

## eCH-0018 XML Best Practices

<b>Name</b>	XML Best Practices
<b>eCH-Nummer</b>	eCH-0018
<b>Kategorie</b>	<b>Standard</b>
<b>Reifegrad</b>	<b>Definiert</b>
<b>Version</b>	2.0
<b>Status</b>	<b>Genehmigt</b>
<b>Beschluss am</b>	2018-09-05
<b>Ausgabedatum</b>	2018-09-11
<b>Ersetzt Version</b>	1.0 < Major Change >
<b>Voraussetzungen</b>	-
<b>Beilagen</b>	-
<b>Sprachen</b>	Deutsch (Original), Französisch (Übersetzung)
<b>Autoren</b>	Fachgruppe XML Willy Müller, Informatiksteuerungsorgan Bund (ISB), <a href="mailto:Willy.Mueller@ISB.admin.ch">Willy.Mueller@ISB.admin.ch</a>
<b>Herausgeber / Vertrieb</b>	Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich T 044 388 74 64, F 044 388 71 80 <a href="http://www.ech.ch">www.ech.ch</a> / <a href="mailto:info@ech.ch">info@ech.ch</a>

### Zusammenfassung

Das vorliegende Dokument beschreibt Regeln und Empfehlungen für das Design von XML-Schemas und Namespaces sowie deren Nutzung im Kontext von eCH-Standard.

## Inhaltsverzeichnis

<b>Zusammenfassung</b> .....	<b>1</b>
<b>Inhaltsverzeichnis</b> .....	<b>2</b>
<b>Hinweis</b> .....	<b>3</b>
<b>1 Einleitung</b> .....	<b>4</b>
1.1 <b>Status</b> .....	4
1.2 <b>Anwendungsgebiet</b> .....	4
1.3 <b>Terminologie der Empfehlungen</b> .....	4
1.4 <b>Normative Referenzen</b> .....	5
<b>2 Allgemeine Grundlagen</b> .....	<b>5</b>
2.1 <b>Modell</b> .....	5
2.2 <b>XML Dokumente als Instanzen von Klassen</b> .....	6
<b>3 Design von XML-Schemas</b> .....	<b>6</b>
3.1 <b>Strukturierung von Schemas und Schema-Dokumenten</b> .....	7
3.2 <b>Allgemeine Namenskonventionen</b> .....	7
3.3 <b>XML-Namespace</b> .....	7
3.4 <b>Abbildung von Datenstrukturen auf XML</b> .....	9
3.4.1 <b>Elemente und Attribute</b> .....	9
3.4.2 <b>Repräsentation leerer Werte</b> .....	10
3.4.3 <b>Umgang mit Identifikation und Referenzen</b> .....	10
3.5 <b>Binäre Daten</b> .....	10
3.6 <b>Dokumentation</b> .....	11
3.7 <b>Mehrsprachigkeit</b> .....	11
3.7.1 <b>Die Sprache der Namen</b> .....	11
3.7.2 <b>Sprachmarkierung von Inhalten</b> .....	12
3.7.3 <b>Aufzählungswerte (XSD-Enumeration)</b> .....	12
3.8 <b>Umgang mit Wertelisten</b> .....	12
3.9 <b>Umgang mit Substitution</b> .....	13
3.9.1 <b>Element und Type Substitution</b> .....	13
3.9.2 <b>Substitution Groups</b> .....	13
3.10 <b>Versionierung</b> .....	14
3.11 <b>Erweiterbarkeit</b> .....	15
<b>4 XML-Instanzen</b> .....	<b>15</b>
4.1.1 <b>Zeichensätze (Character Encodings)</b> .....	15
4.2 <b>Character References</b> .....	16
4.3 <b>Entity declarations and references</b> .....	16
4.4 <b>Umgang mit Whitespace (Zeilenumbrüche etc.)</b> .....	16
4.5 <b>Zugehörigkeit zu einem XML-Schema</b> .....	16
<b>5 Sicherheitsüberlegungen</b> .....	<b>16</b>

---

<b>6</b>	<b>Haftungsausschluss/Hinweise auf Rechte Dritter .....</b>	<b>18</b>
<b>7</b>	<b>Urheberrechte .....</b>	<b>18</b>
	<b>Anhang A – Referenzen &amp; Bibliographie.....</b>	<b>19</b>
	<b>Anhang B – Mitarbeit &amp; Überprüfung .....</b>	<b>20</b>
	<b>Anhang C – Abkürzungen und Glossar .....</b>	<b>20</b>
	<b>Anhang D – Änderungen gegenüber Vorversion .....</b>	<b>22</b>
	<b>Anhang E – Abbildungsverzeichnis .....</b>	<b>23</b>

## **Hinweis**

Aus Gründen der besseren Lesbarkeit und Verständlichkeit wird im vorliegenden Dokument bei der Bezeichnung von Personen ausschliesslich die maskuline Form verwendet. Diese Formulierung schliesst Frauen in ihrer jeweiligen Funktion ausdrücklich mit ein.

# 1 Einleitung

## 1.1 Status

*Genehmigt:* Das Dokument wurde vom Expertenausschuss genehmigt. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

## 1.2 Anwendungsgebiet

Der vorliegende Standard definiert Richtlinien für die Verwendung und Erstellung von XML-Dokumenten und XML-Schemas [XMLSchema1], [XMLSchema2] für den Austausch strukturierter Daten. Ferner definiert er Richtlinien für die Beschreibung von XML-Namespaces [XMLNS]. Neben allgemein gültigen Richtlinien definiert er solche, die bei der Erstellung von XML-Schemas für eCH-Standards speziell zu beachten sind.

Die Richtlinien gelten nicht für XML-Schemas und –Instanzen zur Strukturierung von Texten (docbook, xhtml, oder ähnliche).

## 1.3 Terminologie der Empfehlungen

Richtlinien in diesem Dokument werden gemäss der Terminologie aus [RFC2119] angegeben, dabei kommen die folgenden Ausdrücke zur Anwendung, die durch GROSSSCHREIBUNG als Wörter mit den folgenden Bedeutungen kenntlich gemacht werden (Zitat aus RFC 2119):

- **MUST:** This word, or the terms "**REQUIRED**" or "**SHALL**", mean that the definition is an absolute requirement of the specification.
- **MUST NOT:** This phrase, or the phrase "**SHALL NOT**", mean that that definition is an absolute prohibition of the specification.
- **SHOULD:** This word, or the adjective "**RECOMMENDED**", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- **SHOULD NOT:** This phrase, or the phrase "**NOT RECOMMENDED**" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- **MAY:** This word, or the adjective "**OPTIONAL**", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option **MUST** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option **MUST** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

**Für das** Erstellen von eCH-Standards und deren XML-Schemas gelten zum Teil strengere Vorgaben. Wird bei Empfehlungen nichts Spezifisches zu eCH erwähnt, so gelten diese

auch für das Schreiben von eCH-Standards und deren XML-Schemas. Gelten für den Bereich eCH strengere Vorgaben, so sind diese jeweils im Kapitel „**Empfehlungen für eCH-Standards**“ ersichtlich.

## 1.4 Normative Referenzen

Die folgenden Vorgaben sind aus Sicht des eCH-0018 als normativ zu erachten. Siehe Anhang A:

[ISO3166], [ISO639], [XMLSchema1], [XMLSchema2], [XMLNS]

## 2 Allgemeine Grundlagen

### 2.1 Modell

Die nachfolgende Grafik gibt einen groben Überblick hinsichtlich der im eCH-0018 erwähnten Objekte.

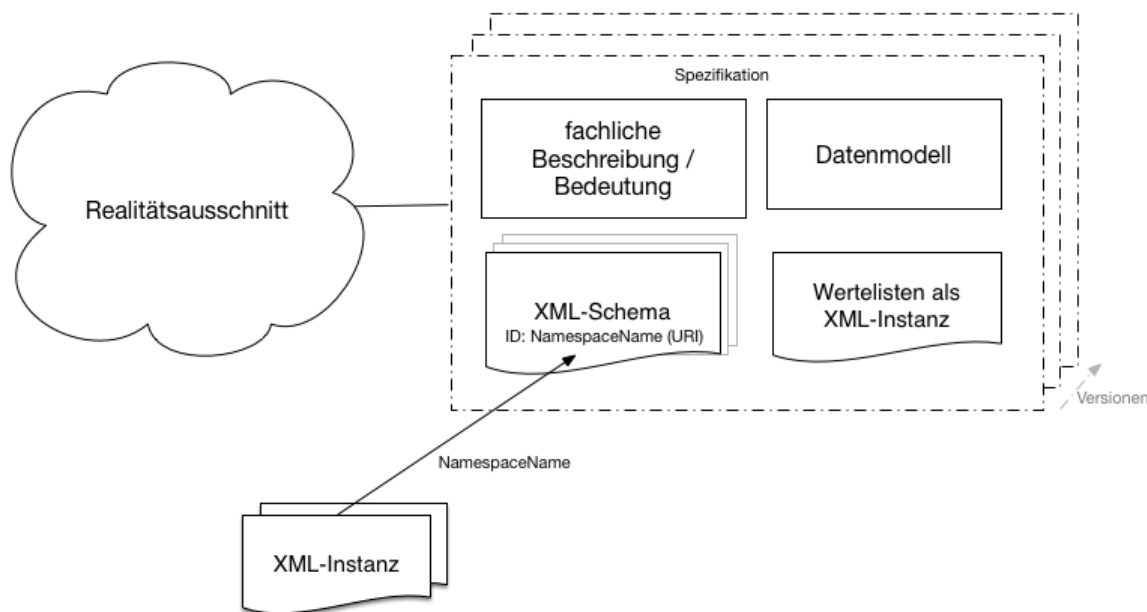


Abbildung 1: Übersicht der wichtigsten Objekte

Für die Beschreibung eines für den Datenaustausch oder eine Verarbeitung relevanten Realitätsausschnitts bedarf es einer Spezifikation, welche die Semantik der betrachteten Elemente beschreibt. Eine solche Spezifikation besteht in der Regel aus einer Kombination aus einer fachlichen Beschreibung (im Falle von eCH dem Standard-Dokument), einem Datenmodell mit Klassen/Entitäten und deren Beziehungen (bei eCH ebenfalls als Bestandteil des

Standard-Dokuments), einem oder mehreren XML-Schemen und ggf. spezifischen Wertelisten als XML-Instanzen. Wird von XML-Dokumenten gesprochen, so sind damit sowohl XML-Schemen wie auch XML-Instanz-Dokumente gemeint. Die Beziehung zwischen einem XML-Instanz-Dokument und dem entsprechenden XML-Schema wird über den Namespace-Namen hergestellt. Die Identifikation des XML-Schemas erfolgt ebenfalls durch den Namespace-Namen welcher als URI festgehalten wird.

Jede Anpassung an einem der Bestandteile der Spezifikation führt zu einer neuen Version der Spezifikation.

## 2.2 XML Dokumente als Instanzen von Klassen

XML-Dokumente sind Textdokumente, die gemäss dem XML-Standard wohl geformt sind. In unserem Kontext unterscheiden wir zwei Typen von XML-Dokumenten:

XML-Schemas	XML-Dokument, das die Regeln des Standards [XMLSchema] erfüllt und damit XML-Instanzen beschreibt.
XML-Instanzen	XML-Dokument, das die Regeln eines bestimmten XML-Schemas erfüllt.

Generell gilt:

- **SHOULD:** Zur maschinellen Verarbeitung von getypten Daten (d.h. nahezu allen Daten in Business-to-Business-Szenarien) sollte das Schema mittels eines XML-Schema definiert werden, da nur in diesem die Datentypen in einer anwendungsnahe Form definiert werden können.
- **SHOULD:** XML-Dokumente sollen mindestens durch ein XML-Schema beschrieben werden. Dies ermöglicht den Benutzern die Regeln, nach welchen entsprechende Instanz-Dokumente erstellt werden, zu verstehen. Ausserdem gibt es dafür Werkzeuge, die automatisiert prüfen können, ob ein Instanz-Dokument dem zugeordneten XML-Schema entspricht.

### Empfehlungen für eCH-Standards

- **MUST:** XML-Dokumenten müssen durch ein XML-Schema beschrieben werden.

## 3 Design von XML-Schemas

Das Kapitel behandelt die innere Struktur von XML-Schemas, d.h. den Aufbau und den Zusammenhang der sogenannten "XML-Schema Komponenten".

- **MUST:** Zu jedem XML-Schema gehört eine Dokumentation, welche die fachliche Semantik der Schema-Komponenten und den vorgesehenen Anwendungskontext beschreibt.
- **SHOULD:** Es soll `elementFormDefault="qualified"` verwendet werden.
- **SHOULD NOT:** Für Elemente und Attribute sollen keine Defaultwerte definiert werden.

## 3.1 Strukturierung von Schemas und Schema-Dokumenten

Für die Strukturierung von Schema-Dokumenten gilt:

- **SHOULD:** Ein Schema soll in einem Schema-Dokument implementiert werden.
- **SHOULD:** Zur Wiederverwendung von bestehenden Definitionen soll der ***xs:import*** Mechanismus verwendet werden.
- **SHOULD NOT:** ***xs:include*** sollen nicht verwendet werden, weil so wiederverwendbare Komponenten nicht klar über den Namespace getrennt sind.
- **SHOULD NOT:** Der ***xs:redefine*** Mechanismus von XML-Schema sollte vermieden werden, da er zu komplizierten Abhängigkeiten zwischen verschiedenen Schema-Dokumenten führt und vor allem bei späteren Änderungen von Schema-Dokumenten zu schwierig auffindbaren Fehler führen kann.

## 3.2 Allgemeine Namenskonventionen

Um die Lesbarkeit von XML-Dokumenten zu erhöhen und eine allgemein gültige Grundlage für alle XML-Entwickler zu schaffen, ist eine Standardisierung der Struktur der verwendeten Namen (und nicht nur ihrer Sprache) wichtig.

- **MUST:** Namen bestehen ausschliesslich aus ASCII Gross- und Kleinbuchstaben (a-z und A-Z), die Ziffern 0-9, Underscore (\_), Punkt (.), und Bindestrich (-).
- **SHOULD:** Setzen sich Namen von Schemapartikeln aus mehreren Worten zusammen, so werden sie in Camel Case-Notation geschrieben.  
Beispiel: `<recordNumber>12345678</recordNumber>`
- **SHOULD:** Abkürzungen sollen nur verwendet werden, wenn es sich um gängige, in der Fachdomäne bekannte Abkürzungen handelt
- **SHOULD:** Bei Abkürzungen soll die etablierte Schreibweise unverändert übernommen werden.
- **MUST:** Namen von Simple und Complex Types enden auf "Type".
- **MUST:** Namen von Attribute Groups enden auf "AttributeGroup".
- **MUST:** Namen von Named Model Groups enden auf "Group".

### Empfehlungen für eCH-Standards

- **SHOULD:** Abkürzungen sollen im eCH-Standard erklärt werden, zudem sollen sie im XML-Schema mittels Annotation beschrieben werden.

## 3.3 XML-Namespace

XML-Namespace ist ein Mechanismus, um zu gewährleisten, dass nicht identische, aber gleich benannte Elemente eindeutig referenziert werden können. Durch Kombination des Namespace-Namen und des lokal verwendeten Names (local name) wird der lokale Name eindeutig. Namespace-Namen sind URIs, müssen jedoch nicht zwingend auf eine existierende Ressource verweisen.

Die Deklaration von XML Namespaces in einer XML-Instanz geschieht durch ***xmlns*** Attribute beliebiger Elemente in XML Dokumenten. Einem Namespace wird damit entweder ein Präfix zugeordnet (***xmlns:prefix="uri"***, wobei ***prefix*** ein beliebiger XML Name ohne Doppelpunkte

sein darf und *uri* der Namespace-Name ist), oder es ist der Default Namespace (*xmlns="uri"*, wobei *uri* der Namespace Name ist).

Die Namespace Präfixe werden verwendet, um in einem XML Dokument den Bezug von einem qualifizierten Namen (*<html:title>*) zur Namespace-Deklaration (*xmlns:html="http://www.w3.org/1999/xhtml"*) herzustellen.

**ACHTUNG:** Der Default Namespace (deklariert mit *xmlns="..."*) gilt nicht für Attribute, d.h. Attribute ohne Präfix sind nie einem Namespace zugeordnet.

Im XML-Schema wird der XML-Namespace für die durch das Schema beschriebenen XML-Instanzen durch das Attribut *target-namespace* definiert.

- **MUST:** Jedes XML-Schema muss seinen eigenen XML-Namespace definieren. Auf den von einem Schema definierten Namespace muss mit dem *targetNamespace*-Attribut des *xs:schema*-Elementes verwiesen werden.
- **SHOULD:** Der Namespace-Name (=der URI) eines XML-Schemas soll nicht direkt auf das Schema zeigen, sondern als http-URL auf eine Beschreibung des Namespace.
- **MUST:** Der für den Namespace-Namen verwendete Domainname muss Eigentum der schema-definierenden Stelle sein.
- **SHOULD:** Für den Namespace-Namen eines Schemas sollte eine schemaspezifische Subdomain verwendet werden (siehe nachfolgendes Beispiel).
- **MUST:** Das XML-Schema muss im http-URL Pfad unterhalb der Namespace-URL für den Menschen auffindbar sein.
- **MUST:** Die Schema-Validierung muss mit lokal abgelegten XML-Schemas erfolgen. Die Verfügbarkeit des Servers mit der Namespace-URL ist nicht sichergestellt.
- **SHOULD:** Ein Namespace-Name soll aus folgenden Elementen zusammengesetzt werden:

http	://	<a href="http://www.ech.ch">www.ech.ch</a>	/	xmlns	/	eCh-0018	/	1
URI Sche me		Domain Name		allgemeiner Name für Namespace- Beschreibun- gen		Anwendungs- fall		Ver- sion

Die Ablage von XML-Schemen sollte unter einer spezifischen Subdomain erfolgen und nicht unter *www...* Dadurch lassen sich Konflikte mit einem CMS (Content Management System) vermeiden.

Bsp. *schemas.geo.admin.ch* statt *www.geo.admin.ch /xmlns*

Um auch bei lokal abgelegten XML-Schemas die Validierung auf Basis der Namespace-URL (ohne lokale Pfade) zu ermöglichen, kann der Mechanismus XML Catalog (OASIS) verwendet werden.

**Empfehlungen für eCH-Standards**

- **MUST:** Der Namespace-Name für eCH-Standards muss mit <http://www.ech.ch/xmlns> beginnen, gefolgt von der Nummer des entsprechenden eCH-Standards.



Bsp. <https://www.ech.ch/xmlsn/eCH-0018>

- **SHOULD:** Die Version des Namespaces muss am Namespace-Namen erkennbar sein.

Bsp. <https://www.ech.ch/xmlsn/eCH-0018/2>

### 3.4 Abbildung von Datenstrukturen auf XML

Die folgenden Abschnitte beschreiben verschiedene Aspekte, die bei der Abbildung von Datenstrukturen auf XML wichtig sind, und geben Empfehlungen für besser und Information über weniger gut geeignete Mechanismen, mit diesen Problemen umzugehen.

- **SHOULD:** Die Abbildung des Austauschdatenmodells (Entitäten) und jene der Schnittstelle (auszutauschende Ereignisse, bzw. Schnittstellenformat) soll in getrennten XML-Schemas erfolgen.

#### Empfehlungen für eCH-Standards

- **MUST:** Datenstandards und Schnittstellenstandards sind zu trennen.
- **SHOULD:** Identifikatoren, Entitäten und Schnittstellen sollen je separat definiert werden.
- **SHOULD NOT:** Auf unspezifische Basis-Schemas (in Sinne von Commons) ist zugunsten von dedizierten Basis-Schemas mit klarer Kohäsion zu verzichten.

#### 3.4.1 Elemente und Attribute

Obwohl der Einsatz von Attributen zu technisch kleineren XML-Dokumentinstanzen führt, ist der Einsatz von Elementen vorzuziehen. Durch den Einsatz von Elementen lassen sich einfachere und vor allem wiedererkennbare Strukturen abbilden.

Nachteil der Nutzung von Attributen bestehen darin, dass:

- die Reihenfolge von Attributen nicht definiert werden kann,
- keine Strukturierungsmöglichkeit besteht,
- eine zukünftige Typ-Änderung führt bei der Verwendung von Attributen in den Instanzen zu einer grundsätzlichen Strukturänderung,

was zu Problemen bei der Verarbeitung entsprechender XML-Instanzen führen kann. Aus dieser Perspektive besteht bei XML Schema kein Grund überhaupt noch Attribute zu verwenden, ausser dort, wo sie extern vorgegeben sind durch bestehende Standards (z.B. XML Namespaces oder XLink) oder andere Vereinbarungen.

- **SHOULD:** Generell sollten Elemente zur Strukturierung verwendet werden, da diese im Bedarfsfall einfacher erweitert werden können, z.B. mit zusätzlichen Attributen, oder durch die Unterteilung in feiner gegliederte Strukturen.
- **MUST NOT:** Fachliche Aspekte dürfen nicht mittels Attributen abgebildet werden. Die Reihenfolge von Attributen lässt sich nicht spezifizieren. Müssen sie später strukturiert werden, muss die Verarbeitung entsprechender XML-Instanzen stark überarbeitet werden.
- **MAY:** Attribute sollen höchstens für Aspekte der Metadaten oder technischen Aspekten von XML-Dokumenten verwendet werden.

- **MUST NOT:** Default-Werte von Attributen dürfen im Schema nicht definiert werden, weil der Parser ohne Kenntnis des Schemas der Applikation die falschen Daten liefert.
- **SHOULD:** Aus Sicherheitsgründen soll Max-Occurs nicht unbounded definiert werden und keine rekursiven Strukturen verwendet werden.
- **SHOULD:** Aus Sicherheitsgründen sollen Maximallängen bei Textelementen definiert werden.
- **SHOULD NOT:** Aktive Inhalte sollten nicht ausgetauscht werden.
- **SHOULD:** Durch geeignete Schemadefinitionen soll verhindert werden, dass leere Elemente und Attribute codiert werden können (Bsp. minLength oder token)

### 3.4.2 Repräsentation leerer Werte

Oftmals besteht in einem Anwendungskontext die Anforderung, dass gewisse Angaben optional sind, also in Instanzen nicht zwingend angegeben werden müssen. XML-Schema bietet die Möglichkeit dies mittels „xsi:nil“ Mechanismus oder mittels optionaler Elemente abzubilden.

- **SHOULD:** Die Darstellung von Null-Werten sollte mit optionalen Elementen erfolgen, d.h. durch Nichterscheinen der jeweiligen Komponenten.
- **SHOULD NOT:** "*xsi:nil*" sollte vermieden werden.
- **SHOULD:** Hat die Abwesenheit eines Wertes eine fachliche Bedeutung, so soll dies durch einen expliziten Wert abgebildet werden (Bsp. "nicht bekannt").

### 3.4.3 Umgang mit Identifikation und Referenzen

Häufig gibt es in Datenmodellen Beziehungen zwischen einzelnen Entitäten/Klasse. Eine Art, diese Anforderung technisch umzusetzen, ist die Vergabe von Identifikatoren/Namen, mit denen die zu identifizierenden Entitäten/Objekte benannt werden, und die in Referenzen verwendet werden können.

- **SHOULD NOT:** Referenzen sollen nicht verwendet werden.
- **SHOULD NOT:** *QName* soll nicht als Referenz verwendet werden.
- **SHOULD:** Eindeutigkeitsbedingungen (unique) sollen dokumentiert werden.
- **SHOULD NOT:** Für die Sicherstellung von Identity Constraints (IDREF und keyref) sollen nicht XML-, sondern applikatorische Mechanismen verwendet werden.

## 3.5 Binäre Daten

XML ist hauptsächlich auf textlich repräsentierbare Daten ausgerichtet. Für binäre Daten definiert XML Schema die Datentypen *xs:hexBinary* (hexadezimale Ziffern als Zeichen) und *xs:base64Binary* (binäre Daten dargestellt als Buchstaben gemäss [RFC2045]). Die *xs:base64Binary* Darstellung ist kompakter als die *xs:hexBinary* Darstellung.

- **SHOULD:** Umfangreicheren Nutzdaten sollen als BASE64 ausgetauscht werden und entsprechend im Schema als base64Binary modelliert werden.

Falls die binären Daten, z.B. aufgrund ihrer Grösse, beim Transport oder der Verarbeitung

Probleme bieten, ist eine Umgehungslösung zu bevorzugen, die aus Sicht des XML-Schemas transparent ist.

- z.B. bei Transport – aufteilen in Pakete
- z.B. bei der Verarbeitung – ersetzen der binären Daten durch interne Referenzen.

## 3.6 Dokumentation

Ein XML-Schema definiert die syntaktische Struktur und beschränkt den Zweck und die Bedeutung von XML-Dokumenten. Der Zweck und die Bedeutung der Syntax soll daher zusätzlich explizit beschrieben werden. Dies kann durch Dokumentation im Schema selber oder in begleitender Dokumentation erfolgen.

- **SHOULD:** Falls direkt im Schema dokumentiert wird, so soll dies kurz und prägnant im `xs:documentation` erfolgen.
- **SHOULD:** Falls direkt im Schema dokumentiert wird, so soll dies in der gleichen Sprache der Elemente / Elementnamen erfolgen.
- **MUST NOT:** Dokumentation in XML-Schemas darf nicht in XML Kommentaren (`<!-- ... -->`) geführt werden.

### Empfehlungen für eCH-Standards

- **MUST:** Der Zweck und die Bedeutung aller Elemente muss im Standarddokument beschrieben werden.

## 3.7 Mehrsprachigkeit

### 3.7.1 Die Sprache der Namen

Wenn die Schemas mit lokalisierten Namen entwickelt werden, besteht die Gefahr, dass sie nur in dieser spezifischen Sprachregion akzeptiert werden. Um der Mehrsprachigkeit der Schweiz, aber auch internationalen Anforderungen, gerecht zu werden, muss eine weltweit gültige Lösung angestrebt werden. Zudem wird dadurch die Konsistenz bei der Kombination verschiedener Schemas erhöht, die durch das Auftreten verschiedensprachiger Namen im gleichen XML Dokument stark leiden würde.

- **SHOULD:** Die Sprache der Namen ist Englisch.
- **MAY:** Falls im Schema Namen verwendet werden, die eine sprachspezifische Bedeutung haben, die sich nicht übersetzen lässt (z.B. juristische Fachwörter), so können diese in der Sprache geschrieben werden, für die die Bedeutung erhalten bleiben muss.

### Empfehlungen für eCH-Standards

- **MUST:** Die Sprache der Namen ist Englisch, sofern für den Begriff eine englische Übersetzung verfügbar ist.

### 3.7.2 Sprachmarkierung von Inhalten

Sprachmarkierungen erlauben Inhalte von Elementen eindeutig mit der Sprache des Inhaltes zu bezeichnen. Dies erlaubt mehrsprachige Inhalte eindeutig und normiert darzustellen (Ländercodes ISO 3166, Sprachcodes ISO 639).

- **SHOULD:** Sprachangaben sollen als Elemente umgesetzt werden (Value-Pair) wobei das eine Element die Kennzeichnung der Sprache, bestehend aus Sprache und Land (Bsp. „de-CH“) und das andere den eigentlichen Text in der jeweiligen Sprache enthält.  

```
<voteDescriptionInfo>  
  <language>de-CH</language>  
  <voteDescription>Volksabstimmung vom 10.06.2018</voteDescription>  
</voteDescriptionInfo>
```
- **SHOULD NOT:** Sprach-Codierungen mittels *xml:lang*-Attribut sind zu vermeiden.  
**MUST:** Sprach-Codierungen sind gemäss RFC 3066 [RFC3066] darzustellen. Applikationen müssen die zweistelligen ISO 639 [ISO639] Codierungen (de, fr, it, rm, en) oder die kombinierten ISO 639/3166 [ISO3166] Codierungen (de-CH, en-US) verwenden.

### 3.7.3 Aufzählungswerte (XSD-Enumeration)

Für die durch das Schema vorgegebenen Aufzählungswerte gilt:

- **SHOULD:** Aufzählungswerte (enumeration) sollen sprechende Namen verwenden.
- **SHOULD:** Die Sprache der Aufzählungswerte ist Englisch (konsistent mit der Sprache der Elemente)
- **MAY:** Falls im Schema Aufzählungswerte verwendet werden, die eine sprachspezifische Bedeutung haben, die sich nicht übersetzen lässt (z.B. juristische Fachwörter), so können diese in der Sprache geschrieben werden, für die die Bedeutung erhalten bleiben muss.

## 3.8 Umgang mit Wertelisten

Für Wertelisten, die nicht als XSD-Enumeration abgebildet werden können, gilt:

- **MUST:** Es muss entschieden werden, ob Wertelisten zusammen mit dem XML-Schema versioniert werden sollen oder nicht.
- **SHOULD NOT:** Wenn die Werte der Liste häufig ändern, so soll diese nicht im XML-Schema enthalten sein.
- **MUST:** Wenn die Werteliste Teil des XML-Schemas ist, so bedingt jede Änderung eine neue Version des Namespace-Namens.
- **SHOULD NOT:** Für den Typ des Werts soll nicht XSD-QName verwendet werden.

## 3.9 Umgang mit Substitution

### 3.9.1 Element und Type Substitution

Beim Einsatz von Vererbung zur Abbildung von fachlichen Begriffshierarchien können die folgenden XML-Schema Konstrukte angewendet werden.

- Element Substitution
- Type Substitution

Generell ist der Einsatz von Vererbung im XML Umfeld nicht gebräuchlich. In Spezialfällen, welche sich konsequent am objektorientierten Paradigma ausrichten, können sie vorkommen. Dies vor allem beim Einsatz von programmieretechnischen Frameworks, wie der XML-Serialisierung von JAVA oder .NET

- **SHOULD NOT:** Auf Vererbungsmechanismen sollte möglichst verzichtet werden. Sollten dies trotzdem notwendig sein, so ist dies auf Type-Substitution zu beschränken.
- **MUST:** Ist im Schema explizit dokumentiert oder sogar durch das Schema-Design erzwungen (Typen als *abstract="true"* definiert), das *xsi:type* verwendet werden kann oder soll, so ist dies zulässig, muss aber angemessen dokumentiert werden. In der Verarbeitung dieser Instanzen ist dann darauf zu achten, dass diese typbasiert erfolgt, so dass die Typzuweisung nicht über den Typ eines Elements im Schema, sondern über den *xsi:type* aus der Instanz erfolgt.
- **MUST:** Werden Mechanismen der Type Substitution verwendet, so ist dies deutlich und angemessen zu dokumentieren, so dass Benutzer des Schemas deutlich über diesen Aspekt des Schemas informiert werden.
- **MUST:** Wird einer der beiden Mechanismen eingesetzt (*xsi:type* oder *substitution-Group*), so darf nur dieser Mechanismus eingesetzt werden. Eine Mischung beider Stile würde ein Schema schwer zu verstehen und den Umgang damit unnötig kompliziert machen.

### 3.9.2 Substitution Groups

In einem XML-Schema kann ein Element explizit mit dem *substitutionGroup*-Attribut angeben, dass es das referenzierte Element in dieser Instanz ersetzt. Es darf dann in einer Instanz überall dort vorkommen, wo das im *substitutionGroup*-Attribut referenzierte Element erlaubt ist. Der Mechanismus kann auch über mehrere Ebenen definiert werden.

- **SHOULD NOT:** Generell sollten Mechanismen der Substitution Groups vermieden werden, solange nicht gewichtige Gründe dafürsprechen. Diese Mechanismen erschweren die Verarbeitung und das Verständnis des Schemas und der durch das Schema definierten Instanzen.
- **MUST:** Werden Mechanismen der Substitution Groups verwendet, so ist dies deutlich und angemessen zu dokumentieren, so dass Benutzer des Schemas deutlich über diesen Aspekt des Schemas informiert werden.

Der Verzicht auf Substitution Groups bringt folgende Vorteile mit sich:

- es führt im Instanz-Dokument – sofern keine Vererbung notwendig ist – zu einfacheren Instanzen.

## 3.10 Versionierung

XML-Anwendungen können eine lange Lebensdauer haben und Änderungen unterliegen, die eine Koexistenz verschiedener Versionen der zugrundeliegenden Schemas bedingen. Um die Identifikation der Schemaversion zu ermöglichen, müssen Schemas versioniert werden. Für die Versionierung von XML-Schemas gelten folgende Regeln:

- **MUST:** Zwischen der Namespace-Version und der Schema-Version muss eine 1:1 Beziehung bestehen. D.h. der Namespace-Name identifiziert das Schema eindeutig.
- **MUST:** Änderungen an der Syntax und redaktionelle Anpassungen, welche die Semantik ändern, führen zu einer neuen Version und damit zu einem neuen Namespace-Namen.  
Bsp.: (../eCH-0018/1) wird zu (../eCH-0018/2)
- **MUST NOT:** Rein redaktionelle Anpassungen ohne Konsequenz auf die Semantik (z.B. die Korrektur eines Tippfehlers in einem Kommentar) führen im Namespace zu keiner neuen Version. (Sie können über andere Mittel, z.B. einem neuen Ausgabedatum, kenntlich gemacht werden).

Damit die Applikationen die richtige Verarbeitung einleiten können, muss die Version in der Instanz vermerkt sein:

- **MUST:** XML Dokumente müssen ausschliesslich via Namespace-Namen den Bezug zur Schemaversion herstellen.
- **SHOULD NOT:** XML-Dokumente sollen keinen Bezug zum Ausgabedatum des Schemas herstellen.

Die Angabe der Version im Sinne des eCH-0018 enthält keine Information bezüglich einer eventuell vorhandenen Abwärtskompatibilität. Mit Bezug zum Thema Abwärtskompatibilität gilt es aber folgendes zu beachten:

Damit von Abwärtskompatibilität gesprochen werden kann, muss folgende Bedingung erfüllt sein. Abwärtskompatibilität ist immer nur unidirektional (Sender mit alter Version an Empfänger mit neuer Version).

Eine neue, abwärtskompatible Version eines XML-Schemas muss eine XML-Datei, die mit einer vorangehenden XML-Schema Version erstellt wurde, ohne Fehler validieren und korrekt verarbeiten können.

*Bsp. Bei einem in der neuen Schema-Version neu aufgenommenen, optionalen Element handelt es sich um eine abwärtskompatible Anpassung. Das Fehlen des neuen Elements in der auf Basis der alten Schema-Version erstellten XML-Datei, führt zu keinem Fehler in der Validierung. Aber in der Verarbeitung könnte dies trotzdem zu einer Fehlinterpretation führen.*

Echte Abwärtskompatibilität kann im Kontext von eCH mit einer unbekanntem Anzahl von Sendern und Empfängern nicht gewährleistet werden. In überschaubaren Communities (Menge der verarbeitenden Stellen ist abschliessend bekannt) kann Abwärtskompatibilität in Einzelfällen erreicht werden.

## 3.11 Erweiterbarkeit

Für die Erstellung erweiterbarer Schemas gelten folgende Regeln:

- **SHOULD:** Zur Erweiterung von XML-Schemas soll im zu erweiternden Schema der Mechanismus von `xs:any` verwendet werden.
- **SHOULD:** Für komplexe Erweiterungen soll ein entsprechendes XML-Schema bereitgestellt werden.
- **SHOULD:** Wird für eine Erweiterung ein XML-Schema bereitgestellt, so muss der entsprechend Namespace auf Stufe des Anwendungsfalls (siehe Kapitel 3.3) mittels Präfix den Querbezug zum erweiterten Namespace herstellen. Der Namespace einer Erweiterung muss den Querbezug zum erweiterten Namespace herstellen. Bsp. Wird der ursprüngliche Namespace <http://www.ech.ch/xmlns/eCH-0018/1> mittels eines XML-Schemas für die Strukturierung einer *Adresse* erweitert, so muss der Querbezug durch den Präfix eCH-0018 in der Kennung des Anwendungsfalls hergestellt werden. Bsp. [http://schema.meineFirma.ch/xmlns/eCH-0018\\_Adresse/1](http://schema.meineFirma.ch/xmlns/eCH-0018_Adresse/1)

## 4 XML-Instanzen

### 4.1.1 Zeichensätze (Character Encodings)

Das spezifische Character Encoding (also die Frage, wie ein Zeichen des Zeichensatzes als Bytes codiert wird), kann unterschiedlich sein.

- **SHOULD:** Als Codierung sollte UTF-8 verwendet werden, da diese Codierung zwingend (durch den XML Standard gefordert) von jeder XML-Software unterstützt werden muss.
- **SHOULD:** Die „encoding“ Deklaration in der XML-Deklaration soll angegeben werden.

#### Empfehlungen für eCH-Standards

- **MUST:** Die „encoding“ Deklaration in der XML-Deklaration muss immer angegeben werden.



## 4.2 Character References

Character References (<http://www.w3.org/TR/REC-xml/#NT-CharRef>) referenzieren ein durch seinen Code Point identifiziertes Zeichen aus dem Unicode-Zeichensatz. In der Praxis werden nur Zeichen als Character Reference angegeben, die sich mit dem für das XML Dokument verwendeten Character Encoding nicht direkt codieren lassen.

- **SHOULD:** Character References sollen soweit wie möglich vermieden werden. Dies ist am einfachsten und in den meisten Fällen möglich durch eine geeignete Wahl (UTF-8) des Character Encodings des XML Dokuments.
- **SHOULD NOT:** Für Zeichen, die im gewählten Character Encoding direkt verwendet werden können, sollen keine Character References verwendet werden.

## 4.3 Entity declarations and references

- **MUST NOT:** Entity Declarations dürfen nicht verwendet werden. Insbesondere externe Deklarationen stellen ein Sicherheitsrisiko dar (XML External Entity Injection (XXE)).
- **MUST NOT:** Entity References dürfen nicht verwendet werden, mit Ausnahme der vordefinierten `&amp;`, `&lt;`, `&gt;`, `&apos;`, `&quot;`;

## 4.4 Umgang mit Whitespace (Zeilenumbrüche etc.)

XML basiert auf einer zeichenbasierten Syntax, in der die Struktur von Daten durch Elemente und Attribute bestimmt wird. Aus Applikationssicht sind so genannte Whitespace Text Nodes fast nie relevant.

- **MUST:** Applikationen dürfen nicht davon ausgehen, dass der Whitespace in XML-Dokumenten in einer bestimmten Art formatiert ist. Falls aber Whitespace in Text Nodes signifikant ist, muss dies durch das `xml:whitespace="preserve"` Attribut gekennzeichnet werden.

## 4.5 Zugehörigkeit zu einem XML-Schema

Die Zugehörigkeit eines XML Dokuments zu einem Schema kann über zwei Mechanismen sichtbar gemacht werden: über den verwendeten Namespace oder dadurch, dass mittels des `xsi:schemaLocation`-Attributes auf das Schema-Dokument verwiesen wird.

- **SHOULD NOT:** XML Dokumente sollten nicht mittels des `xsi:schemaLocation`-Attributes auf das Schema verweisen. Ausschliesslich der Namespace-Name sollte die relevante Identifikation des Schemas sein.

# 5 Sicherheitsüberlegungen

**MUST:** Datenschutz- und sicherheitsrelevante Vorgaben sind beim Design von XML-Schemen zu berücksichtigen.

Insbesondere zu prüfen sind die Themen:



- Externe Referenzen, siehe Kapitel 4.3
- Nicht lokal gespeicherten Schemen, siehe Kapitel 3.3
- Defaultwerte, siehe Kapitel 3.4.1
- Binäre Daten, siehe Kapitel 3.5
- Constraints, siehe Kapitel 3.4.3
- Aktive Inhalte, siehe Kapitel 3.4.1

## 6 Haftungsausschluss/Hinweise auf Rechte Dritter

**eCH**-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellen oder welche **eCH** referenzieren, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

## 7 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende, sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

**eCH**-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

## Anhang A – Referenzen & Bibliographie

[ISO3166]	International Organization for Standardization, <i>Codes for the Representation of Names of Countries and their Subdivisions</i> , ISO 3166, November 2001.
[ISO639]	International Organization for Standardization, <i>Codes for the Representation of Names of Languages</i> , ISO 639, July 2002.
[ISO8601]	International Organization for Standardization, <i>Data Elements and Interchange Formats — Information Interchange — Representation of Dates and Times</i> , ISO 8601, December 2004. <a href="http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874">http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874</a>
[RFC2119]	Scott O. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , Internet RFC 2119, March 1997. <a href="ftp://ftp.isi.edu/in-notes/rfc2119.txt">ftp://ftp.isi.edu/in-notes/rfc2119.txt</a>
[RFC3066]	Harald Tveit Alvestrand, <i>Tags for the Identification of Languages</i> , Internet RFC 3066, January 2001. <a href="ftp://ftp.isi.edu/in-notes/rfc3066.txt">ftp://ftp.isi.edu/in-notes/rfc3066.txt</a>
[XMLSchema1]	Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, <i>XML-Schema Part 1: Structures Second Edition</i> , World Wide Web Consortium, Recommendation REC-xmlschema-1-20041028, Oktober 2004. <a href="http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/">http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/</a>
[xml11schema10]	Henry S. Thompson, <i>Processing XML 1.1 Documents with XML-Schema 1.0 Processors</i> , World Wide Web Consortium, Note NOTE-xml11schema10-20050511, May 2005. <a href="http://www.w3.org/TR/2005/NOTE-xml11schema10-20050511">http://www.w3.org/TR/2005/NOTE-xml11schema10-20050511</a>
[XMLSchema2]	Paul V. Biron, Ashok Malhotra, <i>XML-Schema Part 2: Datatypes Second Edition</i> , World Wide Web Consortium, Recommendation REC-xmlschema-2-20041028, Oktober 2004. <a href="http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/">http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/</a>
[XMLNS]	Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, Henry S. Thompson - Third Edition 8. December 2009 <a href="http://www.w3.org/TR/2009/REC-xml-names-20091208/">http://www.w3.org/TR/2009/REC-xml-names-20091208/</a>
[OWASP]	XML Security Cheat Sheet, The Open Web Application Security Project, <a href="https://www.owasp.org/index.php/XML_Security_Cheat_Sheet">https://www.owasp.org/index.php/XML_Security_Cheat_Sheet</a>

## Anhang B – Mitarbeit & Überprüfung

Eisenhut Claude	Eisenhut Informatik AG
Hugentobler Wolfgang	iDPARC AG
Maitre Giles	SBB AG
Müller Willy	Informatiksteuerungsorgan des Bundes
Stingelin Martin	Stingelin Informatik GmbH

## Anhang C – Abkürzungen und Glossar

Ein ausführliches Online-Glossar ist unter <http://dret.net/glossary/> zu finden.

ASCII	American Standard Code for Information Interchange	Der kleinste gemeinsame Nenner (fast) aller Zeichencodierungen (7-Bit Codierung), darauf aufbauende Codierungen erweitern ASCII z.B. um Umlaute (ISO 8859-1) oder ganz allgemein internationale Zeichen (Unicode).
Attribut		Attribute sind Elementen zugeordnet, sie enthalten zusätzliche Informationen zu einem Element.
Content Model		Bestimmt in einem Schema (DTD oder XML-Schema) den erlaubten Inhalt eines Element-Typs (definiert damit die erlaubte Verwendung des Element-Typs).
DOM	Document Object Model	
DSDL	Document Schema Definition Languages	
DTD	Document Type Definition	Die im XML Standard selber definierte Schema Sprache für XML Dokumente, hat insbesondere Schwächen im Bereich der Datentypen.
Element		Grundlegendstes Strukturierungsmerkmal eines XML Dokuments, Nur Elemente können verschachtelt werden und erlauben damit die komplexen Strukturen, die in XML oftmals verwendet werden.
Entity		Ein Stück XML Text, kann definiert und referenziert werden (es gibt verschiedene Typen, die wichtigsten sind General und Parameter Entities).
GRDDL	Gleaning Resource Descriptions from Dia-	

	lects of Languages	
HTML	Hypertext Markup Language	Dokumentenformat des WWW, basiert auf SGML und ist inkompatibel mit XML (aus diesem Grund gibt es mit XHTML eine XML-basierte Version von HTML).
IANA	Internet Assigned Numbers Authority	
IETF	Internet Engineering Task Force	
Information Set		Das Datenmodell von XML, modelliert XML als Information Items mit Properties.
IRI	Internationalized Resource Identifier	
ISO 8859		8-Bit Codierung von Zeichen, weit verbreitet, aber inkompatibel mit UTF-8.
Markup		Die physische Form eines XML Dokuments (Serialisierung des Information Set).
Namespaces		Methode zur Benennung und Verwendung von Namensräumen in XML.
Processing Instruction		Verarbeitungsanweisungen in XML-Dokumenten, nicht Teil des eigentlichen Inhalts, sondern eher Zusatzinformationen (Syntax: "<?name content?>").
RDBMS	Relational Database Management System	
RDDL	Resource Directory Description Language	
RDF	Resource Description Framework	
RFC	Request for Comments	
Schema		Beschreibung einer Klasse von XML-Dokumenten, die bekanntesten Dialekte sind DTD (Teil von XML selber) und XML-Schema (separater Standard).
Schema-Dokument		Dokument mit der Beschreibung der Datentypen und Elemente von Instanzen
Unicode		Standard für die Referenzierung und Codierung sehr vieler Zeichen.

URI/URL	Universal Resource Identifier/Locator	Standard für die Adressierung von Ressourcen auf dem Web. besteht aus einem Schema Identifier (z.B. "http:") und weiterer Information zur Adressierung.
UTF-16	Unicode Transformation Format 16	Eine Zeichencodierung, die jedes Unicode Zeichen als 16 oder 32 Bit codiert (alle geläufigen Zeichen werden als 16 Bit codiert). Jede XML Software muss UTF-16 unterstützen.
UTF-8	Unicode Transformation Format 8	Die XML Default-Codierung von Unicode Zeichen, jedes ASCII Dokument ist auch ein UTF-8 Dokument (UTF-8 codiert Zeichen als 1-6 Bytes). Jede XML Software muss UTF-8 unterstützen.
W3C	World Wide Web Consortium	
well-formed		Ein XML Dokument ist well-formed, wenn es den XML Syntax-Regeln gehorcht.
XHTML	Extensible Hypertext Markup Language	
XInclude	XML Include	Regelt die Einbindung von externen XML-Ressourcen in XML Dokumente.
XLink	Extensible Linking Language	
XML	Extensible Markup Language	Sprache für die Repräsentation von baumstrukturierten Dokumenten und ihr Schema.
XML-Schema		Schemasprache für XML, die deutlich leistungsfähiger ist als die in XML DTDs definierten Mechanismen.
XSD	XML-Schema Definition Language,	häufig verwendete (aber nicht offizielle) Abkürzung für XML-Schema
XSLT	XSL Transformations	Teil von XSL, auf XML-Transformation spezialisierte Programmiersprache die es ermöglicht, ein XML Dokument in eine andere Repräsentation (XML, HTML, ...) zu transformieren.

## Anhang D – Änderungen gegenüber Vorversion

- Der Standard wurde grundlegend überarbeitet.
- Das Dokument wurde neu strukturiert.
- Die folgenden Standards wurden aufgelöst und deren Empfehlungen in diesen Standard übernommen.
  - eCH-0033: Beschreibung von XML-Namespaces

- eCH-0035: Design von XML-Schemas
- eCH-0050: XML\_Hilfskomponenten
- eCH-0062: Design von XML-Schemas – Zusammenfassung

## Anhang E – Abbildungsverzeichnis

Abbildung 1: Übersicht der wichtigsten Objekte ..... 5