



Zentrum für sichere Informationstechnologie – Austria Secure Information Technology Center - Austria

A-1040 Wien, Weyringergasse 35
Tel.: ++43 1 - 503 19 63 - 0
Fax: ++43 1 - 503 19 63 - 66

A-8010 Graz, Inffeldgasse 16a
Tel.: ++43 316 - 873 5514
Fax: ++43 316 - 873 5520

Homepage: www.a-sit.at
E-Mail: office@a-sit.at

Bescheinigung nach §18(5) SigG: BDC hot:Sign, Signatur-Client zur Erstellung sicherer Signaturen, Version 1.0

Antragsteller:
BDC EDV Consulting GmbH
Gredlerstraße 4
1020 Wien

1. Beschreibung der bescheinigten Komponente

Der Gegenstand der Bescheinigung ist BDC hot:Sign, ein Signatur-Client zur Erstellung sicherer Signaturen, Version 1.0, nachstehend der Signatur-Client genannt. Hersteller des Signatur-Clients ist die BDC EDV Consulting GmbH, Gredlerstraße 4, 1020 Wien.

Der Signatur-Client hat sein typisches Anwendungsfeld im privaten Bereich und in einer normalen Büroumgebung, er wird vom Hersteller auf einem read-only Datenträger bzw. per Download über HTTPS (SSL bzw. TLS) mit einer gesicherten Konfiguration für die beschriebenen Komponenten ausgeliefert. Jede Erweiterung der zu benutzenden Komponenten erfordert vom Hersteller das Erzeugen einer erweiterten gesicherten Konfiguration. Zum Lieferumfang gehört ein Benutzerhandbuch für Endbenutzer bzw. ein Entwicklerhandbuch für Entwickler von Anwendungen, die auf dem Signatur-Client basieren.

1.1. Funktionsumfang und technische Einsatzumgebung

Der Signatur-Client ist eine unter Win32 lauffähige Applikation, es werden folgende Versionen des Microsoft Windows Betriebssystems unterstützt:

- Microsoft Windows 98 SE, Internet Explorer 6.0
- Microsoft Windows ME, Internet Explorer 6.0
- Microsoft Windows NT 4.0 (SP6+), Internet Explorer 6.0
- Microsoft Windows 2000 professional (SP2), Internet Explorer 6.0
- Microsoft Windows XP, Internet Explorer 6.0

Der Signatur-Client benötigt mindestens folgende Hardware: PC mit min. 500 MHz, 128 MB RAM, 100 MB freier Festplattenspeicher, Bildschirmauflösung min. 800x600, max. 1280x1024, min. 256 Farben.

Folgende Funktionen des Signatur-Clients sind für diese Bescheinigung relevant:

- **Secure Viewer:** Diese Funktion prüft das Format der zu signierenden Daten und bringt diese nach erfolgreicher Prüfung sicher zur Anzeige.
- **Hashberechnung:** Diese Funktion berechnet den Hashwert über die zu signierenden Daten, als Hashverfahren wird SHA-1 benutzt, das laut SigV Anhang 2, Punkt 2 bis 31.12.2005 als sicher anzusehen ist.

Zusätzlich stellt der Signatur-Client folgende Funktionen zur Verfügung:

- Ändern der Signatur-PIN
- Entsperren von PINs
- Verifikation von Signaturen
- Verwaltung von Zertifikaten und Zertifizierungsinstanzen

Diese Funktionen sind nicht Gegenstand der Bescheinigung.

Es wird folgender CT-API Treiber zum Ansprechen des Chipkartenterminals unterstützt:

- CT-API Treiber für Kobil KAAAN professional, v2001.4.3.1

Es wird folgendes Chipkartenterminal unterstützt:

- Kobil KAAAN professional v2.08 GK v.1.04

Das Chipkartenterminal verfügt über ein integriertes PIN-Pad und ein integriertes Display. Eine Signaturerstellung ist nur möglich, wenn die PIN-Eingabe am PIN-Pad des Terminals durchgeführt wird. Das Ändern der Signatur-PIN erfolgt ebenfalls mit Mitteln des unterstützten Chipkartenterminals.

Es werden folgende Signaturkarten unterstützt:

- A-Trust TrustSign v1.2 für sichere Signaturen nach dem Signaturgesetz.
- A-Trust TrustMark v3.12 für einfache Signaturen.

1.2. Funktionsbeschreibung

Der Signatur-Client ist ein einfacher HTTP-responder, der Anfragen auf dem TCP/IP-Port 20000 entgegennehmen und bearbeiten kann. Anfragen werden nur vom lokalen Rechner entgegengenommen, als Protokoll zwischen aufrufender Anwendung und Signatur-Client wird HTTP verwendet, dabei wird ausschließlich die HTTP-Operation „post“ unterstützt. Es werden zwei unterschiedliche Kommandos zur Erstellung von Signaturen unterstützt:

- **CreateXMLSignatureRequest:** Erstellung von Signaturen im XMLDSIG Format (nach Internet RFC 3275 bzw. ETSI TS 101 902)
- **CreateCMSSignatureRequest:** Erstellung von Signaturen im CMS Format (nach Internet RFC 2630 bzw. ETSI TS 101 733)

Das Content-Format der zu signierenden Daten wird als Parameter übergeben. Der Secure Viewer prüft, ob die Daten dem spezifizierten Format entsprechen und bringt die Daten nach erfolgreicher Prüfung sicher zur Anzeige. Entsprechen die zu signierenden Daten nicht einem unterstütztem Format, wird die Signaturerstellung mit einer Fehlermeldung abgebrochen. Es werden drei Content-Formate unterstützt:

- **text/plain:** Die Daten werden ohne Interpretation der Formatierung angezeigt. Bei der Interpretation des Inhalts ist das Character-Set relevant, in dem die Daten vorliegen, dazu kann das Character-Encoding als zusätzlicher Parameter übergeben werden. Wird kein Character-Encoding angegeben, werden die Daten als ISO-8859-1 interpretiert. Im **Anhang A** findet sich eine Liste aller unterstützten Character-Sets.
- **text/xml:** Die Daten werden als XML-Daten interpretiert und die XML-Elemente werden hierarchisch dargestellt. Als Einrückungszeichen werden jeweils vier Leerzeichen verwendet, die zulässigen Character-Sets sind in **Anhang B** aufgelistet.
- **text/html:** Die Daten werden nach dem HTML-Standard interpretiert und angezeigt, die zulässigen Character-Encodings entsprechen den Character-Sets für „text/xml“. Um sicherzustellen, dass alle HTML-Elemente vom Secure Viewer interpretiert und richtig angezeigt werden können, ist der HTML-Sprachumfang eingeschränkt. Die zulässigen HTML-Elemente sind in einem XML-Schema festgelegt (siehe **Anhang C**). Wenn die vorliegenden Daten nicht dem spezifiziertem HTML-Sprachumfang entsprechen, wird der Signaturvorgang abgebrochen. Zusätzlich werden die verwendbaren Text- und Hintergrundfarben eingeschränkt, die Verwendung einer Farbe als Text- und Hintergrundfarbe wird ausgeschlossen. Es werden außerdem bestimmte Farbkombinationen für Text und Hintergrund ausgeschlossen (siehe **Anhang D**).

Die Formate XML und HTML sehen Kommentar-Tags vor, die üblicherweise nicht angezeigt werden. Aus diesem Grund ist sowohl beim Content-Format text/xml als auch beim Content-Format text/html die Verwendung von Kommentaren ausgeschlossen.

Um zuverlässig nachvollziehen zu können, wie die signierten Daten dem Signator im Viewer dargestellt wurden, werden das Content-Format und auch das Character-Encoding, die für die Darstellung verwendet wurden, in die resultierende Signatur miteinbezogen. Die Form der Einbindung in die Signatur ist vom verwendeten Signatur-Format abhängig. Bei XMLDSIG erfolgt dies im „Mimetype“ – Attribut des Elements „Reference“. Bei CMS erfolgt dies im ersten UTF8String – Element des Signaturattributes „ContentHints“.

Der Signatur-Client unterstützt XSLT (Stylesheet) –Transformationen nach der XMLDSIG-Spezifikation. Die zu signierenden Daten müssen im text/xml-Format vorliegen, die mittels eines Stylesheets, das ebenfalls im CreateXMLSignatureRequest übergeben wird, für die Anzeige aufbereitet werden. Die Ausgangsdaten dieser Transformation können das Format text/html oder text/xml annehmen und werden entsprechend vom Secure-Viewer geprüft und angezeigt. Signiert werden nur die transformierten Daten.

Nach dem Aufrufen eines der beiden oben beschriebenen Signatur-Requests wird der Signator aufgefordert, eine Signaturkarte in das Chipkartenterminal zu stecken. Dann wird das Format der zu signierenden Daten geprüft. Ist das Format der Daten ungültig, wird der Vorgang abgebrochen und eine entsprechende Fehlermeldung erzeugt. Nach erfolgreicher Prüfung werden die Daten entsprechend ihres Formates im Fenster des Secure-Viewers angezeigt. Wird eine Signaturkarte mit qualifiziertem Zertifikat verwendet, wird ein zusätzlicher Hinweis angezeigt, dass eine sichere Signatur erstellt wird. Der Signator kann dann durch Klicken eines entsprechenden Buttons entweder den Signaturvorgang starten oder abbrechen. Zusätzlich besteht noch die Möglichkeit sich das Signaturzertifikat der verwendeten Karte anzeigen zu lassen. Zum endgültigen Auslösen des Signaturvorganges muss die Signatur-PIN am PIN-Pad des verwendeten Kartenlesers eingegeben werden. Die Berechnung des Hashwertes über die zu signierenden Daten geschieht mittels der im Signatur-Client integrierten „**Embedded Intel Cryptographic Library**“ (**EICL**) der Intel **CDSA-Suite** (Common Data Security Architecture), als Algorithmus wird SHA-1 eingesetzt. Die Berechnung der Signatur aus dem Hashwert wird ausschließlich auf der verwendeten Signaturkarte durchgeführt.

Nach erfolgter Signaturberechnung liefert der Signatur-Client der aufrufenden Applikation das angeforderte Signatur-Format zurück:

- **XMLDSIG:** Für jedes in der Anfrage übergebene Datenobjekt enthält die XML-Signatur ein „dsig:Reference“ Element. In dessen „dsig:Transforms“ Element ist jene Transformationskette angegeben, die während der Signatur-Erstellung durchlaufen wurde, um aus dem übergebenen Datenobjekt jene Daten zu erhalten, die für die Berechnung des Hash-Wertes sowie für die Anzeige im Secure Viewer verwendet worden sind. Zusätzlich ist ein Signaturattribut enthalten, mit dem das für die Verifizierung der Signatur zu verwendende Zertifikat eindeutig identifiziert wird. Dazu wird das signierte Signaturattribut „etsi:SigningCertificate“ nach ETSI TS 101 902 (XAdES) verwendet.
- **CMS:** Die Antwort enthält die erzeugte Signatur nach CMS (ETSI TS 101 733) in base64-kodierter Form. In der CMS-Signatur ist ein signiertes Signaturattribut „ContentHints“ enthalten, das Informationen über das Format der Daten enthält. Zusätzlich wird ein signiertes Signaturattribut „signingCertificate“ laut Enhanced Security Services for S/MIMEv3 (RFC 2634) in die CMS-Signatur aufgenommen, mit dem das für die Verifikation der Signatur zu verwendende Zertifikat eindeutig identifiziert wird.

2. Erfüllung der Anforderungen des SigG und der SigV

BDC hot:Sign, Signatur-Client zur Erstellung sicherer Signaturen, Version 1.0 erfüllt

- die Anforderung nach §18(1) SigG an technische Komponenten und Verfahren, dass sie die Fälschung von Signaturen sowie die Verfälschung signierter Daten zuverlässig erkennbar machen und die unbefugte Verwendung von Signaturerstellungsdaten verlässlich verhindern,
- die Anforderung nach §18(2) SigG an technische Komponenten und Verfahren, dass die zu signierenden Daten nicht verändert werden,
- die Anforderung nach §18(2) SigG an technische Komponenten und Verfahren, dass dem Signator die zu signierenden Daten vor Auslösung des Signaturvorgangs dargestellt werden,
- die Anforderung nach §7(1) SigV an zulässige Hashverfahren (Anhang 2 Punkt 2 SigV)
- die Anforderung nach §7(2) SigV an technische Komponenten und Verfahren, dass sie die vollständige Anzeige der zu signierenden Daten ermöglichen müssen,
- die Anforderung nach §7(2) SigV, dass die Spezifikation der Formate der zu signierenden Daten allgemein verfügbar ist,
- die Anforderung nach §7(2) SigV, dass in den zu signierenden Daten dynamische Veränderungen oder unsichtbare Daten ausgeschlossen sind.

Die genannten Anforderungen werden unter den unten genannten Einsatzbedingungen erfüllt.

BDC hot:Sign, Signatur-Client zur Erstellung sicherer Signaturen, Version 1.0 ist daher in folgenden Kategorien bescheinigt:

- Komponenten und Verfahren zur Darstellung der zu signierenden Daten.
- Komponenten und Verfahren zum Erzeugen des Hashwertes aus dem Dokument.

3. Gültigkeitsdauer der Bescheinigung

Diese Bescheinigung ist bis 30.6.2004 gültig.

4. Einsatzbedingungen

4.1. Chipkartenterminal

Zum Zeitpunkt der Ausstellung der Bescheinigung ist der Betrieb mit folgendem Chipkartenterminal vorgesehen:

- Kobil KAAN Professional v2.08 GK v1.04

Das Chipkartenterminal muss direkt am benutzten PC angeschlossen sein und muss sich bei der Signaturerstellung im selben Raum wie der benutzte PC befinden. Der Signator muss sich von der unmittelbaren Verbindung des Chipkartenterminals und des PC vergewissern können, da diese sicherheitsrelevant ist.

4.2. Signaturkarte

Zum Zeitpunkt der Ausstellung der Bescheinigung ist der Betrieb mit folgender Signaturkarte vorgesehen:

- A-Trust TrustSign Karte v1.2
(Smart Card mit Chip - Philips Smart Card Controller P8WE5032V0G und Betriebssystem STARCOS SPK 2.3 und Digital Signature Application TrustSign, von A-SIT am 22-10-2001 nach §18(5) SigG bescheinigt)

4.3. Einsatzumgebung

Die Verantwortung für die Integrität der Daten bei der Übertragung zum Chipkartenterminal liegt nicht im Verantwortungsbereich der bescheinigten Komponente. Die Integrität der

Daten ist durch geeignete technische und/oder organisatorische Maßnahmen in der Einsatzumgebung sicherzustellen.
Der Signator muss außerdem sicherstellen, dass die Sicherheit der technischen Einsatzumgebung des Signatur-Clients, nicht durch bösartige Software kompromittiert ist. (Siehe auch Abschnitt 1.1.)

5. Algorithmen und zugehörige Parameter

Zum Erzeugen des Hashwertes aus dem Dokument wird vom Signatur-Client das SHA-1 Hashverfahren mit 160 Bit bereitgestellt. Dadurch sind die Anforderungen gemäß Anhang 2 Punkt 2 SigV erfüllt.

6. Prüfstufe und Mechanismenstärke

BDC hot:Sign, Signatur-Client zur Erstellung sicherer Signaturen, Version 1.0 wurde im Rahmen der Bescheinigung von A-SIT begutachtet.

Als Leitlinie für die Begutachtung der Vertrauenswürdigkeit wurden die Gemeinsamen Kriterien für die Prüfung und Bewertung der Sicherheit von Informationstechnik (Common Criteria – ISO/IEC 15408) - Teil 3: Anforderungen an die Vertrauenswürdigkeit herangezogen. Folgende Bereiche wurden positiv begutachtet:

- Sicherheitsziele
- Konfigurationsmanagement
- Auslieferung und Betrieb der bescheinigten Komponente
- Entwicklung der bescheinigten Komponente
- Handbücher
- Lebenszyklus-Unterstützung
- Tests
- Schwachstellenbewertung

Eine Beurteilung der Mechanismenstärke ist nicht anwendbar.

Die Korrektheit der verwendeten Implementierung des SHA-1 Hashverfahrens wurde durch Tests mit den SHA-1 Referenzvektoren des NIST (<http://csrc.nsl.nist.gov/cryptval/shs.html>) nachgewiesen.

Wien, 20.6.2002

A-SIT Zentrum für sichere Informationstechnologie - Austria



o. Univ.-Prof. Dipl.-Ing. Dr. Reinhard Posch
Wissenschaftlicher Gesamtleiter



Manfred Holzbach
Geschäftsführender Vorstand

Anhang A – Unterstützte Character-Encodings bei text/plain

ANSI_X3.4-1968 (US-ASCII)	Cp869	Cp1124
Big5	Cp870	Cp1258
Big5-HKSCS	Cp871	Cp1381
Cp037	Cp874	Cp1383
Cp273	Cp875	Cp33722
Cp277	Cp918	EUC_CN
Cp278	Cp921	EUC_JP
Cp280	Cp922	EUC_JP_LINUX
Cp284	Cp930	EUC_KR
Cp285	Cp933	EUC_TW
Cp297	Cp935	ISO2022JP
Cp420	Cp937	ISO8859_1
Cp424	Cp939	ISO8859_2
Cp437	Cp942	ISO8859_3
Cp500	Cp942C	ISO8859_4
Cp737	Cp943	ISO8859_5
Cp775	Cp943C	ISO8859_6
Cp838	Cp948	ISO8859_7
Cp850	Cp949	ISO8859_8
Cp852	Cp949C	ISO8859_1
Cp855	Cp950	ISO8859_9
Cp856	Cp964	ISO8859_13
Cp857	Cp970	ISO8859_15_FDIS
Cp860	Cp1006	MS874
Cp861	Cp1025	MS932
Cp862	Cp1026	SJIS
Cp863	Cp1097	TIS620
Cp864	Cp1098	UTF8
Cp865	Cp1112	UnicodeLittleUnmarked
Cp866	Cp1122	
Cp868	Cp1123	

Anhang B – Unterstützte Character-Encodings bei text/xml und text/html

ANSI_X3.4-1968 (US-ASCII)	EBCDIC-CP-IT	ISO-8859-1
Big5	EBCDIC-CP-NL	ISO-8859-2
EBCDIC-CP-AR1	EBCDIC-CP-NO	ISO-8859-3
EBCDIC-CP-AR2	EBCDIC-CP-ROECE	ISO-8859-4
EBCDIC-CP-CA	EBCDIC-CP-SE	ISO-8859-5
EBCDIC-CP-CH	EBCDIC-CP-US	ISO-8859-6
EBCDIC-CP-DK	EBCDIC-CP-YU	ISO-8859-7
EBCDIC-CP-ES	EUC-JP	ISO-8859-8
EBCDIC-CP-FI	EUC-KR	ISO-8859-9
EBCDIC-CP-FR	GB2312	Shift_JIS
EBCDIC-CP-GB	KOI8-R	UTF-8
EBCDIC-CP-HE	ISO-2022-JP	Windows-31J
EBCDIC-CP-IS	ISO-2022-KR	

Anhang C – XML-Schema zur Validierung von HTML-Daten

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema targetNamespace="http://www.w3.org/1999/xhtml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/1999/xhtml">
  <xs:group name="Heading.class">
    <xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="h1" />
        <xs:element ref="h2" />
        <xs:element ref="h3" />
        <xs:element ref="h4" />
        <xs:element ref="h5" />
        <xs:element ref="h6" />
      </xs:choice>
    </xs:choice>
  </xs:group>
  <xs:group name="List.class">
    <xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="ul" />
        <xs:element ref="ol" />
        <xs:element ref="dl" />
      </xs:choice>
    </xs:choice>
  </xs:group>
  <xs:group name="Block.class">
    <xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="p" />
        <xs:element ref="blockquote" />
        <xs:element ref="table" />
        <xs:element ref="hr" minOccurs="0" maxOccurs="unbounded" />
        <xs:group ref="List.class" />
      </xs:choice>
    </xs:choice>
  </xs:group>
  <xs:group name="Block.mix">
    <xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="Heading.class" />
        <xs:group ref="Text.class" />
        <xs:group ref="Block.class" />
      </xs:choice>
    </xs:choice>
  </xs:group>
  <xs:group name="Flow.mix">
    <xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:group ref="Block.class" />
        <xs:group ref="Text.class" />
      </xs:choice>
    </xs:choice>
  </xs:group>
</xs:schema>
```

```

<xs:group name="Text.class">
  <xs:choice>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="font" />
      <xs:element ref="br" />
      <xs:element ref="tt" />
      <xs:element ref="i" />
      <xs:element ref="b" />
      <xs:element ref="big" />
      <xs:element ref="sub" />
      <xs:element ref="sup" />
    </xs:choice>
  </xs:choice>
</xs:group>
<xs:simpleType name="Color">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Black" />
    <xs:enumeration value="black" />
    <xs:enumeration value="Silver" />
    <xs:enumeration value="silver" />
    <xs:enumeration value="Gray" />
    <xs:enumeration value="gray" />
    <xs:enumeration value="White" />
    <xs:enumeration value="white" />
    <xs:enumeration value="Maroon" />
    <xs:enumeration value="maroon" />
    <xs:enumeration value="Red" />
    <xs:enumeration value="red" />
    <xs:enumeration value="Purple" />
    <xs:enumeration value="purple" />
    <xs:enumeration value="Fuchsia" />
    <xs:enumeration value="fuchsia" />
    <xs:enumeration value="Green" />
    <xs:enumeration value="green" />
    <xs:enumeration value="Lime" />
    <xs:enumeration value="lime" />
    <xs:enumeration value="Olive" />
    <xs:enumeration value="olive" />
    <xs:enumeration value="Yellow" />
    <xs:enumeration value="yellow" />
    <xs:enumeration value="Navy" />
    <xs:enumeration value="navy" />
    <xs:enumeration value="Blue" />
    <xs:enumeration value="blue" />
    <xs:enumeration value="Teal" />
    <xs:enumeration value="teal" />
    <xs:enumeration value="Aqua" />
    <xs:enumeration value="aqua" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FontSize">
  <xs:restriction base="xs:string">
    <xs:enumeration value="-1" />
    <xs:enumeration value="+1" />
    <xs:enumeration value="+2" />
    <xs:enumeration value="+3" />
  </xs:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="+4" />
    <xs:enumeration value="2" />
    <xs:enumeration value="3" />
    <xs:enumeration value="4" />
    <xs:enumeration value="5" />
    <xs:enumeration value="6" />
    <xs:enumeration value="7" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FontFace">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Arial" />
    <xs:enumeration value="arial" />
    <xs:enumeration value="Times New Roman" />
    <xs:enumeration value="times new roman" />
    <xs:enumeration value="Verdana" />
    <xs:enumeration value="verdana" />
    <xs:enumeration value="courier new" />
    <xs:enumeration value="Courier New" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Length">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="1000" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="restrictedLength">
  <xs:restriction base="xs:string">
    <xs:pattern value="\\d{2}%" />
    <xs:pattern value="\\d{1}%" />
    <xs:pattern value="100%" />
    <xs:pattern value="\\d{1}" />
    <xs:pattern value="\\d{2}" />
    <xs:pattern value="\\d{3}" />
    <xs:pattern value="\\d{4}" />
    <xs:pattern value="10000" />
    <xs:pattern value="\\d{1}px" />
    <xs:pattern value="\\d{2}px" />
    <xs:pattern value="\\d{3}px" />
    <xs:pattern value="\\d{4}px" />
    <xs:pattern value="10000px" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Pixels">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="1000" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Number">
  <xs:restriction base="xs:nonNegativeInteger" />
</xs:simpleType>
<xs:complexType name="empty.type" />
<xs:attributeGroup name="CellVAlign.attrib">
  <xs:attribute name="valign">

```

```

    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="top" />
        <xs:enumeration value="middle" />
        <xs:enumeration value="bottom" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="CellHAlign.attrib">
  <xs:attribute name="align">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="left" />
        <xs:enumeration value="center" />
        <xs:enumeration value="right" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
<xs:complexType name="body.type" mixed="true">
  <xs:group ref="Block.mix" />
  <xs:attribute name="bgcolor" type="Color" />
  <xs:attribute name="text" type="Color" />
</xs:complexType>
<xs:element name="body" type="body.type" />
<xs:simpleType name="FPI">
  <xs:restriction base="xs:normalizedString" />
</xs:simpleType>
<xs:complexType name="html.type">
  <xs:sequence>
    <xs:element ref="head" minOccurs="0" />
    <xs:element ref="body" />
  </xs:sequence>
  <xs:attribute name="version" type="FPI" fixed="XHTML1.1" />
</xs:complexType>
<xs:element name="html" type="html.type" />
<xs:element name="title" type="xs:string" />
<xs:group name="head.content">
  <xs:sequence>
    <xs:element ref="title" minOccurs="0" />
  </xs:sequence>
</xs:group>
<xs:complexType name="head.type">
  <xs:group ref="head.content" />
</xs:complexType>
<xs:element name="head" type="head.type" />
<xs:complexType name="blockquote.type" mixed="true">
  <xs:group ref="Block.mix" />
</xs:complexType>
<xs:element name="blockquote" type="blockquote.type" />
<xs:complexType name="heading.type" mixed="true">
  <xs:group ref="Text.class" />
  <xs:attributeGroup ref="CellHAlign.attrib" />
</xs:complexType>
<xs:element name="h1" type="heading.type" />

```

```

<xs:element name="h2" type="heading.type" />
<xs:element name="h3" type="heading.type" />
<xs:element name="h4" type="heading.type" />
<xs:element name="h5" type="heading.type" />
<xs:element name="h6" type="heading.type" />
<xs:complexType name="p.type" mixed="true">
  <xs:group ref="Text.class" />
  <xs:attributeGroup ref="CellHAlign.attrib" />
</xs:complexType>
<xs:element name="p" type="p.type" />
<xs:complexType name="dt.type" mixed="true">
  <xs:group ref="Text.class" />
</xs:complexType>
<xs:element name="dt" type="dt.type" />
<xs:complexType name="dd.type" mixed="true">
  <xs:group ref="Flow.mix" />
</xs:complexType>
<xs:element name="dd" type="dd.type" />
<xs:group name="dl.content">
  <xs:choice>
    <xs:element ref="dt" />
    <xs:element ref="dd" />
  </xs:choice>
</xs:group>
<xs:complexType name="dl.type" mixed="true">
  <xs:group ref="dl.content" minOccurs="1" maxOccurs="unbounded" />
</xs:complexType>
<xs:element name="dl" type="dl.type" />
<xs:complexType name="li.type" mixed="true">
  <xs:group ref="Flow.mix" />
</xs:complexType>
<xs:element name="li" type="li.type" />
<xs:complexType name="ol.type">
  <xs:sequence>
    <xs:element ref="li" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:element name="ol" type="ol.type" />
<xs:complexType name="ul.type">
  <xs:sequence>
    <xs:element ref="li" minOccurs="1" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:element name="ul" type="ul.type" />
<xs:attributeGroup name="table.attlist">
  <xs:attribute name="width" type="restrictedLength" />
  <xs:attribute name="border" type="Pixels" />
  <xs:attribute name="cellspacing" type="Length" />
  <xs:attribute name="cellpadding" type="Length" />
  <xs:attribute name="bgcolor" type="Color" />
  <xs:attribute name="align">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="left" />
        <xs:enumeration value="center" />
        <xs:enumeration value="right" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>

```

```

        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:attributeGroup>
<xs:group name="table.content">
    <xs:choice>
        <xs:element ref="tr" minOccurs="1" maxOccurs="unbounded" />
    </xs:choice>
</xs:group>
<xs:complexType name="table.type" mixed="true">
    <xs:group ref="table.content" />
    <xs:attributeGroup ref="table.attlist" />
</xs:complexType>
<xs:element name="table" type="table.type" />
<xs:attributeGroup name="tr.attlist">
    <xs:attribute name="bgcolor" type="Color" />
    <xs:attributeGroup ref="CellHAlign.attrib" />
    <xs:attributeGroup ref="CellVAlign.attrib" />
</xs:attributeGroup>
<xs:group name="tr.content">
    <xs:choice>
        <xs:element ref="th" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="td" minOccurs="0" maxOccurs="unbounded" />
    </xs:choice>
</xs:group>
<xs:complexType name="tr.type">
    <xs:group ref="tr.content" />
    <xs:attributeGroup ref="tr.attlist" />
</xs:complexType>
<xs:element name="tr" type="tr.type" />
<xs:attributeGroup name="td.attlist">
    <xs:attribute name="nowrap" fixed="nowrap" />
    <xs:attribute name="bgcolor" type="Color" />
    <xs:attribute name="text" type="Color" />
    <xs:attribute name="rowspan" type="Length" default="1" />
    <xs:attribute name="colspan" type="Length" default="1" />
    <xs:attributeGroup ref="CellHAlign.attrib" />
    <xs:attributeGroup ref="CellVAlign.attrib" />
</xs:attributeGroup>
<xs:complexType name="td.type" mixed="true">
    <xs:group ref="Block.mix" />
    <xs:attribute name="height" type="restrictedLength" />
    <xs:attribute name="width" type="restrictedLength" />
    <xs:attributeGroup ref="td.attlist" />
</xs:complexType>
<xs:element name="td" type="td.type" />
<xs:attributeGroup name="th.attlist">
    <xs:attribute name="nowrap" fixed="nowrap" />
    <xs:attribute name="bgcolor" type="Color" />
    <xs:attribute name="width" type="restrictedLength" />
    <xs:attribute name="height" type="restrictedLength" />
    <xs:attribute name="rowspan" type="Length" default="1" />
    <xs:attribute name="colspan" type="Length" default="1" />
    <xs:attributeGroup ref="CellHAlign.attrib" />
    <xs:attributeGroup ref="CellVAlign.attrib" />
</xs:attributeGroup>

```

```
<xs:complexType name="th.type" mixed="true">
  <xs:group ref="Block.mix" />
  <xs:attributeGroup ref="th.attlist" />
</xs:complexType>
<xs:element name="th" type="th.type" />
<xs:element name="br" type="empty.type" />
<xs:element name="b" type="xs:string" />
<xs:element name="big" type="xs:string" />
<xs:element name="i" type="xs:string" />
<xs:element name="sub" type="xs:string" />
<xs:element name="sup" type="xs:string" />
<xs:element name="tt" type="xs:string" />
<xs:element name="hr" type="empty.type" />
<xs:attributeGroup name="font.attlist">
  <xs:attribute name="size" type="FontSize" />
  <xs:attribute name="color" type="Color" />
  <xs:attribute name="face" type="FontFace" />
</xs:attributeGroup>
<xs:complexType name="font.type" mixed="true">
  <xs:group ref="Text.class" minOccurs="0" maxOccurs="unbounded" />
  <xs:attributeGroup ref="font.attlist" />
</xs:complexType>
<xs:element name="font" type="font.type" />
</xs:schema>
```

Anhang D – Zulässige Farben und unzulässige Farbkombination bei der Anzeige von HTML-Daten

1. Zulässige Farben

Folgende Farben können wahlweise als Text- und Hintergrundfarbe verwendet werden:

Black
Silver
Gray
White
Maroon
Red
Purple
Fuchsia
Green
Lime
Olive
Yellow
Navy
Blue
Teal
Aqua

2. Unzulässige Farbkombinationen

Die Verwendung einer Farbe sowohl für den Text als auch für den Hintergrund wird ausgeschlossen. Zusätzlich werden noch folgende Farbkombinationen ausgeschlossen:

Navy	Black
Olive	Gray
Yellow	White
Purple	Maroon
Red	Fuchsia
Teal	Green
Aqua	Lime