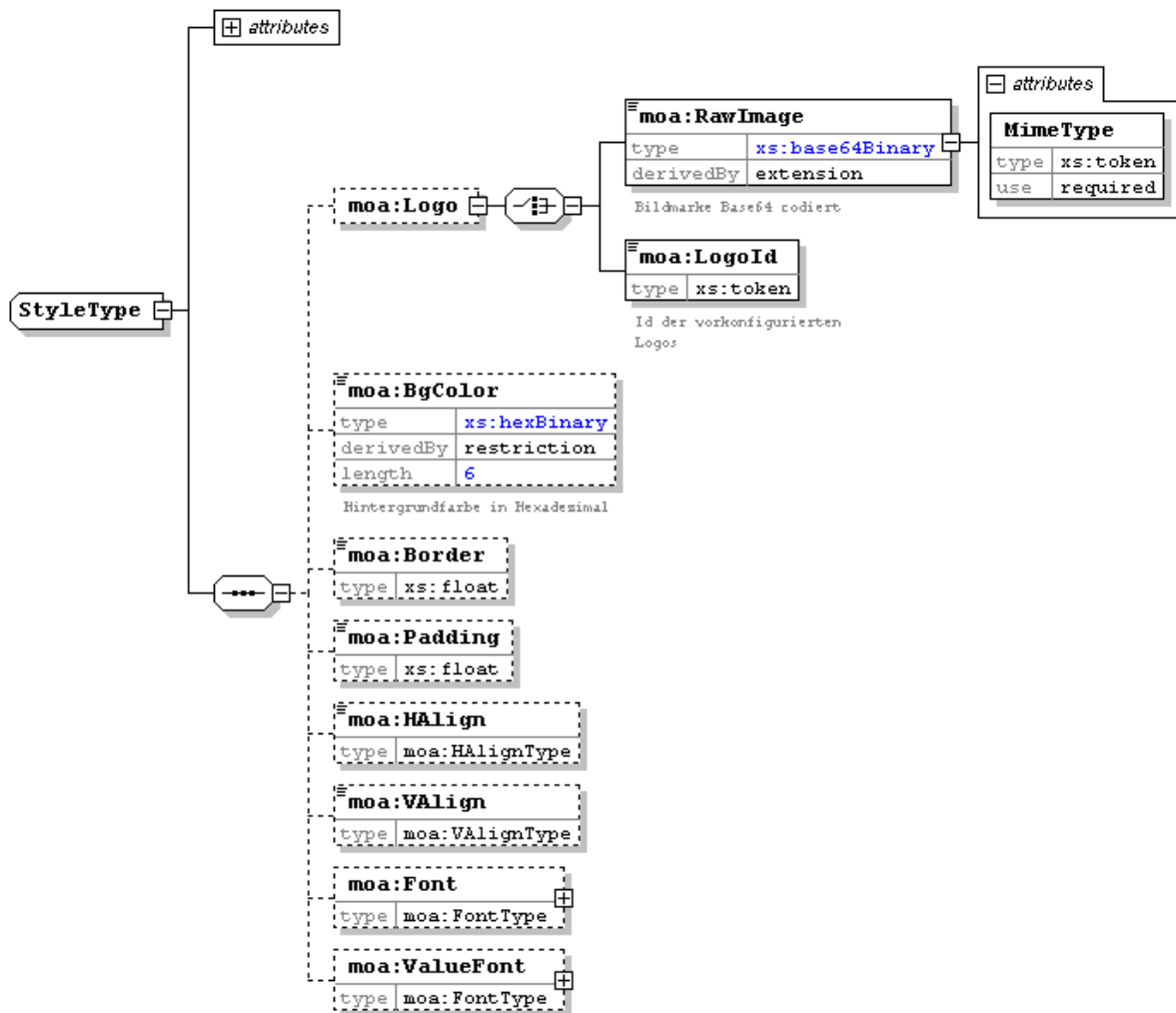


Abbildung 10: Style Definition

Für jedes `Style` Element muss über das `StyleId` Attribut eine eindeutige ID vergeben werden. Diese ID kann dann von jeder Tabelle referenziert werden und erhält somit das in diesem Style definierte Aussehen. Das `Style` Element erlaubt die Definition einer Bildmarke, der Hintergrundfarbe, Rahmen, Innenabstand, horizontale und vertikale Ausrichtung des Textes und der Schriften für Feldbezeichner und Feldwerte.

4.3.4.4.2.1 Bildmarke

Das optionale Element `Logo` ermöglicht die Angabe einer benutzerdefinierten Bildmarke. Diese kann auf zwei Arten referenziert werden:

1. Über die direkte Einbindung der binären Repräsentation des Bildes der Bildmarke über das `RawImage` Element. Das Bild kann in Base64 codierter Form direkt in dieses Element eingebettet werden. Zusätzlich muss das Attribut `MimeType` angegeben werden, welches das Format des binär codierten Bildes angibt. Z.B. `image/gif` für das GIF Format, `image/jpeg` für das JPEG Format oder `image/png` für das PNG Format.
2. Eine weitere Möglichkeit ist die Angabe einer `LogoId`. Diese ID referenziert auf ein Bild, das in der MOA-AS Konfiguration bereits vordefiniert sein muss und somit von MOA-AS verwaltet wird.

4.3.4.4.2.2 Hintergrundfarbe

Die Hintergrundfarbe wird über das Element `BgColor` definiert. Der Wert muss in hexadezimaler Form angegeben werden. Z.B. „#00FF00“

4.3.4.4.2.3 Innenabstand

Der Innenabstand definiert den Abstand des Spalteninhalts zum Spaltenrand (Rahmen). Dieser kann über das Element `Padding` angegeben werden. Der Wert muss eine Fließkommazahl entsprechen und in PDF User-Space Einheiten angegeben werden.

4.3.4.4.2.4 Horizontale Ausrichtung

Die horizontale Ausrichtung des Tabellenspalteninhalts wird über das Element `HAlign` definiert. Mögliche Werte sind:

1. `center` - zentrale Ausrichtung
2. `left` - linksbündig Ausrichtung
3. `right` - rechtsbündig Ausrichtung
4. `justified` - blocksatzweise Ausrichtung (Text)

4.3.4.4.2.5 Vertikale Ausrichtung

Die vertikale Ausrichtung des Spalteninhalts wird über das Element `VAlign` definiert. Mögliche Werte sind:

1. `top` - Ausrichtung am oberen Spaltenende
2. `bottom` - Ausrichtung am unteren Spaltenende
3. `middle` - Ausrichtung an der Spaltenmitte

4.3.4.4.2.6 Schriftart

Es kann sowohl eine Schriftart für den Feldbezeichner als auch für den Feldwert definiert werden. Die Schriftart für den Feldbezeichner wird über das `Font` Element definiert, jene für den Feldwert über das Element `ValueFont`.

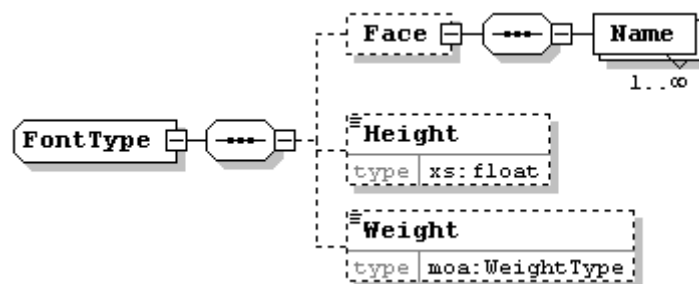


Abbildung 11: Schriftart

4.3.4.4.2.6.1 Schriftarten

Die Liste der zu verwendenden Namen der Schriftarten kann über das Element `Face` definiert werden. Die Reihenfolge der im `Face` Element auftretenden `Name` Elemente bestimmt auch die Reihenfolge der zu verwendenden Schriftartnamen im PDF Dokument.

4.3.4.4.2.6.2 Schriftgröße

Die Schriftgröße kann über das Element `Height` angegeben werden. Der Wert muss eine Fließkommazahl entsprechen und in PDF User-Space Einheiten angegeben werden.

4.3.4.4.2.6.3 Gewichtung

Die Gewichtung der Schrift muss eine der folgenden sein:

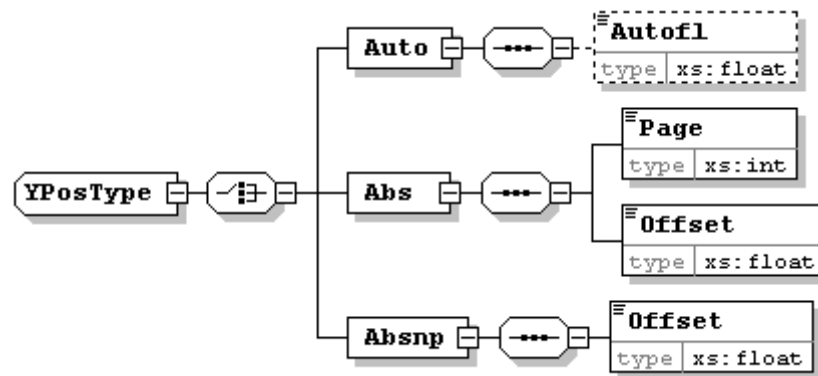


Abbildung 14: Vertikale Positionierung des Signaturblocks

Die vertikale Positionierung des Signaturblocks erfolgt entweder über die Angabe des Elements `Auto` (Automatische Positionierung) oder über die Element `Abs` oder `Absnp` (Manuelle Positionierung).

4.3.4.4.4 Automatische Positionierung

Bei automatischer Positionierung unter Berücksichtigung der Fußzeile (Element `Autofl`) gibt dieser Parameter die y-Koordinate der Oberkante der Fußzeile an. Ist zwischen dem Ende des Fließtextes und der Oberkante der Fußzeile genügend Platz für den Signaturblock, so wird dieser dort platziert. Ansonsten wird der Signaturblock auf eine neue Seite gesetzt.

4.3.4.4.5 Manuelle Positionierung

Eine absolute Positionierung kann über das Element `Abs` erfolgen. Das Element `Page` gibt dabei die Seite an, auf welcher der Signaturblock angebracht werden soll. Diese muss eine positive Ganzzahl im Bereich der im Dokument verfügbaren Seiten sein.

Über das Element `Offset` wird der Abstand zum unteren Seitenrand in PDF User-Space Einheiten definiert.

Eine weitere Möglichkeit der absoluten Positionierung ist die Verwendung des Elements `Absnp`. Bei Verwendung dieses Elements wird eine neue leere Seite an das Dokument angehängt und der Signaturblock auf dieser Seite manuell mit Angabe des `Offset` Elements positioniert. Diese Angabe muss wiederum in PDF User-Space Einheiten erfolgen.

Hinweis: Es ist durchaus möglich den Signaturblock so zu positionieren, dass er nicht sichtbar ist. Weiters kann er durch die Wahl einer sehr kleinen Breite unschön entstellt werden. Es liegt in der Verantwortung des Users eine ansprechende Darstellung und vernünftige Werte für die absolute Positionierung zu wählen.

4.3.5 Antwort

Die Antwort der Erstellung einer PDF-Amtssignatur ist wie folgt:

Entweder kann das Dokument als zusätzliches Attachment zur Response (SwA) mittels dem `LocRefContent` Element des `SignatureEnvironment` Elements referenziert werden, oder das Dokument ist in Base64 codierter Form direkt im `SignatureEnvironment` Element enthalten.

5 Signaturprüfung

Dieser Abschnitt beschreibt die Struktur der XML-Nachrichten (Request/Response) für die Prüfung einer Amtssignatur mittels MOA-AS.

5.1 Anfrage

Der Request für die Prüfung einer Amtssignatur erfolgt über das Element `VerifySignatureRequest`.

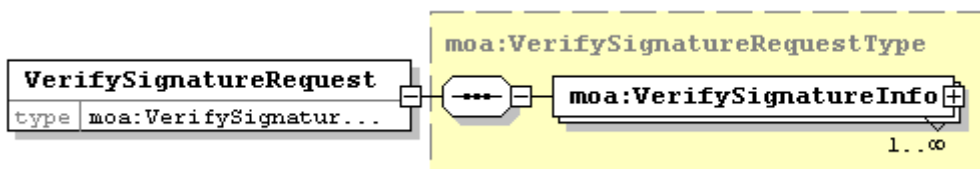


Abbildung 15: VerifySignatureRequest Element

Dieses Element dient als Container für die Prüfung einer beliebigen Anzahl von Amtssignaturen. Jede einzelne zu prüfende Amtssignatur wird über das Element `SignatureInfo` definiert.

5.1.1 SignatureInfo

Das Element `SignatureInfo` definiert eine einzelne zu prüfende Amtssignatur.

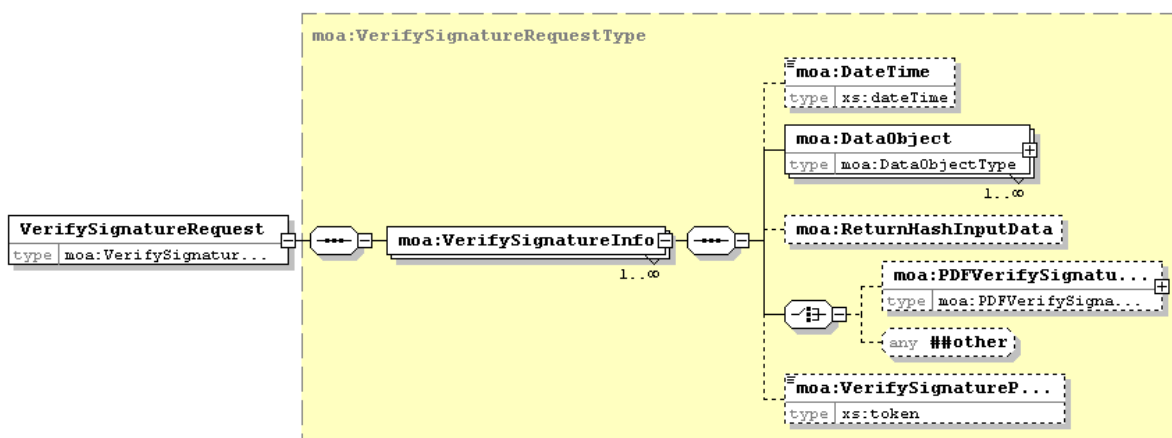


Abbildung 16: SignatureInfo Element zur Prüfung einer Amtssignatur

5.1.2 Angabe des Signaturzeitpunkt

Die Signatur kann auf Gültigkeit zu einem bestimmten Zeitpunkt geprüft werden. Dieser wird über das Element `DateTime` definiert.

5.1.3 DataObject

Innerhalb des `SignatureInfo` Elements kann eine beliebige Anzahl von Datenobjekten (`DataObject`) definiert werden.

Das erste Datenobjekt in der Liste muss das zu prüfende Dokument sein. Allfällige Beilagen und Dokumente, die für den Prüfprozess benötigt werden, können als weitere Datenobjekte

angeführt werden. Für jedes Datenobjekt muss explizit über das `Type` Attribut der Mime-Typ des enthaltenen Objekts angegeben werden.

Abhängig vom Mime-Typ des ersten Dokuments – welches das zu prüfende Dokument darstellt – wird das entsprechende in MOA-AS für diesen Mime-Typ registrierte Modul zur Prüfung der Amtssignatur aufgerufen.

Es gibt drei Möglichkeiten den Inhalt des Datenobjekts zu referenzieren, von denen genau ein Mechanismus verwendet werden muss:

1. Der Inhalt des Datenobjekts ist ein XML Dokument und kann somit direkt innerhalb des `XMLContent` Elements angegeben werden.

Namespace Definitionen für darüber liegende Elemente (z.B. `DataObject`, `SignatureInfo`, `CreateSignatureRequest`) müssen beim Extrahieren des Inhalts aus dem `XMLContent` Element entfernt werden.

2. Der Inhalt des Datenobjekts kann binär als Base64-kodierter Wert innerhalb des `Base64Content` Elements angegeben werden.
3. Über das `LocRefContent` Element des Elements `DataObject`. Der Wert dieses Elements kann eine beliebige URI sein, die jedoch von MOA-AS aus zugänglich sein muss. Referenzen müssen auch für alle Attachments in der SOAP-Nachricht (SwA) aufgelöst werden können.

Für größere Dokumente wird empfohlen, diese als SOAP Attachment zu definieren und das Datenobjekt mittels dem `LocRefContent` Element zu referenzieren.

5.1.4 Signierte Daten

Durch Angabe des optionalen, leeren Elements `ReturnHashInputData` wird MOA-AS angewiesen, im Response jene Daten zurückzuliefern, die von der Signatur abgedeckt sind, d. h. tatsächlich signiert wurden (siehe unten). Diese Information ist für die MOA-AS verwendende Anwendung essentiell, da sie wissen muss, ob tatsächlich die von ihr geforderten Daten signiert wurden. Wird `ReturnHashInputData` im Request nicht angegeben, muss die Anwendung selbst die Signatur analysieren, um diese Information zu erhalten.

5.1.5 Prüfparameter

Die Parameter für die Prüfung der Signatur (z.B. Name des Vertrauensprofils – `TrustProfile` - usw.) können entweder direkt über die Prüfparameter (z.B. `PDFVerifySignatureParameters`) oder über ein Profil (Element `VerifySignatureProfile`) angegeben werden. Falls ein Prüfprofil verwendet wird, muss dieses in der Konfiguration von MOA-AS definiert werden können.

5.1.6 Fehlermeldungen

Folgende Liste zeigt die standardisierten allgemeinen Fehlermeldungen, die während der Prüfung einer Signatur mittels MOA-AS auftreten können:

Kategorie	Beschreibung
1xxx	
2xxx	
3xxx	

4xxx	
5xxx	Fehler des Prüfmoduls

Fehlernummer	Beschreibung

5.2 Antwort

Als Antwort auf die Anfrage für die Prüfung einer Amtssignatur liefert MOA-AS das `VerifySignatureResponse` Element zurück.

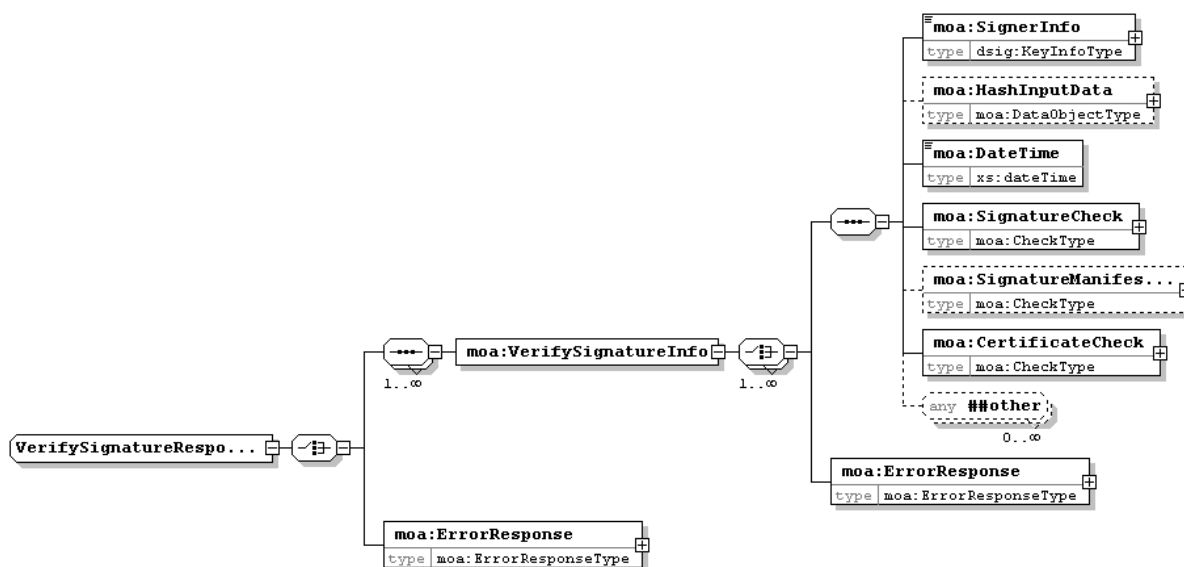


Abbildung 17: VerifySignatureResponse Element

Die Antwort kann entweder das Prüfergebnis der einzelnen zu prüfenden Signaturen im Erfolgsfall oder im Fehlerfall eine entsprechende Meldung enthalten.

5.2.1 Prüfergebnis

Die Antwort auf die Anfrage zur Prüfung einer einzelnen Signatur (`VerifySignatureInfo`) enthält zunächst Informationen zum öffentlichen Schlüssel des Signators (`SignerInfo`). Konnte bei der Überprüfung ein dem öffentlichen Schlüssel entsprechendes Signatorzertifikat nach X.509 identifiziert werden, muss `SignerInfo` zumindest folgende Informationen enthalten:

- Genau ein `dsig:X509Data` Element, das wiederum mindestens drei Elemente enthalten muss:
 - `dsig:SubjectName` enthält den Namen des Signators aus dem Signatorzertifikat. Das Element ist wie in [XMLDSIG], Kapitel 4.4.4 angegeben zu kodieren. In `dsig:X509Data` muss genau ein solches Element vorkommen.
 - `dsig:IssuerSerial` enthält den Namen des Ausstellers und die Seriennummer des Signatorzertifikats. Das Element ist wie in [XMLDSIG], Kapitel 4.4.4

angegeben zu kodieren. In `dsig:X509Data` muss genau ein solches Element vorkommen.

- Weiters muss `dsig:X509Data` das leere Element `QualifiedCertificate` enthalten, wenn das Signatorzertifikat als qualifiziert anzusehen ist. Dazu müssen folgende Bedingungen erfüllt sein:
 - Es muss die Zertifikatserweiterung `qcStatements` (siehe [QCert], 3.2.5) vorhanden sein.
 - Innerhalb dieser Zertifikatserweiterung muss das Statement `esti-qcStatement-1` (siehe [ETSIQCert], 4.2.1) vorhanden sein.

Ob weitere Elemente - wie beispielsweise das Signatorzertifikat selbst - zurückgeliefert werden, bleibt dem Prüfmodul überlassen. Konnte bei der Überprüfung kein Signatorzertifikat nach X.509 identifiziert werden, bleibt es ebenfalls dem Prüfmodul überlassen, welche Informationen über den öffentlichen Schlüssel es zurückliefert.

Wurde im Request das Element `ReturnHashInputData` angegeben, so müssen in der Antwort jene Daten zurückgeliefert werden, die von der Signatur abgedeckt sind (`HashInputData` Element).

Das Element `DateTime` liefert den genauen Signaturzeitpunkt zurück.

Weiters enthält die Antwort getrennte Ergebnisse für die kryptographische Überprüfung der Signatur (`SignatureCheck`), für die Prüfung des Signaturmanifests (`SignatureManifestCheck` – optional falls keine Manifest vorhanden), für die Überprüfung ggf. vorhandener weiterer Manifeste (`any ##other`) oder anderer Prüfergebnisinformationen, sowie für die Prüfung der Signaturprüfdaten (`CertificateCheck`).

Alle Ergebnisse besitzen eine ähnliche Struktur: Das Element `s1:Code` enthält das Ergebnis der Prüfung in maschinenlesbarer Form, das Element `s1:Info` kann genauere Zusatzinformationen enthalten (siehe dazu die folgenden Unterabschnitte).

5.2.1.1 Prüfung der Gültigkeit der Signatur

Folgende vordefinierte Werte sind für den Inhalt des Elements `Code` in `SignatureCheck` definiert:

Code	Bedeutung
0	Die Überprüfung der Hash-Werte und des Werts der Signatur konnte erfolgreich durchgeführt werden.
1	Bei der Überprüfung des Hash-Werts zumindest einer Referenz der Signatur ist ein Fehler aufgetreten. Der Wert der Signatur wurde nicht überprüft.
2	Die Überprüfung der Hash-Werte konnte erfolgreich durchgeführt werden. Beim Überprüfen des Werts der Signatur ist jedoch ein Fehler aufgetreten.

Weitere Fehlermeldungen können von jedem Prüfmodul nach Belieben definiert werden.

5.2.1.2 Prüfung des Signaturmanifests

Kann die zu prüfende Signatur ein Manifest enthalten, kann ein solches geprüft werden. In einem solchen Fall ist der Hash-Wert jeder Referenz des Signaturmanifests zu überprüfen. Die Prüfung der Signaturprüfdaten darf unterbleiben, wenn bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist.

Folgende Werte sind für den Inhalt des Elements `Code` in `SignatureManifestCheck` definiert:

Code	Bedeutung
0	Dieser Code hat eine der folgenden Bedeutungen: <ul style="list-style-type: none"> • Für diese Signatur ist kein Signaturmanifest notwendig. • Die Signatur enthält eine Referenz auf das notwendige Signaturmanifest. Das Signaturmanifest entspricht vom Umfang her den Anforderungen dieser Spezifikation. Für jede Referenz des Signaturmanifests konnte der Hash-Wert erfolgreich überprüft werden.
1	Die Signatur enthält keine Referenz auf das notwendige Signaturmanifest.
2	Die Signatur enthält zwar eine Referenz auf das Signaturmanifest, dieses entspricht vom Umfang her jedoch nicht den Anforderungen dieser Spezifikation. Die Hash-Werte der im Signaturmanifest vorhandenen Referenz-Elemente wurden nicht überprüft.
3	Die Signatur enthält eine Referenz auf das Signaturmanifest. Das Signaturmanifest entspricht vom Umfang her den Anforderungen dieser Spezifikation. Bei der Überprüfung des Hash-Werts zumindest einer Referenz des Signaturmanifests ist jedoch ein Fehler aufgetreten.
99	Die Prüfung des Signaturmanifests wurde nicht durchgeführt, da bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist.

5.2.1.3 Prüfung der Signaturprüfdaten

Diese umfasst die Konstruktion der Zertifikatskette vom Signatorzertifikat bis zu einem vertrauenswürdigen Wurzelzertifikat, sowie die Statusprüfung für jedes Zertifikat der konstruierten Zertifikatskette. Die Prüfung der Signaturprüfdaten darf unterbleiben, wenn bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist.

Folgende Werte sind für den Inhalt des Elements `Code` in `CertificateCheck` definiert:

Code	Bedeutung
0	Eine formal korrekte Zertifikatskette vom Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert werden. Jedes Zertifikat dieser Kette ist zum in der Anfrage angegebenen Prüfzeitpunkt gültig.
1	Es konnte keine formal korrekte Zertifikatskette vom Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konstruiert werden.
2	Eine formal korrekte Zertifikatskette vom Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert werden. Für zumindest ein Zertifikat dieser Kette fällt der Prüfzeitpunkt nicht in das Gültigkeitsintervall.
3	Eine formal korrekte Zertifikatskette vom Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert werden. Für alle Zertifikate dieser Kette fällt der Prüfzeitpunkt in das jeweilige Gültigkeitsintervall. Für zumindest ein Zertifikat konnte der Zertifikatsstatus nicht festgestellt werden.
4	Eine formal korrekte Zertifikatskette vom Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert werden. Für alle Zertifikate dieser Kette fällt der Prüfzeitpunkt in das jeweilige Gültigkeitsintervall. Zumindest ein Zertifikat ist zum Prüfzeitpunkt widerrufen.
5	Eine formal korrekte Zertifikatskette vom Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert werden. Für alle Zertifikate dieser Kette fällt der Prüfzeitpunkt in das jeweilige Gültigkeitsintervall. Kein Zertifikat

	dieser Kette ist zum Prüfzeitpunkt widerrufen. Zumindest ein Zertifikat ist zum Prüfzeitpunkt gesperrt.
99	Die Prüfung der Signaturprüfdaten wurde nicht durchgeführt, da bei der Prüfung der Gültigkeit der Signatur ein Fehler aufgetreten ist.

5.2.2 ErrorResponse

Im Fehlerfall wird eine entsprechende Meldung mittels dem `ErrorResponse` Element zurückgeliefert.

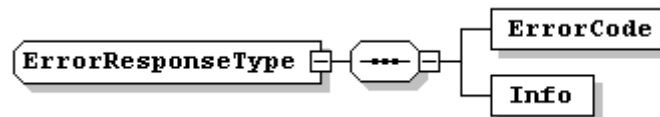


Abbildung 18: ErrorResponse

Die Antwort enthält einen Fehlercode (`ErrorCode` Element) und eine Fehlermeldung (`Info` Element).

5.3 PDF Modul

Das PDF-AS Modul ist ein integraler Bestandteil von MOA-AS und kann zum Prüfen von Amtssignaturen von PDF Dokumenten verwendet werden.

5.3.1 Mime-Typ

Der Mime-Typ, der die Verwendung des PDF-Moduls zur Prüfung einer Amtssignatur kennzeichnet ist „application/pdf“.

5.3.2 PDF-AS Spezifikation

Das Modul MOA-AS unterstützt die Prüfung von PDF Amtssignaturen nach der Spezifikation 1.0.0 [PDF-AS].

5.3.3 Prüfparameter

Dieser Abschnitt beschreibt die Parameter für die Prüfung einer PDF Amtssignatur mittels MOA-AS. Diese Parameter können sowohl im Request enthalten sein bzw. können auch als Profil in der MOA-AS Konfiguration definiert werden.

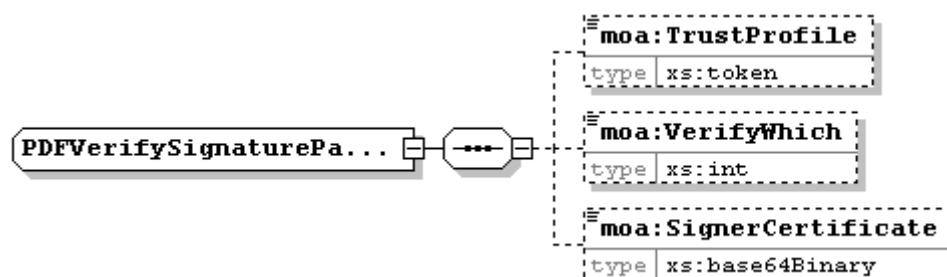


Abbildung 19: PDF-VerifySignaturparameters

Für die Erstellung einer PDF-Amtssignatur müssen folgende obligatorische Parameter angegeben werden.

5.3.3.1 Vertrauensprofil

Da die PDF-Amtssignatur mittels MOA-SP überprüft wird, muss über das Element `TrustProfile` das zu verwendende Vertrauensprofil angegeben werden.

5.3.3.2 Angabe der Prüfung einer speziellen Signatur

Über die Angabe des Elements `VerifyWhich` kann dediziert eine einzelne Signatur (falls mehrere Signatur im Dokument vorhanden sind) geprüft werden. Der Wert ist ein Integer Wert. Die erste Signatur muss mit dem Wert 0 gekennzeichnet sein.

5.3.3.3 Angabe des Signaturzertifikats

Mit dem `SignerCertificate` Element kann das Signaturzertifikat übergeben werden. Bei PDF-AS wird das Signaturzertifikat aus einem lokalen Store geholt und erfordert somit alle Signaturzertifikate in diesem Store zu halten. Mit dieser Erweiterung muss das Zertifikat nicht zwingend im Store liegen.

6 Anhang A

6.1 PDF User-Space Einheiten

Die Größe der Einheit des PDF User-Space ist 1/72 Zoll (2,54 cm).

Eine hochformatige A4 Seite ist demnach 595 Einheiten breit und 842 Einheiten hoch.

6.2 Beispiele zu Ausführungen des `Td` Elements

6.2.1 Bildmarke in Tabellenspalte anzeigen

```
<Td Image="true"></Td>
```

oder

```
<Td>SIG_LABEL</Td>
```

Ergebnis z.B.:



6.2.2 Bezeichner des Signaturzeitpunkts anzeigen

```
<Td Caption="true">SIG_DATE</Td>
```

Ergebnis z.B.: „Zeitpunkt der Signatur“

6.2.3 Wert des Signaturzeitpunkts anzeigen

```
<Td Value="true">SIG_DATE</Td>
```

Ergebnis z.B. 2007-08-02T08:57:13

6.2.4 Anfrage für die Erstellung einer Amtssignatur

6.2.4.1 PDF Dokument direkt eingebettet

```
<moa:CreateSignatureRequest xmlns:moa="http://reference.e-  
government.gv.at/namespace/moa/20070611#">  
  <moa:SignatureInfo>  
    <moa:DataObject Type="application/pdf">  
      <moa:Base64Content>RGFzIGlzdCBlaW4</moa:Base64Content>  
    </moa:DataObject>  
    <moa:PDFCreateSignatureParameters>  
      <!-- Hier stehen die Signaturparameter -->  
    </moa:PDFCreateSignatureParameters>  
  </moa:SignatureInfo>  
</moa:CreateSignatureRequest>
```

6.2.4.2 PDF Dokument über LocRefContent referenziert

POST /moaas/services/SignatureCreation HTTP/1.1

MOA-Amtssignatur (MOA-AS) Signatur im E-Government

```
POST /moaas/services/SignatureCreation HTTP/1.1
Host: localhost
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
    start="MyId"
Content-Length: 27897
SOAPAction: CreateSignature
Content-Description: This request signs a PDF document with MOA-SS

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: MyId

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <moa:CreateSignatureRequest xmlns:moa="http://reference.e-
government.gv.at/namespace/moa/20070611#">
      <moa:SignatureInfo Type="application/pdf">
        <moa:DataObject>
          <moa:LocRefContent>DocumentToSign.pdf</moa:LocRefContent>
        </moa:DataObject>
        <moa:PDFSignatureParameters>
          <!-- Hier stehen die Signaturparameter -->
        </moa:PDFSignatureParameters>
      </moa:SignatureInfo>
    </moa:CreateSignatureRequest>
  </soap:Body>
</soap:Envelope>

--MIME_boundary
Content-Type: application/pdf
Content-Transfer-Encoding: base64
Content-ID: DocumentToSign.pdf

...Base64 encoded PDF document goes here...
--MIME_boundary--
```

6.2.5 Antwort auf die Erstellung einer PDF Amtssignatur

6.2.5.1 PDF Dokument direkt eingebettet

```
<moa:CreateSignatureResponse xmlns:moa="http://reference.e-
government.gv.at/namespace/moa/20070611">
  <moa:SignatureEnvironment>
    <moa:DataObject Type="application/pdf">
      <moa:Base64Content>Base64-codiertes signiertes PDF
Dokument</moa:Base64Content>
    </moa:DataObject>
  </moa:SignatureEnvironment>
</moa:CreateSignatureResponse>
```

6.2.5.2 PDF Dokument über LocRefContent referenziert

```
Host: localhost
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
    start="MyId"
Content-Length: 27897
Content-Description: This response contains a PDF-AS verification result

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: MyId

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```
<moa:CreateSignatureResponse xmlns:moa="http://reference.e-  
government.gv.at/namespace/moa/20070611#">  
  <moa:SignatureEnvironment>  
    <moa:DataObject>  
      <moa:LocRefContent>SignedDocument-1.pdf</moa:LocRefContent>  
    </moa:DataObject>  
  </moa:SignatureEnvironment>  
</moa:CreateSignatureResponse>  
</soap:Body>  
</soap:Envelope>
```

```
--MIME_boundary  
Content-Type: application/pdf  
Content-Transfer-Encoding: base64  
Content-ID: SignedDocument-1.pdf
```

```
...Base64 encoded signed PDF document goes here...  
--MIME_boundary--
```

6.2.6 Anfrage für die Prüfung einer Amtssignatur

6.2.6.1 PDF Dokument direkt eingebettet

```
<moa:VerifySignatureRequest xmlns:moa="http://reference.e-  
government.gv.at/namespace/moa/20070611#">  
  <moa:VerifySignatureInfo>  
    <moa:DateTime>2001-12-17T09:30:47.0Z</moa:DateTime>  
    <moa:DataObject Type="application/pdf">  
      <moa:Base64Content>RGFzIGlzdCBlaW4gZWl</moa:Base64Content>  
    </moa:DataObject>  
    <moa:ReturnHashInputData/>  
    <moa:PDFVerifySignatureParameters>  
      <!-- Hier stehen die Prüfparameter -->  
    </moa:PDFVerifySignatureParameters>  
  </moa:VerifySignatureInfo>  
</moa:VerifySignatureRequest>
```

6.2.6.2 PDF Dokument über LocRefContent referenziert

```
POST /moaas/services/SignatureVerification HTTP/1.1  
Host: localhost  
Content-Type: multipart/related; boundary=MIME_boundary; type=text/xml;  
  start="MyId"  
Content-Length: 27897  
SOAPAction: CreateSignature  
Content-Description: This request verifies a PDF document with MOA-SP
```

```
--MIME_boundary  
Content-Type: text/xml; charset=UTF-8  
Content-Transfer-Encoding: 8bit  
Content-ID: MyId
```

```
<?xml version="1.0"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
  <soap:Body>  
    <moa:VerifySignatureRequest xmlns:moa="http://reference.e-  
government.gv.at/namespace/moa/20070611#">  
      <moa:VerifySignatureInfo>  
        <moa:DateTime>2001-12-17T09:30:47.0Z</moa:DateTime>  
        <moa:DataObject Type="application/pdf">  
          <moa:Base64Content>RGFzIGlzdCBlaW4gZWluZmFjaGVyIFRleHQ</moa:Base64Content>  
        </moa:DataObject>  
        <moa:ReturnHashInputData/>  
        <moa:PDFVerifySignatureParameters>  
          <!-- Hier stehen die Prüfparameter -->  
        </moa:PDFVerifySignatureParameters>  
      </moa:VerifySignatureInfo>  
    </moa:VerifySignatureRequest>  
  </soap:Body>  
</soap:Envelope>
```

```
        </moa:PDFVerifySignatureParameters>
    </moa:VerifySignatureInfo>
</moa:VerifySignatureRequest>
</soap:Body>
</soap:Envelope>
--MIME_boundary--
```

6.2.7 Antwort auf die Prüfung einer PDF Amtssignatur

6.2.7.1 PDF Dokument direkt eingebettet

```
<moa:VerifySignatureResponse xmlns:moa="http://reference.e-
government.gv.at/namespace/moa/20070611#">
  <moa:VerifySignatureInfo>
    <moa:SignerInfo>
      <dsig:X509Data>
        <dsig:X509IssuerSerial>
          <dsig:X509IssuerName>AuthorityName</dsig:X509IssuerName>
          <dsig:X509SerialNumber>123456</dsig:X509SerialNumber>
        </dsig:X509IssuerSerial>
        <dsig:X509Certificate>
          <!-- Base 64 encoded signer certificate goes here -->
        </dsig:X509Certificate>
      </dsig:X509Data>
    </moa:SignerInfo>
    <moa:DateTime>2001-12-17T09:30:47.0Z</moa:DateTime>
    <moa:SignatureCheck>
      <moa:Code>0</moa:Code>
      <moa:Info>Die Überprüfung der Hash-Werte und des Werts der Signatur
konnte erfolgreich durchgeführt werden.</moa:Info>
    </moa:SignatureCheck>
    <moa:SignatureManifestCheck>
      <moa:Code>0</moa:Code>
      <moa:Info>Für diese Signatur ist kein Signaturmanifest
notwendig.</moa:Info>
    </moa:SignatureManifestCheck>
    <moa:CertificateCheck>
      <moa:Code>0</moa:Code>
      <moa:Info>Eine formal korrekte Zertifikatskette vom Signatorzertifikat
zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert werden. Jedes
Zertifikat dieser Kette ist zum in der Anfrage angegebenen Prüfzeitpunkt
gültig.</moa:Info>
    </moa:CertificateCheck>
  </moa:VerifySignatureInfo>
</moa:VerifySignatureResponse>
```

6.2.7.2 PDF Dokument über LocRefContent referenziert

```
Host: localhost
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
  start="MyId"
Content-Length: 27897
Content-Description: This response contains a signed PDF document

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: MyId

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <moa:VerifySignatureResponse xmlns:moa="http://reference.e-
government.gv.at/namespace/moa/20070611#"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
      <moa:VerifySignatureInfo>
        <moa:SignerInfo>
          <dsig:X509Data>
            <dsig:X509IssuerSerial>
```

MOA-Amtssignatur (MOA-AS)
Signatur im E-Government

```

        <dsig:X509IssuerName>AuthorityName</dsig:X509IssuerName>
        <dsig:X509SerialNumber>123456</dsig:X509SerialNumber>
    </dsig:X509IssuerSerial>
    <dsig:X509Certificate>
        <!-- Base 64 encoded signer certificate goes here -->
    </dsig:X509Certificate>
    </dsig:X509Data>
</moa:SignerInfo>
<moa:DateTime>2001-12-17T09:30:47.0Z</moa:DateTime>
<moa:SignatureCheck>
    <moa:Code>0</moa:Code>
    <moa:Info>Die Überprüfung der Hash-Werte und des Werts der
Signatur konnte erfolgreich durchgeführt werden.</moa:Info>
</moa:SignatureCheck>
<moa:SignatureManifestCheck>
    <moa:Code>0</moa:Code>
    <moa:Info>Für diese Signatur ist kein Signaturmanifest
notwendig.</moa:Info>
</moa:SignatureManifestCheck>
<moa:CertificateCheck>
    <moa:Code>0</moa:Code>
    <moa:Info>Eine formal korrekte Zertifikatskette vom
Signatorzertifikat zu einem vertrauenswürdigen Wurzelzertifikat konnte konstruiert
werden. Jedes Zertifikat dieser Kette ist zum in der Anfrage angegebenen
Prüfzeitpunkt gültig.</moa:Info>
</moa:CertificateCheck>
</moa:VerifySignatureInfo>
</moa:VerifySignatureResponse>
</soap:Body>
</soap:Envelope>
--MIME_boundary
```



```

<Td Image="true"/>
<Td>
  <Table Name="info">
    <Tr>
      <Td Caption="true" Width="20%">SIG_NAME</Td>
      <Td Value="true" Width="80%">SIG_NAME</Td>
    </Tr>
    <Tr>
      <Td Caption="true">SIG_DATE</Td>
      <Td Value="true">SIG_DATE</Td>
    </Tr>
    <Tr>
      <Td Caption="true">SIG_ISSUER</Td>
      <Td Value="true">SIG_ISSUER</Td>
    </Tr>
    <Tr>
      <Td Caption="true">SIG_NUMBER</Td>
      <Td Value="true">SIG_NUMBER</Td>
    </Tr>
    <Tr>
      <Td Caption="true">SIG_KZ</Td>
      <Td Value="true">SIG_KZ</Td>
    </Tr>
  </Table>
</Td>
</Tr>
<Tr>
  <Td Caption="true">SIG_META</Td>
  <Td Value="true">SIG_META</Td>
</Tr>
</Table>
<Styles>
  <Style StyleId="MainStyle">
    <BgColor>#FFFFFF</BgColor>
    <Border>0.1</Border>
    <Padding>3</Padding>
    <HAlign>left</HAlign>
    <VAlign>middle</VAlign>
    <Font>
      <Face>
        <Name>Helvetica</Name>
      </Face>
      <Height>8</Height>
      <Weight>Normal</Weight>
    </Font>
    <ValueFont>
      <Face>
        <Name>Courier</Name>
      </Face>
      <Height>8</Height>
      <Weight>Normal</Weight>
    </ValueFont>
  </Style>
</Styles>
</SignatureBlock>

```

7.1.2 Englisch

Die Struktur und Feldbezeichnungen für den Amtssignaturblock sind wie folgt festgelegt:


```
<Tr>
  <Td Caption="true">SIG_NUMBER</Td>
  <Td Value="true">SIG_NUMBER</Td>
</Tr>
<Tr>
  <Td Caption="true">SIG_KZ</Td>
  <Td Value="true">SIG_KZ</Td>
</Tr>
</Table>
</Td>
</Tr>
<Tr>
  <Td Caption="true">SIG_META</Td>
  <Td Value="true">SIG_META</Td>
</Tr>
</Table>
<Styles>
  <Style StyleId="MainStyle">
    <BgColor>#FFFFFF</BgColor>
    <Border>0.1</Border>
    <Padding>3</Padding>
    <HAlign>left</HAlign>
    <VAlign>middle</VAlign>
    <Font>
      <Face>
        <Name>Helvetica</Name>
      </Face>
      <Height>8</Height>
      <Weight>Normal</Weight>
    </Font>
    <ValueFont>
      <Face>
        <Name>Courier</Name>
      </Face>
      <Height>8</Height>
      <Weight>Normal</Weight>
    </ValueFont>
  </Style>
</Styles>
</SignatureBlock>
```

8 Referenzen

- [Layout-AS] T. Rössler, Layout Amtssignatur, Spezifikation, Version 1.0.0
- [PDF-AS] W. Prinz, T. Rössler, PDF Amtssignatur, Spezifikation, Version 1.0.0, vom 04.10.2006.
- [MIME] E. Levinson, Request for Comments 2387 (RFC2387). The MIME Multipart/Related Content-type.
- [SOAP] SOAP Version 1.2. Abgerufen am 26.07.2007 unter <http://www.w3.org/TR/soap/>
- [SwA] J. Barton: SOAP Messages with Attachments, W3C Note 11 December 2002. Abgerufen am 26.07.2007 unter <http://www.w3.org/TR/SOAP-attachments>
- [XMLDSIG] Eastlake, Donald, Reagle, Joseph und Solo, David: XML-Signature Syntax and Processing. W3C Recommendation, Februar 2002. Abgerufen aus dem World Wide Web am 14. 05. 2004 unter <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>
- [MOA-SP/SS] MOA: Serversignatur (SS), Signaturprüfung (SP). Abgerufen aus dem World Wide Web am 17.04.2007 unter <http://www.cio.gv.at/onlineservices/basicmodules/moa-spss/>
- [AVG] Allgemeines Verwaltungsverfahrensgesetz, 1991. Abgerufen am 07.08.2007 unter <http://www.bka.gv.at/DocView.axd?CobId=9880>
- [EGovG] Bundesgesetzblatt der Republik Österreich, ausgegeben am 27. Februar 2004, Bundesgesetz, mit dem ein E-Government-Gesetz erlassen wird sowie das Allgemeine Verwaltungsverfahrensgesetz 1991, das Zustellgesetz, das Gebührengesetz 1957, das Meldegesetz 1991 und das Vereinsgesetz 2002 geändert werden. Abgerufen am 23.07.2007 unter http://ris1.bka.gv.at/authentic/findbgbl.aspx?name=entwurf&format=pdf&docid=COO_2026_100_2_30412
- [KEYWORDS] Bradner, S.: RFC 2119: Key words for use in RFCs to Indicate Requirement Levels. IETF Request For Comment, März 1997. Abgerufen aus dem World Wide Web am 14. 05. 2004 unter <http://www.ietf.org/rfc/rfc2119.txt>.