

eCH-0018 XML Best Practices

Nom	XML Best Practices
eCH- nombre	eCH-0018
Catégorie	Norme
Stade	Défini
Version	2.0
Statut	Approuvé
Date de décision	2018-09-05
Date de publication	2018-11-09
Remplace version	1.0 < Major Change >
Condition préalable	-
Annexes	-
Langues	Allemand (original), français (traduction)
Auteurs	Groupe spécialisé XML Willy Müller, Unité de pilotage informatique de la Confédération (UPIC), Willy.Mueller@UPIC.admin.ch
Éditeur / Distribution	Association eCH, Mainaustrasse 30, case postale, 8034 Zurich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Condensé

Le présent document décrit les règles et recommandations à prendre en compte lors de la conception de schémas XML et d'espaces nominatifs ou Namespaces et de leur utilisation dans le cadre des normes eCH.

Table des matières

Condensé	1
Table des matières	2
Remarque	3
1 Introduction	4
1.1 Statut	4
1.2 Champ d'application	4
1.3 Terminologie des recommandations	4
1.4 Références normatives	5
2 Principes généraux	5
2.1 Modèle	5
2.2 Documents XML comme instances de classes	6
3 Conception de schémas XML	6
3.1 Structuration de schémas et documents de schéma	7
3.2 Conventions générales de nommage	7
3.3 XML Namespace	7
3.4 Représentation des structures de données en XML	9
3.4.1 Eléments et attributs	9
3.4.2 Représentation de valeurs vides	10
3.4.3 Gestion de l'identification et des références	10
3.5 Données binaires	10
3.6 Documentation	11
3.7 Plurilinguisme	11
3.7.1 La langue des noms	11
3.7.2 Marquage linguistique des contenus	12
3.7.3 Valeurs d'énumération (XSD-Enumeration).....	12
3.8 Gestion des listes de valeurs	12
3.9 Gestion de la substitution	13
3.9.1 Element et Type Substitution.....	13
3.9.2 Substitution Groups.....	13
3.10 Gestion des versions	14
3.11 Extensibilité	15
4 Instances XML	15
4.1.1 Jeux de caractères (Character Encodings).....	15
4.2 Character References	15
4.3 Entity declarations and references	16
4.4 Gestion du Whitespace (retours à la ligne etc.)	16
4.5 Appartenance à un schéma XML	16
5 Sécurité	16

6	Exclusion de responsabilité/droits de tiers.....	17
7	Droits d'auteur	17
	Annexe A – Références & bibliographie.....	18
	Annexe B – Collaboration & vérification	19
	Annexe C – Abréviations & glossaire.....	19
	Annexe D – Modifications par rapport à la version précédente	22
	Annexe E – Liste des illustrations	22

Remarque

En vue d'une meilleure lisibilité et compréhension, seul le genre masculin est utilisé pour la désignation des personnes dans le présent document. Cette formulation s'applique également aux femmes dans leurs fonctions respectives.

1 Introduction

1.1 Statut

Approuvé: Le document a été approuvé par le comité d'experts. Il a pouvoir normatif pour le domaine d'utilisation défini dans le domaine de validité donné.

1.2 Champ d'application

La présente norme définit les directives à appliquer pour l'utilisation et l'élaboration de documents XML et de schémas XML [XMLSchema1], [XMLSchema2] dans le cadre d'échanges de données structurées. Il définit par ailleurs les directives applicables à la description des Namespaces XML [XMLNS]. Outre les directives générales, il définit aussi les règles spécifiquement applicables à l'élaboration de schémas XML pour les normes eCH.

Ces directives ne s'appliquent pas aux schémas et instances XML servant à structurer les textes (docbook, xhtml, ou semblables).

1.3 Terminologie des recommandations

Les directives de ce document respectent la terminologie de [RFC2119]. Ce faisant, les expressions suivantes, apparaissant en LETTRES MAJUSCULES ont les significations suivantes: (citation tirée de RFC 2119):

- **MUST/DOIT:** ce mot ou les termes "**REQUIRED/OBLIGATOIRE**" ou "**SHALL/DEVRA**" veut dire que la définition est une exigence absolue de la spécification.
- **MUST NOT/NE DOIT PAS:** cette phrase ou l'expression "**SHALL NOT/NE DEVRA PAS**" veut dire que cette définition est une interdiction absolue de la spécification.
- **SHOULD/DEVRAIT:** ce mot ou l'adjectif "**RECOMMENDED/RECOMMANDÉ**" veut dire qu'il peut exister des raisons valables dans des circonstances particulières pour ignorer un item particulier, mais les répercussions doivent être comprises et soigneusement évaluées avant d'opter pour un changement de cap.
- **SHOULD NOT/NE DEVRAIT PAS:** cette phrase ou la phrase "**NOT RECOMMENDED/NON RECOMMANDÉ**" veut dire qu'il peut exister des raisons valables dans des circonstances particulières pour lesquelles le comportement particulier est acceptable voir utile, mais les répercussions doivent être comprises et le cas être soigneusement évalué avant de mettre en œuvre tout comportement décrit par cette étiquette.
- **MAY/PEUT:** ce mot ou l'adjectif "**OPTIONAL/FACULTATIF**" veut dire qu'un item est vraiment facultatif ou optionnel. Un vendeur peut choisir d'inclure l'item parce qu'une place de marché particulière l'exige ou parce que le vendeur pressent que cela améliore le produit alors qu'un autre vendeur peut omettre le même item. Une implémentation qui n'inclut pas d'option particulière DOIT être préparée à interopérer avec une autre implémentation qui inclut l'option, bien que la fonctionnalité puisse s'en trouver réduite. De la même manière, une implémentation qui inclut une option particulière

DOIT être préparée à interopérer avec une autre implémentation, qui n'inclut pas l'option (exception faite, bien entendu, pour la fonction fournie par l'option)

L'élaboration de normes eCH et de leurs schémas XML est parfois soumise à des règles plus astreignantes. Sauf mention spécifique d'eCH dans le contexte des recommandations, ces dernières s'appliquent également à la rédaction des normes eCH et de leurs schémas XML. Si d'autres règles plus strictes sont en vigueur pour eCH, elles sont évoquées à la rubrique «**Recommandations pour les normes eCH**».

1.4 Références normatives

Les règles suivantes doivent être considérées comme normatives du point de vue du document eCH-0018.

Voir l'annexe A:

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

2 Principes généraux

2.1 Modèle

Le graphique suivant offre une vue d'ensemble approximative des objets évoqués dans la norme eCH-0018.

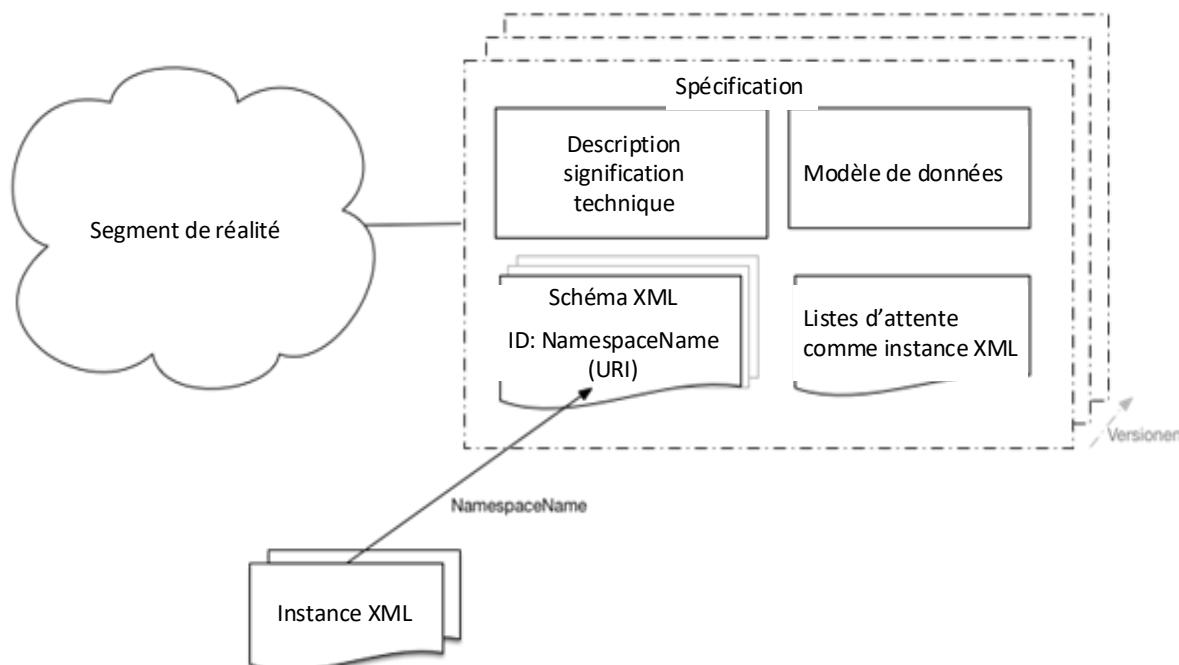


Figure 1: Vue d'ensemble des principaux objets

La description du segment de réalité présentant un intérêt pour l'échange de données ou une révision requiert une spécification décrivant la sémantique des éléments considérés. Une telle spécification se compose, en règle générale, de la combinaison d'une description

technique (le document de norme dans le cas d'eCH), d'un modèle de données avec des classes/entités et leurs relations (également partie intégrante du document de norme dans le cas d'eCH), d'un ou de plusieurs schémas XML et, le cas échéant, de listes de valeurs spécifiques en tant qu'instances XML. Lorsque l'on parle de documents XML, il s'agit aussi bien de schémas XML que de documents d'instance XML. La relation entre un document d'instance XML et le schéma XML correspondant est établie via le nom de Namespace. Le schéma XML est également identifié au moyen du nom de Namespace établi comme URI. Toute adaptation à l'une des parties intégrantes de la spécification donne lieu à une nouvelle version de la spécification.

2.2 Documents XML comme instances de classes

Les documents XML sont des documents de texte bien constitués selon la norme XML. Dans le contexte qui nous intéresse, on distingue deux types de documents XML:

Schéma XML Document XML qui respecte les règles de la norme [XMLSchema] et décrit ainsi les instances XML.

Instance XML Document XML qui respecte les règles d'un schéma XML particulier.

De manière générale:

- **SHOULD:** le schéma devrait être défini – en vue du traitement informatisé de données typées (ce qui signifie presque toutes les données dans des scénarios Business-to-Business) – au moyen d'un schéma XML, car il n'y a que dans ce dernier que les types de données peuvent être définis sous une forme proche de l'application.
- **SHOULD:** les documents XML devraient être décrits par au moins un schéma XML afin de permettre aux utilisateurs de comprendre les règles selon lesquelles sont élaborés les documents d'instance correspondants. Il existe par ailleurs des outils capables de contrôler automatiquement si un document d'instance correspond bien au schéma XML qui lui est affecté.

Recommandations pour les normes eCH

- **MUST:** les documents XML doivent être décrits par un schéma XML.

3 Conception de schémas XML

Ce chapitre traite de la structure interne des schémas XML, c'est-à-dire la construction et la relation de ce que l'on appelle les «composants de schéma XML».

- **MUST:** à chaque schéma XML correspond une documentation décrivant la sémantique technique des composants du schéma et le contexte d'application prévu.
- **SHOULD:** `elementFormDefault="qualified"` devrait être utilisé.
- **SHOULD NOT:** aucune valeur par défaut ne devrait être définie pour les éléments et les attributs.

3.1 Structuration de schémas et documents de schéma

Les points suivants s'appliquent à la structuration de documents de schéma:

- **SHOULD:** un schéma devrait être implémenté dans un document de schéma.
- **SHOULD:** le mécanisme *xs:import* devrait être employé en cas de réutilisation de définitions existantes.
- **SHOULD NOT:** *xs:include* ne permet pas de séparer clairement les composants réutilisables au moyen du Namespace et ne devrait donc pas être utilisé.
- **SHOULD NOT:** le mécanisme *xs:redefine* du schéma XML devrait être évité, car il aboutit à des dépendances complexes entre les différents documents de schéma et pourrait entraîner des erreurs difficiles à repérer notamment en cas de modifications ultérieures de documents de schéma.

3.2 Conventions générales de nommage

La normalisation de la structure des noms utilisés (et non seulement de leur langue) est extrêmement importante pour améliorer la lisibilité des documents XML et créer une base de validité générale pour tous les développeurs XML.

- **MUST:** les noms se composent exclusivement de majuscules et minuscules ASCII (a-z et A-Z), chiffres 0-9, underscore (_), point (.), et tiret (-).
- **SHOULD:** si les noms des particules de schémas se composent de plusieurs mots, ils s'écrivent en notation Camel Case (en commençant par une minuscule).
Exemple: `<recordNumber>12345678</recordNumber>`
- **SHOULD:** les abréviations ne devraient être utilisées que lorsqu'il est question d'abréviations courantes et connues dans les domaines de spécialisation
- **SHOULD:** concernant les abréviations, le mode d'écriture établi devrait être repris tel quel.
- **MUST:** les noms de Simple et Complex Types se terminent par "Type".
- **MUST:** les noms de Attribute Groups se terminent par "AttributeGroup".
- **MUST:** les noms de Named Model Groups se terminent par "Group".

Recommandations pour les normes eCH

- **SHOULD:** les abréviations devraient faire l'objet d'explications dans la norme eCH et être par ailleurs décrites dans le schéma XML au moyen d'annotations.

3.3 XML Namespace

Le XML Namespace est un mécanisme destiné à garantir le référencement sans ambiguïté des éléments non identiques mais portant le même nom. Combiner le nom de Namespace et le nom utilisé au niveau local (local name) élimine toute ambiguïté concernant le nom local. Les noms de Namespace sont des URI, qui ne doivent toutefois pas forcément faire référence à une ressource existante.

La déclaration des Namespaces XML dans une instance XML s'effectue au moyen d'attributs *xmlns* de n'importe quel élément dans les documents XML. Soit un Namespace se voit ainsi attribuer un préfixe (*xmlns:prefix="uri"*, *prefix* pouvant être n'importe quel nom XML sans

double-points et *uri* étant le nom de Namespace), soit il s'agit du Default Namespace (espace nominatif par défaut) (*xmlns="uri"*, *uri* étant le nom de Namespace).

Les préfixes Namespace sont utilisés afin d'établir la relation entre un nom qualifié (*<html:title>*) et la déclaration de Namespace (*xmlns:html="http://www.w3.org/1999/xhtml"*) dans un document XML.

ATTENTION: le Default Namespace (déclaré par *xmlns="..."*) ne s'applique pas aux attributs, ce qui signifie les attributs sans préfixe ne sont jamais alloués à un Namespace.

Dans un schéma XML, le XML Namespace est défini par l'attribut *target-namespace* pour les instances XML décrites par le schéma.

- **MUST:** chaque schéma XML doit définir son propre Namespace XML. L'attribut *targetNamespace* de l'élément *xs:schema* doit être employé pour faire référence à un Namespace par un schéma.
- **SHOULD:** le nom de Namespace (=des URI) d'un schéma XML ne devrait pas orienter directement vers le schéma, mais, en tant que http-URL, vers une description du Namespace.
- **MUST:** le service définissant le schéma doit être propriétaire du nom de domaine utilisé pour le nom de Namespace.
- **SHOULD:** un sous-domaine spécifique au schéma devrait être utilisé pour le nom de Namespace d'un schéma (voir exemple ci-dessous).
- **MUST:** le schéma XML doit pouvoir être repéré par les utilisateurs dans le chemin de la http-URL sous la Namespace-URL.
- **MUST:** la validation du schéma doit être réalisée avec les schémas XML au niveau local. La disponibilité du serveur avec la Namespace-URL n'est pas garantie.
- **SHOULD:** un nom de Namespace devrait être composé des éléments suivants:

http	://	www.ech.ch	/	xmlns	/	eCh-0018	/	1
URI Sche me		Domain Name		Nom général pour les des- criptions de Namespace		Cas d'application		Ver- sion

Les schémas XML devraient être placés dans un sous-domaine spécifique et pas sous *www...* afin d'éviter les conflits éventuels avec un CMS (Content Management System).

Exemple: *schemas.geo.admin.ch* au lieu de *www.geo.admin.ch/xmlns*

Le mécanisme XML Catalog (OASIS) peut être utilisé pour permettre la validation sur la base des Namespace-URL (sans chemin local) pour les schémas XML au niveau local également.

Recommandations pour les normes eCH

- **MUST:** le nom de Namespace pour les normes eCH doit commencer par <http://www.ech.ch/xmlns>, suivi du numéro de la norme correspondante.
Exemple: <https://www.ech.ch/xmlsn/eCH-0018>
- **SHOULD:** la version du Namespace doit pouvoir être identifiée par le nom du Namespace.

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

3.4 Représentation des structures de données en XML

Les chapitres suivants décrivent les différents aspects, qui importent pour la représentation des structures de données en XML, et fournissent des recommandations d'amélioration et des informations concernant les mécanismes moins appropriés afin de mieux traiter de tels problèmes.

- **SHOULD:** la représentation du modèle d'échange de données (entités) et celle de l'interface (événements à échanger respectivement format d'interface) devraient être effectuées dans des schémas XML séparés.

Recommandations pour les normes eCH

- **MUST:** les normes concernant les données et les normes d'interface doivent être séparées.
- **SHOULD:** les identificateurs, entités et interfaces devraient être définis séparément.
- **SHOULD NOT:** les schémas de base non spécifiques (au sens de Commons) devraient être abandonnés au profit de schémas de base dédiés avec une cohésion claire.

3.4.1 Eléments et attributs

Bien que l'utilisation d'attributs entraîne des instances de document XML techniquement plus petites, on privilégiera le recours aux éléments. Ce dernier permet en effet de représenter des structures plus simples et surtout reconnaissables.

L'utilisation d'attributs n'est pas sans inconvénient:

- l'ordre des attributs ne peut être défini,
- il n'y a pas de possibilité de structuration,
- une modification de type à venir entraîne, en cas d'utilisation d'attributs dans les instances, à un changement de structure fondamental,

ce qui peut poser problème lors du traitement des instances XML correspondantes.

De ce point de vue-là, il n'y a aucune raison – concernant le schéma XML – d'utiliser encore des attributs, hormis lorsque des normes existantes (Namespaces XML ou XLink par exemple) ou d'autres accords le prescrivent.

- **SHOULD:** en règle générale, la structuration devrait se faire au moyen d'éléments, sont plus faciles à élargir en cas de besoin, par exemple au moyen d'attributs supplémentaires, ou en les subdivisant en structures articulées de manière plus détaillée.
- **MUST NOT:** les aspects techniques ne doivent pas être représentés au moyen d'attributs. L'ordre des attributs ne peut être spécifié. S'ils doivent être structurés ultérieurement, le traitement des instances XML correspondantes devra faire l'objet d'une révision en profondeur.
- **MAY:** les attributs devraient être utilisés, tout au plus, pour les aspects des métadonnées ou les aspects techniques des documents XML.
- **MUST NOT:** les valeurs par défaut des attributs ne doivent pas être définies dans le schéma, parce que le *parser*, ne connaissant pas le schéma de l'application, fournit

les mauvaises données.

- **SHOULD:** pour des raisons de sécurité, Max-Occurs ne doit pas être défini *unbound* et aucune structure récursive ne doit être utilisée.
- **SHOULD:** pour des raisons de sécurité, les longueurs maximales devraient être définies pour les éléments de texte.
- **SHOULD NOT:** les contenus actifs ne devraient pas être échangés.
- **SHOULD:** Des définitions de schéma adaptées devraient permettre d'empêcher que des éléments et attributs vides puissent être codés (minLength ou token par exemple)

3.4.2 Représentation de valeurs vides

Dans le contexte d'une application, il est souvent exigé que certaines indications soient facultatives, c'est-à-dire qu'elles ne soient pas impérativement indiquées dans les instances. Le schéma XML offre la possibilité de représenter cela au moyen du mécanisme „xsi:nil“ ou d'éléments facultatifs.

- **SHOULD:** les valeurs nulles devraient être représentées en ayant recours à des éléments facultatifs, c'est-à-dire en ne faisant pas apparaître les composants concernés,.
- **SHOULD NOT:** le mécanisme *xsi:nil* devrait être évité.
- **SHOULD:** si l'absence d'une valeur a une importance concrète, cela devrait être représenté par une valeur explicite ("inconnu par exemple").

3.4.3 Gestion de l'identification et des références

On trouve fréquemment dans les modèles de données des relations entre les différentes entités/classe. Une façon de relever techniquement un tel défi consiste à attribuer des identificateurs/noms, qui permettent de nommer les entités/objets à identifier et de les utiliser dans les références.

- **SHOULD NOT:** les références ne devraient pas être utilisées.
- **SHOULD NOT:** *QName* ne devrait pas être utilisé comme référence.
- **SHOULD:** les conditions garantissant l'absence d'ambiguïté (unique) devraient être documentées.
- **SHOULD NOT:** concernant la sécurisation d'Identity Constraints (IDREF et keyref), on devrait avoir recours à des mécanismes non pas XML, mais applicatifs.

3.5 Données binaires

XML se concentre principalement sur les données pouvant être représentées par du texte. Concernant les données binaires, le schéma XML définit les types de données *xs:hexBinary* (chiffres hexadécimaux comme caractères) et *xs:base64Binary* (données binaires représentées par des lettres selon [RFC2045]). La représentation *xs:base64Binary* se révèle plus compacte que la représentation *xs:hexBinary*.

- **SHOULD:** des données utiles plus complètes devraient être échangées comme BASE64 et modélisées en conséquence dans le schéma comme base64Binary.

Dans l'éventualité où les données binaires, en raison de leur taille par exemple, posent problème pour le transport ou le traitement, il faut privilégier une solution alternative, transparente du point de vue du schéma XML.

- exemple concernant le transport – répartition en paquets
- exemple concernant le traitement – remplacer les données par des références internes.

3.6 Documentation

Un schéma XML définit la structure syntaxique et restreint le but et la signification des documents XML. Par conséquent, le but et la signification de la syntaxe devrait en outre être décrits de manière explicite. Ceci peut se faire par une documentation dans le schéma même ou par de la documentation d'accompagnement.

- **SHOULD:** si la documentation a lieu directement dans le schéma, elle devrait être réalisée de manière concise et marquante dans `xs:documentation`.
- **SHOULD:** si la documentation a lieu directement dans le schéma, elle devrait être réalisée dans la même langue que les éléments / noms d'éléments.
- **MUST NOT:** la documentation dans les schémas XML ne doit pas apparaître dans les commentaires XML (`<!-- ... -->`).

Recommandations pour les normes eCH

- **MUST:** le but et la signification de tous les éléments doivent être décrits dans le document de norme.

3.7 Plurilinguisme

3.7.1 La langue des noms

Lorsqu'ils sont développés à l'aide de noms localisés, les schémas risquent de n'être acceptés que dans la région linguistique concernée. Une solution applicable dans le monde entier doit être recherchée pour tenir compte du plurilinguisme de la Suisse, mais aussi des exigences internationales. Lorsque plusieurs schémas sont combinés entre eux, une telle solution augmente en outre la cohérence, qui aurait fortement à souffrir de l'utilisation de noms en différentes langues dans le même document XML.

- **SHOULD:** la langue des noms est l'anglais.
- **MAY:** si le schéma utilise des noms qui ont une signification spécifique à la langue correspondante et ne peuvent être traduits (par ex. certains termes juridiques), ceux-ci peuvent être écrits dans la langue pour laquelle leur sens doit être conservé.

Recommandations pour les normes eCH

- **MUST:** la langue des noms est l'anglais, à la condition qu'il existe une traduction anglaise du terme concerné.

3.7.2 Marquage linguistique des contenus

Les marqueurs linguistiques permettent de désigner des contenus d'éléments sans ambiguïté dans la langue des contenus. Il est ainsi possible de représenter des contenus pluri-lingues sans ambiguïté et de manière normée (codes des pays ISO 3166, codes des langues ISO 639).

- **SHOULD:** les renseignements linguistiques devraient être appliqués en tant qu'éléments (Value-Pair), un élément contenant l'identification de la langue composée de la langue et du pays («fr-CH» par exemple) et l'autre le texte à proprement parler dans la langue en question.

```
<voteDescriptionInfo>  
  <language>de-CH</language>  
  <voteDescription>Votation du 10.06.2018</voteDescription>  
</voteDescriptionInfo>
```
- **SHOULD NOT:** les codages linguistiques au moyen de l'attribut *xml:lang* sont à éviter.
- **MUST:** les codages linguistiques doivent être représentés selon le RFC 3066 [RFC3066]. Les applications doivent utiliser les codages ISO 639 [ISO639] à deux caractères (de, fr, it, rm, en) ou les codages combinés ISO 639/3166 [ISO3166] (fr-CH, en-US).

3.7.3 Valeurs d'énumération (XSD-Enumeration)

Les points applicables concernant les valeurs d'énumération prescrites par le schéma sont les suivants:

- **SHOULD:** les valeurs d'énumération (enumeration) devraient utiliser des noms parlants.
- **SHOULD:** la langue des valeurs d'énumération est l'anglais (cohérence avec la langue des éléments)
- **MAY:** si le schéma utilise des noms qui ont une signification spécifique à la langue correspondante et ne peuvent être traduits (par ex. certains termes juridiques), ceux-ci peuvent être écrits dans la langue pour laquelle leur sens doit être conservé.

3.8 Gestion des listes de valeurs

Les points suivants s'appliquent aux listes de valeurs ne pouvant être représentées sous forme de XSD-Enumeration:

- **MUST:** Il faut décider si les listes de valeurs devraient être ou non prises en compte dans la gestion des versions avec le schéma XML.
- **SHOULD NOT:** des valeurs de liste changeant fréquemment ne devraient pas apparaître dans le schéma XML.
- **MUST:** quand la liste de valeurs fait partie du schéma XML, toute modification doit donner lieu à une nouvelle version du nom de Namespaces.
- **SHOULD NOT:** XSD-QName ne devrait pas être utilisé pour le type de valeur.

3.9 Gestion de la substitution

3.9.1 Element et Type Substitution

Lorsque l'on utilise l'héritage pour représenter la hiérarchisation de termes techniques, il est possible d'employer les constructions de schémas XML suivants:

- Element Substitution
- Type Substitution

Le recours à l'héritage reste peu courant dans l'environnement XML. On le retrouve dans des cas particuliers axés sur le paradigme de programmation orientée objet. C'est notamment le cas lorsque l'on utilise des Frameworks techniques de programmation, comme la sérialisation XML de JAVA ou .NET

- **SHOULD NOT:** les mécanismes d'héritage devraient être proscrits autant que possible. S'ils se révèlent malgré tout nécessaires, ils doivent être cantonnés au Type Substitution.
- **MUST:** si la documentation est explicite dans le schéma ou même que la conception du schéma impose (types définis comme *abstract="true"*) que *xsi:type* peut ou devrait être utilisé, ceci est autorisé mais doit faire l'objet d'une documentation à la hauteur. Lors du traitement de ces instances, il faut faire attention à ce que cela soit fait sur la base du type afin que l'affectation de type s'effectue non pas via le type d'un élément dans le schéma, mais par le biais du *xsi:type* provenant de l'instance.
- **MUST:** l'utilisation de mécanismes de Type Substitution doit faire l'objet d'une documentation claire et à la hauteur afin que les utilisateurs du schéma soient clairement informés à propos de cet aspect du schéma.
- **MUST:** si l'un des deux mécanismes est utilisé (*xsi:type* ou *substitutionGroup*), il est le seul dont l'utilisation est autorisée. Un mélange de deux styles rendrait un schéma difficile à comprendre et en compliquerait ainsi inutilement la gestion.

3.9.2 Substitution Groups

Dans un schéma XML, un élément peut indiquer de manière explicite, au moyen l'attribut *substitutionGroup*, qu'il se substitue dans cette instance à cet élément référencé. Il peut alors survenir dans une instance partout où l'élément référencé dans l'attribut *substitutionGroup* est autorisé. Le mécanisme peut également être défini à plusieurs niveaux.

- **SHOULD NOT:** en règle générale, les mécanismes de Substitution Groups sont à éviter tant qu'aucun motif important ne le préconise. Ces mécanismes compliquent le traitement et la compréhension du schéma et des instances définies par le schéma.
- **MUST:** l'utilisation de mécanismes de Substitution Groups doit faire l'objet d'une documentation claire et à la hauteur afin que les utilisateurs du schéma soient clairement informés à propos de cet aspect du schéma.

Ne pas utiliser de Substitution Groups a pour avantage:

- de simplifier les instances dans le document d'instance – dans la mesure où aucun héritage n'est nécessaire.

3.10 Gestion des versions

Les applications XML peuvent afficher une grande longévité et être sujettes à des modifications imposant à différentes versions des schémas de base de coexister. Pour pouvoir être identifiable, les versions d'un schéma doivent être gérées. Cette gestion des versions des schémas XML est régie par les règles suivantes:

- **MUST:** la version du Namespace et la version du schéma doivent concorder (1:1). Cela signifie que le nom de Namespace identifie le schéma sans ambiguïté.
- **MUST:** les changements apportés à la syntaxe et les adaptations d'ordre rédactionnel, qui modifient la sémantique, donnent lieu à une nouvelle version et par conséquent à un nouveau nom de Namespace.
Exemple: (../eCH-0018/1) devient (../eCH-0018/2)
- **MUST NOT:** les adaptations purement rédactionnelles sans incidence sur la sémantique (faute de frappe rectifiée dans un commentaire par exemple) ne donnent pas lieu à une nouvelle version (il existe d'autres moyens d'en faire part, comme une nouvelle date de publication par exemple). La version devrait être mentionnée dans l'instance afin que les applications soient en mesure d'initier le bon traitement.
- **MUST:** les documents XML doivent établir le lien avec la version de schéma exclusivement via nom de Namespace.
- **SHOULD NOT:** les documents XML ne devraient établir aucune relation avec la date de publication du schéma.

Le renseignement sur la version au sens de la norme eCH-0018 ne contient aucune information relative à une éventuelle compatibilité descendante. Le point suivant doit toutefois être pris en compte concernant la compatibilité descendante:

La condition suivante doit être remplie pour pouvoir parler de compatibilité descendante. La compatibilité descendante est toujours à sens unique (expéditeur ayant une version ancienne à un destinataire ayant une version plus récente).

Une nouvelle version d'un schéma XML, présentant une compatibilité descendante, doit pouvoir valider sans erreur **et** traiter correctement un fichier XML créé avec une version de schéma XML antérieure.

Ex.: quand un élément facultatif est intégré à une nouvelle version de schéma, on parle d'adaptation ayant une compatibilité descendante. L'absence du nouvel élément, dans le fichier XML créé à partir de l'ancienne version du schéma, n'entraîne pas d'erreur lors de la validation. De mauvaises interprétations restent malgré tout possibles lors du traitement.

Dans le contexte eCH, il n'est pas possible de garantir une véritable compatibilité descendante avec un nombre d'expéditeurs et de destinataires inconnus. Dans les Communities de dimensions raisonnables (on connaît précisément le nombre de services intervenant dans le traitement), une compatibilité descendante est tout à fait envisageable dans les cas particuliers.

3.11 Extensibilité

Les règles suivantes s'appliquent à la création de schémas extensibles:

- **SHOULD:** le mécanisme de `xs:any` devrait être utilisé dans le schéma afin de réaliser son extension.
- **SHOULD:** un schéma XML correspondant devrait être préparé pour les extensions plus complexes.
- **SHOULD:** si un schéma XML est préparé pour une extension, le Namespace doit en conséquence établir – au niveau du cas d'application (voir chapitre 3.3) le lien transversal pour le Namespace étendu, au moyen d'un préfixe.

Le Namespace d'une extension doit établir le lien transversal avec le Namespace étendu. Exemple: si le Namespace d'origine <http://www.ech.ch/xmlns/eCH-0018/1> fait l'objet d'une extension au moyen d'un schéma XML pour la structuration d'une *Adresse*, le lien transversal doit être établi au moyen du préfixe eCH-0018 dans l'identification du cas d'application. Exemple: http://schema.meine-Firma.ch/xmlns/eCH-0018_Adresse/1

4 Instances XML

4.1.1 Jeux de caractères (Character Encodings)

Le Character Encoding (codage de caractères) spécifique (la question de savoir comment un caractère d'un jeu est encodé en Bytes) peut en revanche être différent.

- **SHOULD:** le code UTF-8 devrait être utilisé car ce codage doit impérativement pouvoir être pris en charge par chaque logiciel XML (comme l'exige la norme XML).
- **SHOULD:** la déclaration d'«encoding» dans la déclaration XML devrait être indiquée.

Recommandations pour les normes eCH

- **MUST:** la déclaration d'«encoding» doit toujours être indiquée dans la déclaration XML.

4.2 Character References

Les Character References (<http://www.w3.org/TR/REC-xml/#NT-CharRef>) réfèrent un caractère identifié par son Code Point et provenant du jeu de caractères Unicode. En pratique, seuls les caractères, qui ne peuvent être directement encodés avec le Character Encoding utilisé pour le document XML, sont indiqués comme Character Reference.

- **SHOULD:** les Character References devraient être évitées autant que faire se peut. Il s'agit de la façon la plus simple, réalisable dans la plupart des cas en choisissant bien (UTF-8) le Character Encoding du document XML.
- **SHOULD NOT:** les Character References ne devraient pas être utilisées pour les caractères pouvant être utilisés directement dans le Character Encoding choisi.

4.3 Entity declarations and references

- **MUST NOT:** l'utilisation d'Entity Declarations est proscrite. En particulier, les déclarations externes présentent un risque pour la sécurité (XML External Entity Injection (XXE)).
- **MUST NOT:** l'utilisation d'Entity References est proscrite, à l'exception des *&*, *<*, *>*, *'*, *"*; prédéfinis.

4.4 Gestion du Whitespace (retours à la ligne etc.)

XML repose sur une syntaxe basée sur des caractères (Markup), dans laquelle des éléments et attributs déterminent la structure des données. Du point de vue des applications, ce qu'on appelle les Whitespace Text Nodes (nœuds de texte ne se composant que d'espaces blancs) ne revêtent quasiment pas la moindre importance.

- **MUST:** les applications ne doivent pas supposer que le Whitespace est formaté d'une manière déterminée dans les documents XML. **Si toutefois le Whitespace** dans les Text Nodes est significatif, ceci doit être marqué par l'attribut *xml:whitespace="preserve"*.

4.5 Appartenance à un schéma XML

Deux mécanismes permettent de rendre visible l'appartenance d'un document XML à un schéma: le Namespace utilisé d'une part, la référence faite par l'attribut *xsi:schemaLocation* au document de schéma d'autre part.

- **SHOULD NOT:** les documents XML ne devraient pas faire référence au schéma au moyen de l'attribut *xsi:schemaLocation*. Seul le nom du Namespace devrait permettre l'identification pertinente du schéma.

5 Sécurité

MUST: Les règles pertinentes en matière de protection des données et de sécurité doivent être prises en compte lors de la conception de schémas XML.

Les thèmes suivants doivent être vérifiés en particulier:

- Références externes, voir chapitre 4.3
- Schémas enregistrés hors du niveau local, voir chapitre 3.3
- Valeurs par défaut, voir chapitre 3.4.1
- Données binaires, voir chapitre **Fehler! Verweisquelle konnte nicht gefunden werden.**
- Constraints, voir chapitre 3.4.3
- Contenus actifs, voir chapitre 3.4.1

6 Exclusion de responsabilité/droits de tiers

Les normes élaborées par l'Association **eCH** et mises gratuitement à la disposition des utilisateurs, ainsi que les normes de tiers adoptées, ont seulement valeur de recommandations. L'Association **eCH** ne peut en aucun cas être tenue pour responsable des décisions ou mesures prises par un utilisateur sur la base des documents qu'elle met à disposition. L'utilisateur est tenu d'étudier attentivement les documents avant de les mettre en application et au besoin de procéder aux consultations appropriées. Les normes **eCH** ne remplacent en aucun cas les consultations techniques, organisationnelles ou juridiques appropriées dans un cas concret.

Les documents, méthodes, normes, procédés ou produits référencés dans les normes **eCH** peuvent le cas échéant être protégés par des dispositions légales sur les marques, les droits d'auteur ou les brevets. L'obtention des autorisations nécessaires auprès des personnes ou organisations détentrices des droits relève de la seule responsabilité de l'utilisateur.

Bien que l'Association **eCH** mette tout en œuvre pour assurer la qualité des normes qu'elle publie, elle ne peut fournir aucune assurance ou garantie quant à l'absence d'erreur, l'actualité, l'exhaustivité et l'exactitude des documents et informations mis à disposition. La teneur des normes **eCH** peut être modifiée à tout moment sans préavis.

Toute responsabilité relative à des dommages que l'utilisateur pourrait subir par suite de l'utilisation des normes **eCH** est exclue dans les limites des réglementations applicables.

7 Droits d'auteur

Tout auteur de normes **eCH** en conserve la propriété intellectuelle. Il s'engage toutefois à mettre gratuitement, et pour autant que ce soit possible, la propriété intellectuelle en question ou ses droits à une propriété intellectuelle de tiers à la disposition des groupes de spécialistes respectifs ainsi qu'à l'association **eCH**, pour une utilisation et un développement sans restriction dans le cadre des buts de l'association.

Les normes élaborées par les groupes de spécialistes peuvent, moyennant mention des auteurs **eCH** respectifs, être utilisées, développées et déployées gratuitement et sans restriction.

Les normes **eCH** sont complètement documentées et libres de toute restriction relevant du droit des brevets ou de droits de licence. La documentation correspondante peut être obtenue gratuitement.

Les présentes dispositions s'appliquent exclusivement aux normes élaborées par **eCH**, non aux normes ou produits de tiers auxquels il est fait référence dans les normes **eCH**. Les normes incluront les références appropriées aux droits de tiers.

Annexe A – Références & 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. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874
[RFC2119]	Scott O. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , Internet RFC 2119, March 1997. ftp://ftp.isi.edu/in-notes/rfc2119.txt
[RFC3066]	Harald Tveit Alvestrand, <i>Tags for the Identification of Languages</i> , Internet RFC 3066, January 2001. ftp://ftp.isi.edu/in-notes/rfc3066.txt
[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. http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/
[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. http://www.w3.org/TR/2005/NOTE-xml11schema10-20050511
[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. http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/
[XMLNS]	Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, Henry S. Thompson - Third Edition 8. December 2009 http://www.w3.org/TR/2009/REC-xml-names-20091208/
[OWASP]	XML Security Cheat Sheet, The Open Web Application Security Project, https://www.owasp.org/index.php/XML_Security_Cheat_Sheet

a

Annexe B – Collaboration & vérification

Eisenhut Claude	Eisenhut Informatik AG
Hugentobler Wolfgang	iDPARC AG
Maitre Giles	SBB AG
Müller Willy	Unité de pilotage informatique de la Confédération
Stingelin Martin	Stingelin Informatik GmbH

Annexe C – Abréviations & glossaire

Un glossaire en ligne détaillé est disponible sur le site <http://dret.net/glossary/>.

ASCII	American Standard Code for Information Interchange	Le plus petit dénominateur commun de (presque) tous les codes de caractères (codage à 7 bits), qui a été complété par la suite par les caractères avec signe diacritique (ISO 8859-1) ou, d'une manière tout à fait générale, par des caractères internationaux (Unicode)
Attribut		Un attribut est associé à un élément et contient des informations complémentaires le concernant.
Content Model		Détermine dans un schéma (DTD ou XML) le contenu autorisé d'un type d'élément (définit ainsi l'utilisation admise pour ce type)
DOM	Document Object Model	
DSDL	Document Schema Definition Languages	
DTD	Document Type Definition	Le langage pour schéma défini dans la norme XML standard pour les documents XML; le langage DTD comporte des faiblesses, notamment dans le domaine des types de données
Element		La plus basique des caractéristiques de structuration d'un document XML. Seuls les éléments peuvent être imbriqués, permettant ainsi de former les structures complexes qui sont souvent utilisées dans XML.
Entity		Représente un bout de texte XML qui peut être défini et référencé (il existe différents types d'entités, les principaux étant les General Entities et les Parameter Entities)

GRDDL	Gleaning Resource Descriptions from Dialects of Languages	
HTML	Hypertext Markup Language	Format de document Web, basé sur le langage SGML et incompatible avec XML (XHTML étant une version basée sur XML du langage HTML)
IANA	Internet Assigned Numbers Authority	
IETF	Internet Engineering Task Force	
Information Set		Le modèle de données du langage XML, qui modélise XML comme un ensemble d'unités d'information dotées de propriétés
IRI	Internationalized Resource Identifier	
ISO 8859		Codage de caractères à 8 bits, très utilisé, mais incompatible avec le code UTF-8
Markup		Forme physique d'un document XML (sérialisation de l'ensemble d'informations)
Namespaces		Méthode de désignation et d'utilisation des espaces nominatifs dans XML
Processing Instruction		Instructions de traitement dans les documents XML, qui ne fait pas partie du contenu proprement dit, mais constitue plutôt une information supplémentaire (syntaxe: "<?name content?>")
RDBMS	Relational Database Management System	
RDDL	Resource Directory Description Language	
RDF	Resource Description Framework	
RFC	Request for Comments	
Schema		Description d'une classe de documents XML, les dialectes les plus connus étant DTD (qui fait partie du langage XML lui-même) et XML schéma (qui fait l'objet d'une norme séparée)

Schema Document		Document avec la description des types de données et éléments des instances
Unicode		Norme pour le référencement et le codage de très nombreux caractères
URI/URL	Universal Resource Identifier/Locator	Norme pour l'adressage de ressources sur le Web. Se compose d'un Schema Identifier (par ex. "http:") et d'information supplémentaire pour l'adressage.
UTF-16	Unicode Transformation Format 16	Codage de caractères représentant chaque caractère Unicode en une suite de 16 ou de 32 bits (tous les caractères courants sont codés en 16 bits). Tout logiciel XML doit prendre en charge le codage UTF-16.
UTF-8	Unicode Transformation Format 8	Le codage standard XML pour les caractères Unicode, tout document ASCII étant également un document UTF-8 (UTF-8 code chaque caractère par 1 à 6 octets) Tout logiciel XML doit prendre en charge le codage UTF-8.
W3C	World Wide Web Consortium	
well-formed		Un document XML est "well-formed" s'il respecte les règles de la syntaxe XML
XHTML	Extensible Hypertext Markup Language	Règle l'intégration de ressources XML externes dans les documents XML
XInclude	XML Include	
XLink	Extensible Linking Language	Langage de représentation de documents à structure arborescente et de leur schéma
XML	Extensible Markup Language	Langage de représentation de schéma XML, nettement plus performant que les mécanismes définis dans les DTD XML
XML-Schema		Abréviation fréquemment utilisée (mais non officielle) pour le schéma XML
XSD	XML-Schema Definition Language,	Partie de XSL, langage de programmation spécialisé dans la transformation XML et permettant de transformer un document XML dans une autre représentation (XML, HTML, etc.)

Annexe D – Modifications par rapport à la version précédente

- La norme a fait l'objet d'une révision en profondeur.
- Le document a été restructuré.
- Les normes suivantes ont été annulées et leurs recommandations reprises dans cette norme.
 - eCH-0033: Description des Namespaces XML
 - eCH-0035: Conception de schémas XML
 - eCH-0050: Composants auxiliaires XML
 - eCH-0062: Conception de schémas XML – Synthèse

Annexe E – Liste des illustrations

Figure 1: Vue d'ensemble des principaux objets 5