

**BESCHREIBUNG DES DATENFORMATS
ZUR
ERFASSUNG ALLER WESENTLICHEN BELASTUNGEN VON
OBERFLÄCHENWASSERKÖRPERN DURCH EMISSIONEN
VON STOFFEN AUS PUNKTQUELLEN GEMÄß
EMREGV-OW 2017**

DATENFORMAT-VERSION: v0.90
DOKUMENTERSTELLUNGSDATUM: 4. JULI 2024

Bundesministerium für Land- und Forstwirtschaft,
Regionen und Wasserwirtschaft (BML)

INHALTSVERZEICHNIS

1	Einleitung	3
1.1	Inhalt und Zweck des Dokuments	3
1.2	Aufbau des Dokuments	4
1.3	Verwendung des Dokuments	4
1.4	Technische und rechtliche Grundlagen	5
1.5	Kontakt	5
1.6	Änderungsverzeichnis.....	6
2	Beschreibung des Datenformats	14
2.1	Allgemeine Anmerkungen zum Datenformat	14
2.1.1	Zielsetzungen und Prinzipien der Datenmodellierung	14
2.1.2	XML	15
2.1.3	Zeichencodierung: UTF-8	15
2.1.4	XML Schema.....	15
2.1.5	XSLT	15
2.1.6	Schematron	16
2.1.7	Codelisten	16
2.1.8	Identifikationszeichenketten und natürlichsprachige Angaben	18
2.1.9	XML Schema Design Pattern: Venetian Blind.....	19
2.2	Varianten der XML Schema Definitionen	20
2.2.1	XML Schema 1.1 und XML Schema 1.0 Varianten	20
2.3	Schematron-Prüfungen	20
2.4	Zuordnung von Fachbegriffen zu Datenelementen	21
3	Vorgaben an Software	22
3.1	Allgemeines	22
3.2	Vorgaben in Bezug auf Erstellung und Verarbeitung von Dateninstanzen	22
3.3	Vorgaben in Bezug auf den Umgang mit Codelisten	23

1 EINLEITUNG

1.1 Inhalt und Zweck des Dokuments

Zur Erfüllung gemeinschaftsrechtlicher Berichtspflichten sowie als Grundlage für die Ausarbeitung von Gewässerbewirtschaftungsplänen werden im Emissionsregister Daten betreffend Belastungen von Oberflächenwasserkörpern durch Emissionen von chemischen Stoffen aus Punktquellen gesammelt. Die rechtlichen Grundlagen hierfür wurden in der Verordnung über ein elektronisches Register zur Erfassung aller wesentlichen Belastungen von Oberflächenwasserkörpern durch Emissionen von Stoffen aus Punktquellen 2017 (Emissionsregisterverordnung 2017 – EmRegV-OW 2017) geschaffen.

Registerpflichtig sind im Wesentlichen Wasserberechtigte, Inhaber oder Betreiber von Anlagen, die der Industrieemissionsrichtlinie unterliegen, von Abwasserreinigungsanlagen mit einem Bemessungswert nicht kleiner als 2.000 EW₆₀ für kommunales Abwasser aus Siedlungsgebieten sowie von Abfall(mit)verbrennungsanlagen, wenn die Nennkapazität der gesamten (Mit)verbrennungsanlage zwei Tonnen pro Stunde übersteigt.

Die Datensammlung erfolgt mittels elektronischer Datensätze (Stammdaten), in die zunächst auf der Grundlage vorliegender Bewilligungsbescheide die wasserwirtschaftlichen Stammdaten der registerpflichtigen Einwirkungen eingetragen und an den Registerpflichtigen übermittelt werden. Dieser hat die eingetragenen Daten auf Richtigkeit zu überprüfen und Jahresfrachten von Stoffen (Bewegungsdaten) zu ermitteln und in das Register einzutragen. Daten, für die nationale oder internationale Berichtspflichten bestehen, werden im WISA durch Veröffentlichung einsehbar gemacht. Die Jahresfrachten sind entsprechend den Vorgaben in der Verordnung, je nachdem um welche Stoffe es sich handelt, durch Messung, Berechnung oder Schätzung zu ermitteln.

Inhalt und Zielgruppe des Dokuments

Dieses Dokument beschreibt ein im Auftrag des Bundesministeriums für Land- und Forstwirtschaft, Regionen und Wasserwirtschaft (BML) durch die Umweltbundesamt GmbH in Abstimmung mit der Bund-Länder-Arbeitsgruppe „WISA XML-Schnittstellen Gremium“ ausgearbeitetes XML-Datenformat für Angaben zur Erfassung aller wesentlichen Belastungen von Oberflächenwasserkörpern durch Emission von Stoffen aus Punktquellen gemäß EmRegV-OW 2017. Das Datenformat deckt sämtliche durch diese Verordnung geregelte Dateninhalte ab, und ist so gestaltet, dass es für verschiedene Arten von Datenflüssen verwendbar ist. Es kann somit für den Datenaustausch zwischen beliebigen Akteuren im Kontext der Emissionsregisterverordnung genutzt werden (z.B. zuständige Behörden auf Landes- und Bundesebene, Anlagenbetreiber, Labors, Softwareanbieter usw.).

Dieses Dokument richtet sich in erster Linie an **IT-Analysiker** und **Entwickler**, die für die Umsetzung der Unterstützung des Datenformats zuständig sind, d.h. dafür zuständig sind, dass eine Softwareanwendung die Erstellung und/oder Verarbeitung von Daten des spezifizierten Formats ermöglicht.

Darüber hinaus kann das Dokument auch von **Fachpersonal** genutzt werden, um genauen Aufschluss über Art und Struktur der abbildbaren und übermittelbaren Daten zu erlangen.

1.2 Aufbau des Dokuments

Das Dokument ist wie folgt strukturiert:

- Kapitel 1 enthält eine Einleitung, Hinweise zur Verwendung des Dokuments, eine Auflistung von Literaturhinweisen sowie Kontaktinformationen.
- Kapitel 2 enthält eine Beschreibung allgemeiner Aspekte des XML-Formats zur Erfassung aller wesentlichen Belastungen von Oberflächenwasserkörpern durch Emissionen von Stoffen aus Punktquellen gemäß EmRegV-OW 2017. Die detaillierte Beschreibung der XML-Strukturen und der jeweils darin erwarteten Inhalte befindet sich im HTML-Anhang der Schnittstellenbeschreibung, siehe Abschnitt 1.3
- Kapitel 3 enthält Vorgaben an Software, für die ein Schreiben und/oder Lesen von Daten des vorliegenden XML-Datenformats implementiert wird.

1.3 Verwendung des Dokuments

Zur Schnittstellenbeschreibung gibt es einen **Anhang im HTML-Format** im Spezifikationspaket. Dieser Anhang enthält vorwiegend automatisiert generierte Teile der Schnittstellenbeschreibung, z.B. aus XSD-Dateien automatisiert generierte Beschreibungen der XML-Formate.

Das aktuelle Schnittstellenspezifikationspaket ist online unter der folgenden URL abrufbar: https://test.umweltbundesamt.at/dataharmonisation/spec/emreg/emreg_doc_annex.html

Ergänzend zur Schnittstellenbeschreibung und deren HTML-Anhang gehören die folgenden Inhalte zum Spezifikationspaket. Diese Inhalte sind im HTML-Anhang im Detail aufgelistet und erläutert:

- **XML Schema Definition für EmRegV-OW-XML-Dateien**

Diese XML Schema Definitionen (XSD-Datei, Datei mit „.xsd“-Dateiendung) werden in mehreren Varianten bereitgestellt. Das ist in Abschnitt 2.2 auf Seite 20 näher erläutert.

Anmerkung: Was eine XML Schema Definitions-Datei ist und wozu sie verwendet wird, ist in Abschnitt 2.1.4 auf Seite 15 beschrieben

- **Beispiel-XML-Dateninstanzen**

Diese Dateninstanzen illustrieren das Datenformat mit fiktiven Beispielinhalten. Zudem enthält das Spezifikationspaket XML-Beispiel-Instanzen, die gezielt einzelne Schematron-Prüfungen auslösen und somit Prüfprotokolleinträge generieren

- **Codelisten** (online in diversen Formaten öffentlich zugänglich)

Die zum EmRegV-OW-XML-Format gehörenden Codelisten hat das Ministerium am [EDM-Portal](#) im öffentlich zugänglichen Bereich unter *Zuordnungstabellen* → *EMREG* veröffentlicht.

Zudem existiert ein Codelisten-Webservice für den maschinellen Zugriff auf die Codelisten. Für die Nutzung dieses Webservice wenden Sie sich bitte an den unter Abschnitt 1.5 angegebenen Kontakt.

Anmerkung: Codelisten sind in Abschnitt 2.1.5 auf Seite 15 beschrieben

1.4 Technische und rechtliche Grundlagen

Zum Verständnis des Datenformats können die folgenden Dokumente hilfreich oder erforderlich sein:

RECHT:

- [1] *Emissionsregisterverordnung 2017 – EmRegV-OW 2017, BGBl. II Nr. 207/2017;*
- [2] *Allgemeine Abwasseremissionsverordnung – AAEV, BGBl. Nr. 186/1996*
- [3] *Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE);*
- [4] *Wasserrechtsgesetz 1959 (WRG 1959);*
- [5] *Richtlinie 2000/60/EG des Europäischen Parlaments und des Rates vom 23. Oktober 2000 zur Schaffung eines Ordnungsrahmens für Maßnahmen der Gemeinschaft im Bereich der Wasserpolitik;*
- [6] *Richtlinie 2010/75/EU des Europäischen Parlaments und des Rates vom 24. November 2010 über Industrieemissionen (integrierte Vermeidung und Verminderung der Umweltverschmutzung);*

TECHNISCHE STANDARDS:

- [7] *Cascading Style Sheets (CSS); W3C Recommendations; <https://www.w3.org/Style/CSS/>;*
- [8] *Electronic Business Extensible Markup Language (ebXML) – Part 5: ebXML Core Components Technical Specification, ISO/TS 15000-5:2005, Version 2.01 (ebCCTS);*
- [9] *GS1 XML Transport Instruction and Response Implementation Guide; Issue 1.0, Approved, 3-Sep-2012; <https://www.gs1.org/>;*
- [10] *HTML (Hypertext Markup Language) 5.2, W3C Recommendation, 14 December 2017; <https://www.w3.org/TR/html/>;*
- [11] *Information technology — Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation — Schematron; ISO/IEC 19757-3:2020; <https://www.iso.org/>;*
- [12] *UN/CEFACT Core Components Library (CCL); <http://www.unece.org/>;*
- [13] *Universal Multiple-Octet Coded Character Set (UCS); ISO/IEC 10646:2003*
- [14] *XML Extensible Markup Language 1.1 (Second Edition), W3C Recommendation 16 August 2006, edited in place 29 September 2006; <http://www.w3.org/>;*
- [15] *XML Schema Definition Language (XSD) 1.1 Part 1: Structures, W3C Recommendation 5 April 2012; <http://www.w3.org/>;*
- [16] *XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation 5 April 2012; <http://www.w3.org/>;*
- [17] *XSL Extensible Stylesheet Language Version 1.1; Formatting Objects (XSL-FO); W3C Recommendation 05 December 2006; <https://www.w3.org/TR/xsl11/>;*
- [18] *XSL Transformations (XSLT) Version 3.0, W3C Recommendation 8 June 2017; <https://www.w3.org/TR/xslt/>;*
- [19] *GML 3.2.1, Open Geospatial Consortium Standard 2007; <http://www.opengeospatial.org/standards/gml/>;*

1.5 Kontakt

Bitte wenden Sie sich mit Fragen, Anmerkungen und Verbesserungsvorschlägen zur Schnittstelle und deren Spezifikation an xml-helpdesk@umweltbundesamt.at oder telefonisch an +43-(0)1-313 04/8008.

1.6 Änderungsverzeichnis

Datenformat-Version	Datum	Änderungen
0.90	04.07.2024	Siehe HTML-Anhang zur Schnittstellenbeschreibung
0.33	07.03.2024	Siehe HTML-Anhang zur Schnittstellenbeschreibung
0.32	09.03.2023	Siehe HTML-Anhang zur Schnittstellenbeschreibung
0.31	28.09.2022	<ol style="list-style-type: none"> 1. Aus Vereinheitlichungsgründen Umstellung der XML Schema Definition auf die Einstellung <i>elementFormDefault "qualified"</i>, anstelle zuvor „<i>unqualified</i>“. D.h. es besteht nunmehr die Vorgabe, dass auch lokal definierte Unterelemente mit Namespace-Präfix zu qualifizieren sind. Entsprechende Anpassung aller XML-Beispieldateien und formalen Prüfregele 2. Anpassung der Schematron-Prüfungen und XML-Beispieldateien so, dass jede Prüfung unabhängig von anderen Prüfungen ausgelöst werden kann, bzw. dass es für jede Prüfung eine Beispieldatei gibt, welche genau diese aber keine anderen Prüfungen auslöst 3. Aus Vereinheitlichungsgründen Aktualisierung der IDs von Schematron-Prüfungen auf ein neues ID-Schema. Die Datei <i>Map_Prüfregel_ID.xlsx</i> enthält eine Zuordnung zwischen den in v0.31 verwendeten IDs und den in vorherigen Versionen verwendeten IDs 4. Anpassung mehrerer Dateinamen und Verzeichnisnamen, aus Vereinheitlichungsgründen 5. Aktualisierung des Formats der HTML-Prüfregel-Illustration <i>emreg_formalvalidationrules.html</i> 6. Anpassung von einzelnen XML-Beispieldateien und Schematron-Prüfungen so, dass für es für jede Prüfung eine oder mehrere Beispieldateien gibt, bei welcher genau die betreffende Prüfung anschlägt, aber keine der anderen Prüfungen 7. Bereitstellung einer leicht navigierbaren und durchsuchbaren HTML-Beschreibungs-Datei für jede XSD-Datei, als Alternative zur Nutzung der annotierten XSD-Dateien mit Werkzeugen wie <i>Altova XMLSpy</i>

0.30	28.01.2022	<p>1. Zusätzliche ID-Arten i.Z. mit nicht-natürlichen Personen (<i>Organization/ID</i>) unterstützt: <u>Zusätzlich</u> zu den bereits in v0.29 unterstützten ID-Angaben - EDM Personen-GLN, Firmenbuchnummer, Vereinsregisternummer, Ergänzungsregisternummer und Wasserbuch-ID – unterstützt v0.30 die folgenden beiden ID-Angaben:</p> <ol style="list-style-type: none"> 1. Unternehmensregister-Kennzahl KUR 2. Unternehmensregister-Sekundär-ID SEK <p>2. Schematron-Prüfung S02 angepasst:</p> <ul style="list-style-type: none"> • In v0.29 war gemäß dieser Prüfung die Angabe einer EDM-Personen-GLN zu nicht-natürlichen Personen erforderlich • In v0.30 ist gemäß dieser Prüfung die Angabe von <u>mindestens</u> einer der folgenden IDs erforderlich: <ol style="list-style-type: none"> 1. EDM-Personen-GLN 2. Firmenbuchnummer 3. Vereinsregisternummer 4. Ergänzungsregisternummer
0.29	06.10.2021	<p>Schematron-Prüfungen angepasst:</p> <ol style="list-style-type: none"> 1. Parameter-bezogene Prüfungen dahingehend angepasst, dass diese auch bei Verwendung von EHPC-IDs korrekt funktionieren (D12, D16, D24, D56, D62) <p>Schematron-Prüfungen ergänzt:</p> <ol style="list-style-type: none"> 1. Bezug auf nicht mehr als eine Norm je Methode (D68) 2. Für Einzelmesswerte, die zum selben Parameter, Teilstrom und Berichtszeitraum gehören, muss die allfällig vorhandene Methoden-Angabe übereinstimmen (D69, D70) <p>Anmerkung: Gemäß Prüfung D55 wird bei der Übermittlung in EMREG-OW für Einzelmesswerte eine Angabe von Methoden gegenwärtig nicht unterstützt. Die Ergänzung um die Prüfungen D68, D69, D70 erfolgte vorsorglich.</p>
0.28	20.10.2020	<ol style="list-style-type: none"> 1. Möglichkeit hinzugefügt, Parameter anstelle wie bisher per GTIN alternativ per EHPC-Code (Code aus der „Einheitlichen Parametercodierung“) zu identifizieren. Dazu Schematron-Prüfregel D66 ergänzt, mit der sichergestellt wird, dass GTINs und EHPC-Codes in XML-Dateninstanzen nicht gemischt/gemeinsam verwendet werden. <p>Anmerkung: Zum aktuellen Zeitpunkt, 20.10.2020, liegen noch keine realen EHPC-Codes vor. Sobald reale EHPC-Codes vorliegen, müssen noch die Schematron-Prüfungen angepasst werden, da diese zum Teil überprüfen, ob pH-Wert oder Temperatur als Parameter angegeben sind, und dies in Zukunft auch im Falle der Angabe per EHPC-Code funktionieren muss.</p>

		<ol style="list-style-type: none"> 2. Aus Jahreswassermengen (<i>ReferencePeriodProperty</i>) und Wasservolumen-Grenzwerten (<i>WaterVolumeLimitValue</i>) das Element <i>PropertyKindID</i>, mit dem ausschließlich der Parameter „Volumen“ ausgedrückt werden konnte, entfernt, um sie die Verwendung des Datenformats mit EHPC-Codes zu vereinfachen. 3. Zu Adressen (<i>Address</i>) das Element <i>ID</i> ergänzt, und damit die Möglichkeit, Adresscodes gemäß österreichischem Adressregister https://www.adressregister.gv.at abzubilden. 4. Zu Standorten (<i>Site</i>) das Element <i>DistrictID</i> ergänzt, und damit die Möglichkeit, den österreichischen Bezirk zu identifizieren, in welchem sich der Standort befindet. Die Identifikation erfolgt mittels Eintrag aus Codeliste 4786. Da diese Angabe für die korrekte Weiterverarbeitung in der EMREG-OW-Anwendung benötigt wird, wurde zudem die Schematron-Prüfregel D67 ergänzt, mit welcher sichergestellt wird, dass zu jedem Standort der Bezirk angegeben ist. 5. Datentypen auf vereinheitlichte Datentypen umgestellt. Dies wirkt sich nur auf Benennungen und Beschreibungen innerhalb der XSD-Datei aus, ändert aber nichts am dadurch definierten Datenformat. Einzige Ausnahme sind Beschreibungs-Datenlemente, z.B. <i>Description</i>-Elemente. Diese waren zuvor auf maximal 1024 Zeichen limitiert, und sind nach der Umstellung auf vereinheitlichte Datentypen nunmehr auf maximal 1000 Zeichen begrenzt.
0.27	11.09.2020	<ol style="list-style-type: none"> 1. Möglichkeit hinzugefügt, eine Anmerkung zu den Emissionen aus einer Anlage im Berichtsjahr auszutauschen, siehe <i>EmissionRemark</i> in <i>InstallationReferencePeriodResult</i>. Die EMREG-OW-Anwendung sieht eine solche Angabe vor, etwa in der Benutzeroberfläche, in den Datenaustausch-Formaten gab es eine solche Angabe zuvor jedoch noch nicht. 2. Möglichkeit hinzugefügt, zu Abwasserreinigungsanlagen spezifische Anmerkungen auszutauschen, siehe <i>Remark</i> in <i>WasteWaterTreatmentInstallation</i>. Die EMREG-OW-Anwendung sieht eine solche Angabe vor, etwa in der Benutzeroberfläche, in den Datenaustausch-Formaten gab es eine solche Angabe zuvor jedoch noch nicht. 3. Die Information, ob Jahreswerte aus den Messergebnissen eines vorangehenden Messjahres berechnet sind, und wenn ja, aus welchem Messjahr, war zuvor im Schema pauschal für alle zu einem Teilstrom ermittelten Parameter vorgesehen, unter <i>WasteWaterStreamReferencePeriodResult</i>. Aufgrund entsprechenden fachlichen Inputs ist dies dahingehend abgeändert, dass nunmehr für jeden einzelnen Jahreswert angegeben werden kann, ob bzw. aus welchem Messjahr dieser berechnet wurde. Siehe <i>CalculationBasisPeriod</i> in <i>ReferencePeriodParameterProperty</i>.

		<ol style="list-style-type: none"> 4. Elemente <i>WaterVolumeLimitValue/Description</i> und <i>WaterVolumeLimitValue/ConditionDescription</i> entfernt, siehe Beschreibung von <i>WaterVolumeLimitValue</i>. 5. Elemente <i>LimitValueSet/Description</i> und <i>LimitValueSet/Remark</i> entfernt, siehe Beschreibung von <i>LimitValueSet</i>. 6. Schematron-Prüfregel D63 hinzugefügt, mit welcher überprüft wird, dass Jahresfrachten im Zulauf für keine anderen Anlagen als kommunale Abwasserreinigungsanlagen angegeben sind, da Zulauffrachten zu anderen Arten von Anlagen in der EMREG-OW-Anwendung spezifikationskonform nicht angezeigt werden. 7. Schematron-Prüfregel D28, die besagt, dass es zu jeder in einer XML-Instanz angegebenen Anlage Angaben zu Teilströmen aus dieser Anlage geben muss, von einer Fehler-Regel auf eine Hinweis-Regel abgeändert, aufgrund von Fach-Input, dass nicht notwendigerweise jede Anlage, in die indirekt eingeleitet wird, EMREG-meldepflichtig ist. 8. Schematron-Prüfregel D62 hinzugefügt, mit der die Angabe textueller Nebenbestimmungen für die „Nicht-Stoff-Parameter“ pH-Wert und Temperatur verhindert wird, da solche Angaben in der EMREG-OW-Anwendung nicht verwaltet werden können. 9. Schematron-Prüfungen für EMREG-OW-XML-Exporte entfernt, und in Abschnitt 2.3 auf S.20 beschrieben, dass die bereitgestellten Schematron-Prüfungen nur für XML-Importe spezifiziert sind, aber nicht auf XML-Exporte anzuwenden sind, und weshalb das so ist. 10. In den XML Schema Definitionen in patterns (regular expressions) den Bindestrich (Dash) durchgängig mit Backslash versehen („escaped“), um so die Kompatibilität mit einzelnen XSD-Werkzeugen, welche die XSD-Ausprägung von regular expressions nicht unterstützen, zu verbessern. 11. Datenelement-Beschreibung zur Handhabung der Angabe eines Erstellers der XML-Instanz beim EMREG-OW-Import ergänzt, siehe <i>DocumentEvent/PartyReferenceID</i>. 12. Datenelement-Beschreibungen betreffend einer möglichen Angabe von Jahres-Ergebnissen unterhalb Bestimmungsgrenzen bzw. Nachweisgrenzen präzisiert/ergänzt, siehe Beschreibung von <i>ReferencePeriodParamaterProperty</i>.
0.26	09.06.2020	<ol style="list-style-type: none"> 1. Zu Kompartimenten (<i>Compartment</i>) sind nun nicht mehr Angaben von IDs (GLNs, Global Location Numbers) vorgesehen

		<ol style="list-style-type: none"> 2. Systematik ergänzt, mit der bei Übermittlung von Abfluss-Einzelmesswerten die Parameterkategorie (<i>PropertyKindID</i>, Prioritärer Parameter, usw.) mit übermittelt werden kann und muss. Die Angabe ist in <i>WasteWaterStreamReferencePeriodResult</i> vorgesehen, gleichartig wie bei der Übermittlung von Jahresfracht-Informationen. Die Fracht selbst (<i>ValueAssignmentStatement</i>) entfällt dabei bei der bloßen Übermittlung von Einzelmesswerten 3. Einige Schema-Elemente, die zuvor gemäß XSD zwingend anzugeben waren (minOccurs-Wert von 1), werden nun nicht mehr per XSD auf Vorhandensein geprüft, sondern per Schematron-Prüfregeln, z.B. <i>WasteWaterStreamReferenceID</i> in <i>LimitValueSet</i>. Die Verletzung der betreffenden Voraussetzungen, d.h. das Fehlen von Angaben bzw. Elementen, führt auf diese Weise zu aussagekräftigen textuellen Rückmeldungen, und nicht mehr bloß zu einem vergleichsweise schwierig zu interpretierenden Hinweis auf eine Schema-Verletzung 4. Einige Bedingungen, deren Einhaltung zuvor per XML Schema 1.1 assert-Statements überprüft wurden, werden nun per Schematron-Prüfregeln geprüft. Die Verletzung der betreffenden Voraussetzungen führt auf diese Weise zu aussagekräftigen textuellen Rückmeldungen, und nicht mehr bloß zu einem vergleichsweise schwierig zu interpretierenden Hinweis auf eine Schema-Verletzung 5. Schematron-Prüfungen ergänzt. Aktueller Stand: Anzahl von 72 Kriterien, deren Einhaltung geprüft wird 6. Bei Standort-Angaben (<i>Site</i>) das Element <i>Name</i> (Standort-Bezeichnung) zum Element <i>ID</i> verschoben, zur Vereinheitlichung mit anderen Angaben wie jenen zu Anlagen und Teilströmen 7. Die Schematron-Prüfungen (<i>emreg_validationrules.sch</i> sowie dazugehörige XSLT-Datei) so angepasst, dass sie für „Stamm- und Bewegungs-Daten“ sowie „Bewegungsdaten only“-Dateninstanzen gleichermaßen verwendbar sind. Zuvor waren die Schematron-Prüfungen ausschließlich für „Stamm- und Bewegungs-Daten“ Dateninstanzen nutzbar, für „Bewegungsdaten only“ waren zuvor noch keine Schematron-Prüfungen definiert 8. Einige der Schematron-Prüfungen aus <i>emreg_validationrules.sch</i> gelten nur für Datenimporte in die EMREG-OW Anwendung des Bundesministeriums, nicht aber für Exporte aus dieser Anwendung. Separate Schematron-Datei mit zugehöriger XSLT-Datei ergänzt, <i>emreg_validationrules_exportonly.sch</i>, welche die Auswahl von Schematron-Prüfungen enthält, die auch für Exporte gelten
--	--	---

		<p>9. Für sämtliche der Schematron-Prüfungen im Ordner <i>xml_example_stammbew</i> jeweils eine XML-Beispieldatei bereitgestellt, die gültig bezüglich der XML Schema Definition (XSD) ist, aber ungültig bezüglich der im Dateinamen genannten Schematron-Prüfung (z.B. <i>example_violateD08.xml</i> verletzt die Schematron-Prüfung D08). Im korrespondierenden <i>validationresults</i>-Ordner sind die Schematron-Prüfergebnisse ersichtlich. Insbesondere können in diesem Ordner daher auch die Schematron-Prüfergebnistexte eingesehen werden</p> <p>10. In der Schnittstellenbeschreibung Schematron ergänzt (Abschnitte 2.1.6 und 2.3)</p>
0.25	26.05.2020	<p>1. In <i>WasteWaterStream</i> schließen die beiden Angaben <i>OriginInstallationReferenceID</i> und <i>DrainedArea</i> einander nicht mehr aus (<i>sequence</i> anstelle von <i>choice</i>)</p> <p>2. Ergänzung von Schematron-Prüfungen</p>
0.24	03.03.2020	<p>Verbesserungen gemäß Reviewergebnissen:</p> <ul style="list-style-type: none"> • Formatanpassungen: <ul style="list-style-type: none"> ○ Je Abwasserbehandlungsanlage kann nur EIN Siedlungsgebiet zugeordnet werden ○ Datenformat unterstützt Angabe von Messmethoden zu Einzelmesswerten ○ Korrektur der Multiplizitäten bei Einzelmesswerten ○ Bezugsjahr-Angabe ergänzt (Semantik: aus den Werten welchen Jahres wurden die Werte des aktuellen Jahres mithilfe des eingeleiteten Wasservolumens automatisch berechnet)

		<ul style="list-style-type: none"> • Ergänzung von Prüfregeln (Schematron): <ul style="list-style-type: none"> ○ Bei der Übermittlung in EMREG-OW immer ENTWEDER Jahresfrachten ODER Einzelmesswerte ○ Bei der Übermittlung in EMREG-OW immer Angaben zu nur EINEM Berichtsjahr ○ Bei der Übermittlung in EMREG-OW keine Zufluss-Einzelmesswerte (nur Abfluss zulässig) ○ Bei der Übermittlung in EMREG-OW keine keine Messmethoden-Angabe ○ Bei Einzelmesswerten nur mg/l, µg/l und ng/l als Einheiten zulässig ○ pH-Wert und Temperatur in Einzelmesswerten unzulässig ○ Als Abwasserbehandlungsanlagen angegebene Anlagen sind als Herkunftsanlagen von Indirekteinleitungen nicht zulässig ○ Angaben nur zu solchen Anlagen, die als Herkunft von Abwasserströmen aufscheinen ○ Je Parameter und Teilstrom nicht mehr als 12 Einzelmesswerte
0.23	05.02.2020	<p>Korrekturen gemäß Reviewergebnissen:</p> <ul style="list-style-type: none"> • Die Parameterkategorie (PropertyCategoryID) ist nun nur noch in Angaben zu Berichtszeitraum-Abfluss-Ergebnissen verpflichtend. Beim Zufluss ist hingegen nunmehr keine Parameterkategorie-Angabe mehr vorgesehen • Bei den Zahlenwerten sind nun bis zu 10 Nachkommastellen zulässig (zuvor 5) • Bei den Wassermengen ist nun keine Ergebnisart-Angabe (z.B. kleiner Bestimmungsgrenze) mehr vorgesehen • Bei den Berichtszeitraum-Parameter-Ergebnissen ist als Einheit nunmehr nur noch „kg“ (Kilogramm) zulässig. Die Angabe von ph-Wert und Temperatur ist an dieser Stelle nunmehr nicht mehr zulässig

0.22	27.01.2020	<p>Korrekturen gemäß Reviewergebnissen:</p> <ul style="list-style-type: none"> • Beim Bescheid (<i>Decision</i>) sind nun nicht mehr Angaben zu EU-Rechtsgrundlagen (Element <i>EULegislation</i>) vorgesehen • Die „GIS-IDs“ (<i>ShortName</i>-Element) sind nunmehr von einem Datentyp, der nicht mehr als 50 Zeichen zulässt • Bemessungswert, Indirekteinleiter-Belastung und Einwohnerzahl (<i>TechnicalCapacityMeasure</i>, <i>IndirectDischargeLoadMeasure</i> und <i>PopulationLoadMeasure</i>) sind nunmehr von einem Datentyp, der nur Ganzzahlen zulässt
0.21	27.01.2020	<ul style="list-style-type: none"> • Regenwetter/Trockenwetter als strukturierte Angabe bei Wassermengenbeschränkungen ergänzt • Abbildung des Kanalisationsunternehmens entfernt • Diverse Prüfungen ergänzt, die für eine Verarbeitung der Daten erforderlich sind, z.B. Angabe einer Wasserbuch-ID bei Teilströmen, Angabe einer GLN bei nicht-natürlichen Personen, Angabe einer Website (aber keiner sonstigen Kontaktangaben) bei nicht-natürlichen Personen, usw. • Schematron-Prüfungen mit sprechenden Prüfprotokoll-Texten inklusive Bezügen auf XML-Instanz-Inhalte ergänzt • Komplexere existierende asserts in Schematron-Prüfungen transferiert
0.10 bis 0.20	17.12.2019	Einarbeitung der Rückmeldungen aus dem Review, sowie Vervollständigung des Entwurfs
0.09	31.01.2019	Erstentwurf zur Abstimmung mit dem Auftraggeber und interner Review
0.01 bis 0.08	ab 20.03.2018	Interne Entwürfe und Überarbeitungen

2 BESCHREIBUNG DES DATENFORMATS

2.1 Allgemeine Anmerkungen zum Datenformat

2.1.1 Zielsetzungen und Prinzipien der Datenmodellierung

Im Folgenden sind einige wichtige allgemeine Zielsetzungen und Prinzipien angeführt, die bei der Ausarbeitung des Datenformats angewendet wurden. In den Folgeabschnitten wird dann näher auf einzelne technische Standards und Modellierungsprinzipien eingegangen.

- **Zukunftstauglichkeit – Flexibilität in Bezug auf allfällig im Laufe der Zeit erforderliche Änderungen:** Das XML-Datenformat ist so konzipiert, dass es möglichst langfristig verwendet werden kann. Das Grundprinzip ist, dass Datenformat-Vorgaben, von denen nicht ausgeschlossen werden kann, dass sie sich im Laufe der Zeit ändern, als Codelisten abgebildet sind. Anwendungen, die das Lesen oder Schreiben eines Datenformats unterstützen, können so implementiert werden, dass ein Aktualisieren von lokalen Codelisten-Kopien automatisiert ohne die Notwendigkeit der Anpassung von Software (Um- oder Neuprogrammierungen) erfolgt.

Beispiel: Für die Angabe von Wasserparametern (Frachten, Konzentrationen, usw.) kommen im Datenformat Codelisten zum Einsatz. Kommt es zukünftig zu Änderungen bei den Parametern, so gibt es lediglich eine Aktualisierung der entsprechenden Codeliste. Die XML Schema Definition hingegen bleibt gänzlich unverändert. Die Datenformat-Spezifikation ermöglicht es daher, Datenformat-lesende oder -schreibende Anwendungen so zu implementieren, dass die Berücksichtigung von Änderungen bei Parametern keinen Entwicklungsaufwand und auch keinen sonstigen administrativen Aufwand erfordert, sondern die Anwendungen einfach weiterverwendet werden können.

Bemerkung: Dem Datenformat wird durch die Verwendung von Codelisten eine Flexibilität in Bezug auf Anpassungen verliehen. Dies ist als Vorkehrung zu verstehen, und nicht als Absicht, die Codelisten tatsächlich häufig zu ändern. Stattdessen werden Codelisten nur dann geändert, wenn dies fachlich erforderlich ist.

- **Zukunftstauglichkeit – Orientierung an Standards:** Es werden Jahr für Jahr neue Datenformate von unterschiedlichsten Einrichtungen veröffentlicht. Unter anderem wird auf EU-Ebene an diversen standardisierten Datenformaten gearbeitet, z.B. im Rahmen der INSPIRE und SEIS-Initiativen. Vor diesem Hintergrund erscheint auch die Wahrscheinlichkeit hoch, dass in absehbarer Zeit neue Datenformat-Spezifikationen, eventuell Standards, entstehen, welche ähnliche Inhalte abdecken wie das hier vorgestellte Datenformat. Um sicherzustellen, dass das hier vorgestellte Format möglichst in Einklang mit künftigen Entwicklungen steht, d.h. möglichst widerspruchsfrei zu künftig entstehenden Datenformaten ist, ist die Spezifikation des Formats sehr stark an bestehenden Standards orientiert. Eine vollständige Auflistung der berücksichtigten Vorgaben würde den Rahmen sprengen, daher an dieser Stelle eine beispielhafte Aufzählung:

1. World Wide Web Consortium Extensible Markup Language (XML)
2. World Wide Web Consortium XML Schema
3. UN/CEFACT Core Components Technical Specification
4. UN/CEFACT Core Component Library
5. Joint Committee for Guides in Metrology International vocabulary of metrology (VIM)
6. International System of Units (SI)
7. EU-Directive Spatial Information Infrastructure INSPIRE (2007/2)
8. EU-Council Directive Units of Measurement (1980/181)
9. Unified Code for Units of Measure
10. ISO 19100 Series Geographic Information

- **Kriterium Technische Verarbeitbarkeit; Nicht-Kriterien inhaltliche Richtigkeit und Vollständigkeit von Daten:** Das XML-Datenformat ist so spezifiziert, dass eine grundlegende technische Verarbeitbarkeit, insbesondere das Speichern in relationalen Datenbanken, von gemäß dem Datenformat repräsentierten Informationen sichergestellt ist. Darüber hinausgehende Anforderungen an die inhaltliche Richtigkeit und Vollständigkeit von Daten werden durch das XML-Datenformat nicht berücksichtigt. Das ist bewusst so gehalten, um möglichst keine Barrieren für die Repräsentation von Daten, beispielsweise von bereits bestehenden Datensammlungen, in dem

Datenformat zu schaffen: Es handelt sich um eine Anforderung an das Datenformat, dass damit auch die Übermittlung von unvollständigen oder unplausiblen Daten grundsätzlich möglich ist.

2.1.2 XML

XML-Dateien (*Extended Markup Language* Dateien) sind Text-Dateien, in welchen die Inhalte mit Namen gekennzeichnet sind und eine hierarchische Struktur aufweisen.

XML-Dateien zeichnen sich insbesondere dadurch aus, dass sie gut für die maschinelle Verarbeitung geeignet sind, und zugleich auch für Menschen relativ gut lesbar, etwa im Vergleich zu Binärformaten.

XML [15][16] ist ein vom *World Wide Web Consortium* (<http://www.w3.org>) veröffentlichter Standard.

2.1.3 Zeichencodierung: UTF-8

XML Dateien können – so wie alle Text-Dateien – in verschiedenen Zeichencodierungen gespeichert sein, z.B. ISO 8859-1 oder UTF-8.

Unicode und *UTF-8* [12] sind als ISO-Standard veröffentlicht. UTF-8 zählt zu den gebräuchlichsten Zeichencodierungen. Auf bereits bestehende Funktionen zur Speicherung von Text in UTF-8 Zeichencodierung kann in nahezu allen Programmiersprachen zurückgegriffen werden. Auch alle gängigen textverarbeitenden Programme unterstützen diese Codierung.

Das in diesem Dokument beschriebene Datenformat setzt eine Codierung von Daten in UTF-8 voraus – siehe die Vorgabe mit der ID 628 auf Seite 22.

2.1.4 XML Schema

Die Daten müssen gewisse Strukturvorgaben einhalten, um verarbeitet werden zu können. Diese Strukturvorgaben betreffen insbesondere Anzahl, Anordnung und Kennzeichnung der zu übermittelnden Inhalte, und sind daher mit Formularvorlagen im papierbasierten Datenaustausch vergleichbar.

Für die Festlegung von Strukturvorgaben für XML Dateien existieren mehrere Standards. Der verbreitetste davon ist *XML Schema* [15], [16] ein ebenfalls vom *World Wide Web Consortium* (<http://www.w3.org>) veröffentlichter Standard.

Die Strukturvorgaben für EmRegV-OW-XML-Dateien sind als XML Schema definiert. Diese XML Schema Dateien besitzen die Dateiendung ".xsd" und werden gemeinsam mit diesem Beschreibungsdokument zur Verfügung gestellt.

Eine XML Dateninstanz (*Anm.: d.h. eine mit Dateninhalten „befüllte“ XML-Datei des XML Schemas*) heißt *gültig* bezüglich eines XML Schemas, wenn sie die im XML Schema definierten Strukturvorgaben einhält. Es gibt Anwendungen und Funktionsbibliotheken, sogenannte *XML Schema Validatoren*, mit deren Hilfe es möglich ist, bei vorliegendem XML Schema und vorliegender XML Dateninstanz die XML Dateninstanz zu validieren, d.h. deren Gültigkeit bezüglich des XML Schemas zu überprüfen. Solche Validierungen lassen sich sehr einfach in Software integrieren. In einer Server-Client-Datenaustausch-Architektur kann beispielsweise auch Client-seitig die Gültigkeit von XML-Dateninstanzen einfach überprüft werden.

Für XML-Dateien, die bezüglich des EmRegV-OW-XML-Schemas nicht gültig sind, ist bei einer Übermittlung bzw. einem Einlesen mit einer automatischen Zurückweisung zu rechnen.

2.1.5 XSLT

XSLT [18] steht für *Extensible Stylesheet Language Transformation*. Es handelt sich um eine *Recommendation* des *World Wide Web Consortium*.

XSLT nutzt den XML-Standard *XPath*, welcher ebenfalls eine *Recommendation* des *World Wide Web Consortium* ist.

Hauptzweck von XSLT ist das Transformieren (Umwandeln) von XML-Dateien. Die möglichen Anwendungen sind vielfältig, z.B. das Filtern von Daten, das Konvertieren von Strukturen/Formaten oder die Erstellung von XML-Prüfprotokollen.

XSLT ist eine XML-Sprache, in der solche Umwandlungen ausgedrückt werden können. XSLT-Dateien werden typischerweise mit der Dateiendung `.xsl` oder `.xslt` versehen.

Zur Anwendung von XSLT, d.h. zum Durchführen der Umwandlungen, wird ein sogenannter *XSLT Prozessor* benötigt. Dabei handelt es sich um Software, welche geeignet ist, XSL-Transformationen durchzuführen.

Dieser Software wird als Input folgendes übergeben:

1. Ausgangs-(XML-)Datei(en)
2. XSLT-Datei

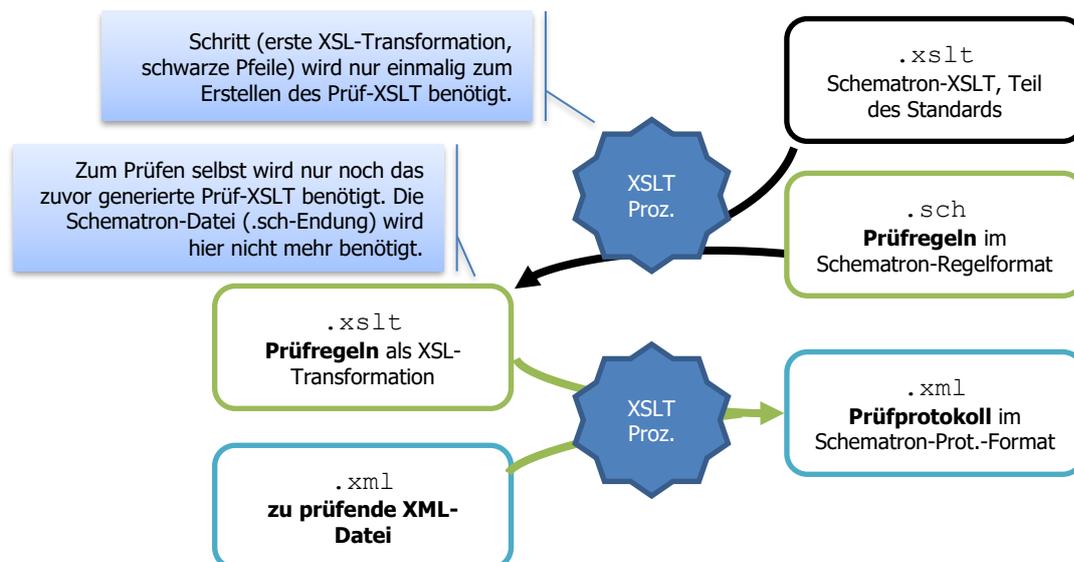
Daraus berechnet die Software die entsprechende Transformation der Ausgangs-Dateien, also z.B. eine Filterung von Daten, eine Konvertierung der Struktur oder ein Prüfprotokoll.

Am Markt stehen mehrere XSLT-Prozessoren zur Verfügung, auch zur Einbettung in Software. Zu den bekannteren zählen Saxon, <http://saxonica.com>, sowie RaptorXML von Altova, <https://www.altova.com>.

2.1.6 Schematron

Schematron [11] ist ein ISO-Standard zur regelbasierten Validierung von XML-Dateninstanzen. Schematron nutzt die XML-Standards XPath und XSLT (beides *Recommendations* des *World Wide Web Consortium*). Einzuhaltende Bedingungen (*assert*) bzw. zu berichtende Bedingungen (*report*) werden in Schematron-Dateien als XPaths ausgedrückt. Schematron-Dateien sind XML-Dateien, deren Dateiendung typischerweise „.sch“ ist.

Technisches Kernelement von Schematron ist das Schematron XSLT-Stylesheet. Mit diesem werden Schematron-Dateien (.sch) in XSLT-Stylesheets umgewandelt, namentlich „Validierungs-XSLTs“, d.h. XSLT-Dateien zur Erstellung von XML-Prüfergebnissen. Solche Prüfergebnisse folgen dem standardisierten Schematron-Prüfergebnis-Format. Dabei handelt es sich um ein sehr einfaches XML-Format, das die Ergebnisse der Prüfungen auflistet. Die Prüfergebnisse können beispielsweise sehr einfach durch Software aus der XML-Datei ausgelesen und in einer Benutzeroberfläche angezeigt werden, oder mit einer weiteren Transformation in ein HTML-Format umgewandelt werden. Das folgende Diagramm illustriert die Verwendung von Schematron.



Die Regeln, und in Folge die Einträge in Schematron-Prüfprotokollen, können verschiedenen Klassen zugeordnet sein, z.B. Fehler und Warnung.

2.1.7 Codelisten

Das Datenformat sieht unter anderem die Identifikation von Objekten vor, und zwar nach den folgenden beiden Prinzipien:

1. Identifikation von Anlagen, Einleitepunkten, Unternehmen, Abwasserteilströmen usw. Zur Identifikation solcher sogenannter Stammdatenobjekte ist die Angabe von Identifikatoren wie Wasserbuchnummer, Firmenbuchnummer, Unternehmensregister-Behörden-GLNs, vom Betreiber verwendete Identifikationszeichenkette, usw. vorgesehen.
2. Identifikation von Objekten aus vorgegebenen Listen. Ein Beispiel ist die Auswahl einer Größeneinheit für einen Volumenangabe, z.B. Kubikmeter, aus einer vorgegebenen Liste von Größeneinheiten. Solche Listen, die die in einem bestimmten Kontext vorgegebene Auswahl von

Einträgen festlegen, werden **Codelisten** genannt. Für jeden Eintrag existiert ein Code, z.B. eine *GTIN (Global Trade Item Number)*, der diesen Eintrag identifiziert.

Die in einem bestimmten Kontext zulässigen Codes, z.B. die Codes, die zur Auswahl einer Größeneinheit zulässig sind, sind bewusst **nicht im XML Schema** hinterlegt. Der wichtigste Grund dafür: Codelisten können sich häufiger ändern, ohne dass sich am Datenformat bzw. an der Schnittstelle etwas ändert. Entsteht beispielsweise aufgrund einer Unabhängigkeitserklärung ein neuer Staat, so muss die Liste der zur Auswahl stehenden Nationalstaaten angepasst werden. Am Datenformat selbst hat sich dadurch aber nichts geändert. Wären die zulässigen Codes im XML Schema hinterlegt, so müsste bei jeder Aktualisierung von Codelisten auch das XML Schema aktualisiert werden. Die XML Schema Definitionen müssten deshalb häufiger aktualisiert werden, und „harmlose“ Aktualisierungen (Codelisten-Update) wären von aufwändigeren Aktualisierungen (z.B. solchen, die es für gewöhnlich notwendig machen, Software anzupassen, neu auszurollen, usw.) schwerer unterscheidbar. Das wird durch die „Auslagerung“ der Codelisten vermieden.

Anstelle der Hinterlegung im XML Schema sind die **Codelisten im Internet veröffentlicht** (siehe Link im Abschnitt 1.3 auf Seite 4). Zudem steht ein Webservice für den Bezug von Codelisten zur Verfügung. Um dieses Webservice verwenden zu können, ist eine Kontaktaufnahme mit dem BMNT erforderlich (siehe Abschnitt 1.5).

Die **Verweise auf Codelisten** sind direkt in den **Datenelementbeschreibungen** angegeben, typischerweise durch den Zusatz „(Codeliste xxxx)“ zum Beschreibungstext, z.B. „Identifikation der Sprache, in der die Beschreibung angegeben ist (Codeliste 7632)“.

Die Verwendung einer großen Zahl von Codelisten im Datenformat dient der Ermöglichung von Flexibilität (einfacher Anpassbarkeit). Für Software, die das Erstellen und/oder das Verarbeiten von Dateninstanzen des vorliegenden Formats unterstützt, bedeutet das keine unbedingte Notwendigkeit, die Software ebenso flexibel bezüglich Anpassungen zu gestalten. Nicht alles, das im Datenformat als Codeliste realisiert wird, muss in solcher Software ebenfalls mittels Codelisten abgebildet sein. Das gilt insbesondere für Software, die ausschließlich das Erstellen, nicht aber das Verarbeiten der Dateninstanzen unterstützt. Ein Beispiel: Für die Angabe der Größeneinheit von eingeleiteten Abwassermengen ist eine Codeliste definiert – Codeliste 4472 – welche aktuell nur einen Eintrag enthält, und zwar den Eintrag Kubikmeter. Wird in einer Software zur Erstellung von Dateninstanzen des vorliegenden Datenformats keine Flexibilität bezüglich dieser Größeneinheit benötigt, dann kann diese Software auch „starr“, ohne Anpassbarkeit und ohne Verwendung einer Codeliste, alle eingeleiteten Wassermengen mit der Größeneinheit Kubikmeter in die XML-Dateninstanz schreiben. Umgekehrt sind im Datenformat aber auch Codelisten in Verwendung, für die in den genannten Software-Anwendungen eine Realisierung als Codeliste geradezu unerlässlich ist, z.B. die Liste der Wasserparameter (Codeliste 2809).

2.1.8 Identifikationszeichenketten und natürlichsprachige Angaben

Identifikationszeichenketten sind für die eindeutige Interpretierbarkeit, die Interoperabilität, sowie die Möglichkeit der Automatisierung von Abfragen und Auswertungen von Daten von großer Bedeutung.

Beispiele für Identifikationszeichenketten:

- Identifikationszeichenketten, z.B. GTINs (Global Trade Item Numbers), die genutzt werden um einen Bezug auf einen Codelisten-Eintrag herzustellen, z.B. um eine Abwasserreinigungsart (etwa Ölabscheider, Filterbecken) zu identifizieren;
- Identifikationszeichenketten, z.B. Firmenbuchnummern, die genutzt werden, um Anlagen, Unternehmen, oder andere „Objekte“ zu identifizieren.

Die Bedeutung von Identifikationszeichenketten am Beispiel der Interoperabilität: Durch die Verwendung einheitlicher Identifikationszeichenketten können dieselben Daten ohne „Übersetzungen“ in verschiedenen Sprachräumen (deutsch, französisch, englisch, usw.) interpretiert und verarbeitet werden. Würden nur natürlichsprachige Bezeichnungen verwendet, so wären Übersetzungen erforderlich.

Identifikationszeichenketten ohne natürlichsprachige Elemente sind für sich alleine für Menschen zumeist nicht interpretierbar, sondern erfordern ein „Nachschlagen“, z.B. die Suche einer Firmenbuchnummer im Firmenbuch oder die Suche einer Telefonnummer im Telefonbuch.

Die im vorliegenden Dokument beschriebenen Datenformate sind mit den folgenden Ansprüchen konzipiert:

1. Die Daten müssen für die automatisierte Verarbeitung und Auswertung gut geeignet sein;
2. Daten sollen soweit vollständig angegeben werden können, dass sie von Menschen unmittelbar interpretierbar sind, dass also gewöhnliche Sprachkenntnisse und Allgemeinbildung genügen, um die Daten interpretieren zu können, und dass kein „Nachschlagen“ in Codelisten oder Registern erforderlich ist.

Eine „technischer“ formulierte Variante dieses Anspruchs: Gemäß XML-Datenformat sollen Daten so angegeben werden können, dass einfache Transformationen (XSLT, XSL-FO) ohne „Lookups“ genügen, um aus diesen XML-Daten beispielsweise PDF-Dateien zu generieren, die mit gewöhnlichen Sprachkenntnissen und Allgemeinbildung ohne weiteres „Nachschlagen“ unmittelbar interpretierbar sind.

Dem wird wie folgt Rechnung getragen:

1. Das Datenformat sieht für die meisten Angaben von Identifikationszeichenketten die zusätzliche Angabe natürlichsprachiger Identifikationselemente, z.B. Namen oder Beschreibungen vor.

Illustriert an einem Beispiel:

```
<TypeID collectionID="2403" objectDesignation="Volumen">9008390104491</TypeID>
```

Im „objectDesignation“-Attribut befindet sich die natürlichsprachige Beschreibung – „Volumen“ – der GTIN 9008390104491 aus Codeliste 2403 (Parameter).

2. Maßgeblich für die Interpretation der Inhalte ist jedenfalls die Identifikationszeichenkette;
3. Dass Identifikationszeichenketten und natürlichsprachige Identifikationselemente zusammenpassen hat durch Softwarehersteller sichergestellt zu werden;
4. Passen Identifikationszeichenketten und natürlichsprachige Identifikationselementen nicht zusammen, ist das ein schwerer inhaltlicher Mangel. Eine (automatische) Zurückweisung der Dateninstanz kann die Folge sein;
5. Bei der Verarbeitung von Daten gilt:
 - a. Natürlichsprachige Identifikationselemente, z.B. Namen oder Beschreibungen, zu Bezügen auf Codelisten-Einträge brauchen nicht gespeichert und weiterverarbeitet zu werden;
 - b. Natürlichsprachige Identifikationselemente, z.B. Namen oder Beschreibungen, zu Bezügen auf Stammdaten-Einträge müssen gespeichert und weiterverarbeitet werden.

2.1.9 XML Schema Design Pattern: Venetian Blind

Es gibt verschiedene sogenannte *Design Patterns* für ein XML Schema. Die gängigsten davon sind unter den Namen *Russian Doll*, *Salami Slice*, *Venetian Blind* und *Garden of Eden* bekannt.

Um den modulartigen, kompakten und weitestgehend redundanzfreien Aufbau aus konzeptionellen Datenmodellen in XML Schema Definitionen übernehmen zu können, werden die Bausteine durchwegs als sogenannte *global types* abgebildet. Das sind benannte und damit wiederverwendbare XML Schema Typdeklarationen. Dieser Ansatz ist genau der *Venetian Blind* XML Schema *Design Pattern*.

Ein Beispiel zur Illustration, was das in der Praxis bedeutet: Eine Adressstruktur braucht im XML Schema nur 1 Mal (als *complex type*) deklariert zu werden, auch dann, wenn die Adressstruktur an mehreren Stellen in der hierarchischen Struktur verwendet wird (z.B. für eine Absender- und eine Empfänger-Adresse).

Für die vorliegende Schnittstellenbeschreibung ergibt sich aus diesem Design Pattern ein konkreter Nutzen: Die Beschreibung kann modulartig erfolgen, d.h. es erfolgt eine Beschreibung der Komponenten (*complex types*) zusammen mit der Information, an welchen Stellen die Komponenten verwendet werden. Auf diese Weise kann auch die Beschreibung von sehr umfassenden Schnittstellen kompakt und weitgehend redundanzfrei erfolgen.

2.2 Varianten der XML Schema Definitionen

2.2.1 XML Schema 1.1 und XML Schema 1.0 Varianten

Die EMREG XML Schema Definitionen sind konform zu XML Schema 1.1. Es werden darin auch Sprachelemente verwendet, die ausschließlich in XML Schema 1.1 zur Verfügung stehen, nicht aber in XML Schema 1.0. Das trifft insbesondere auf „*assert*“-Statements zu, mit welchen in diesen Schema-Definitionen technische Mindestanforderungen definiert werden, z.B. referenzielle Integrität (z.B. ein Bezug auf einen Teilstrom führt weder „ins Leere“, noch auf einen unpassenden Stammdateneintrag wie etwa einen Bescheid) und die Eindeutigkeit von IDs.

Obwohl XML Schema 1.1 [15], [16] im Jahr 2012 veröffentlicht wurde, kommen in der Softwareentwicklung zum Teil noch Werkzeuge und Frameworks zum Einsatz, die lediglich XML Schema 1.0 unterstützen. Um diesem Umstand Rechnung zu tragen, werden zu den EMREG-Datenformaten auch XML Schema 1.0 verträgliche Varianten der Schema-Definitionen im Unterordner `xsd_schema1p0` bereitgestellt. Diese unterscheiden sich von den „regulären“, nur bezüglich XML Schema 1.1 (aber nicht mit 1.0) gültigen EMREG XML Schema Definitionen dadurch, dass Elemente, die in XML Schema 1.0 nicht zur Verfügung stehen, insbesondere sogenannte „*assert*“-Statement, aus den XSD-Dateien entfernt sind. Die XML Schema 1.0 verträglichen Varianten der Schema-Definitionen definieren daher exakt dieselben Datenstrukturen, enthalten aber weniger technische Mindestanforderungen.

Dabei ist wichtig, folgendes zu beachten: Empfänger von EMREG-XML-Dateninstanzen validieren im Allgemeinen gegen die „gewöhnliche“ Variante der EMREG-XSDs, d.h. gegen die XML Schema 1.1 konforme Variante der XSDs mit „*assert*“-Statements. Empfänger prüfen somit automatisch auch die Einhaltung der in diesen XML Schema Definitionen enthaltenen technischen Mindestanforderungen. **Diese per „*assert*“-Statement ausgedrückten Mindestanforderungen müssen also in jedem Softwareprodukt, welches das Schreiben/Erstellen von EMREG-XML-Dateien unterstützt, eingehalten werden, auch dann, wenn für die Entwicklung der betreffenden Software die XML Schema 1.0 verträglichen Varianten der XML Schema Definitionen herangezogen werden, welche diese „*assert*“-Statements nicht enthalten.**

2.3 Schematron-Prüfungen

Für die EMREG-OW XML-Datenformate sind per Schematron Regeln für den XML-Import in die EMREG-OW-Anwendung definiert (siehe Abschnitt 2.1.6 auf Seite 16 für eine Erläuterung von Schematron). Sowohl die Schematron-Datei (Endung: `.sch`) als auch die daraus abgeleitete XSLT-Datei (Endung: `.xslt`) werden bereitgestellt.

Die Verwendung von Schematron-Prüfungen erfolgt vor allem aus diesen Gründen:

1. Ist eine XML-Dateninstanz nicht gültig bezüglich der zum Datenformat gehörenden XML Schema Definition (XSD), so wird die entsprechende Rückmeldung vom XML Validator generiert und ist nicht immer leicht verständlich. Im Gegensatz dazu kann bei Schematron-Prüfungen mit aussagekräftigen Texten auf die Ursachen für einen Prüfprotokolleintrag hingewiesen werden.
2. Die Verletzung von XML Schema Definition (XSD) Vorgaben bedeutet automatisch, dass eine XML-Datei nicht verarbeitbar ist. Im Gegensatz dazu sind für die EMREG-OW-Schematron-Prüfungen verschiedene Klassen festgelegt. Die Klasse „Hinweis“ („warning“) liefert der Benennung entsprechend lediglich Hinweise. So muss die Nicht-Einhaltung von Bedingungen nicht automatisch zur Zurückweisung (Nicht-Verarbeitbarkeit) von XML-Instanzen führen. Bzw. umgekehrt formuliert: Es kann auch bei verarbeitbaren XML-Instanzen auf zu hinterfragende Aspekte der XML-Instanz aufmerksam gemacht werden.
3. Genauso wie die Validierung gegen ein XML Schema, kann auch die Schematron-Validierung leicht in verschiedenster Software mit Standardkomponenten (XSLT Prozessor) implementiert werden. Erstellte XML-Dateninstanzen können bereits vor Übermittlung über eine Schnittstelle geprüft werden.
4. Schematron-Prüfungen sind, im Vergleich zu beispielsweise in Java umgesetzten Prüfungen, leichter anpassbar, korrigierbar und erweiterbar.

Per Schematron sind nur solche Prüfungen sinnvoll realisierbar, die ausschließlich von in der Dateninstanz vorhandenen Daten abhängig sind, aber keine „externen Datenquellen“ benötigen, z.B. Stammdatenregister, Vorjahresmeldung, usw. Bei der Übermittlung einer EMREG-OW XML-Dateninstanz kann es deshalb auch dann zu einer Zurückweisung von Daten kommen, wenn die Dateninstanz gültig bezüglich des Schemas ist und alle Schematron-Regeln erfüllt – schließlich könnte etwa eine solche „externen Datenquellen“ einbeziehende Bedingung verletzt sein.

Zu beachten ist, dass XML-Exporte aus der EMREG-OW-Anwendung nicht notwendigerweise alle per Schematron definierten Regeln erfüllen. Das kann etwa beim Export von „Altdaten“ auftreten, die über Vorgänger-Schnittstellen übermittelt wurden, aber auch bei neueren Daten, wenn beispielsweise bei der Übermittlung Hinweise im Prüfprotokoll ignoriert wurden.

2.4 Zuordnung von Fachbegriffen zu Datenelementen

Fachbegriff	Bezeichnung im Datenformat / Verweis	Anmerkung / alternativer Fachbegriff
Abwasserinhaltsstoff		siehe Parameter
Abwasserstrom	<i>WasteWaterStream</i>	
Altlast	<i>Installation</i>	
Anlage	<i>Installation</i>	
Deponie	<i>Installation</i>	
Fracht	Ab- und Zulauffrachten als <i>InfluentMassFlowEvent</i> und <i>EffluentMassFlowEvent</i> in <i>EnvironmentalData</i>	
Parameter (Abwasserinhaltsstoff)	<i>PropertyKindID</i> in <i>ReferencePeriodParameterProperty</i> und <i>PropertyAnalysisEvent</i> zur Angabe von Jahresfrachten und Einzelmesswerten.	
Teilstrom	siehe Abwasserstrom	

3 VORGABEN AN SOFTWARE

3.1 Allgemeines

Zur Datenformatspezifikation zählen auch die im Folgenden aufgelisteten Vorgaben an Software, für die eine Unterstützung des Datenformats implementiert wird (Erstellen von XML-Daten dieses Formats, oder/und Verarbeiten von XML-Daten dieses Formats). Die Zielsetzung hinter diesen Vorgaben ist ein friktionsfreies, sicheres und für Anwender gut benutzbares Zusammenspiel von verschiedenen Software-Produkten.

3.2 Vorgaben in Bezug auf Erstellung und Verarbeitung von Dateninstanzen

Vorgabe 1 (ID 341): Generierte Dateninstanzen MÜSSEN bezüglich der veröffentlichten XML Schema Definition gültig sein.

Anmerkung: Es darf insbesondere nicht möglich sein, dass Nutzer der Software durch ihre Interaktion mit der Software (z.B. Eingabe unsinniger Daten oder Weglassen erforderlicher Daten) das Generieren ungültiger Dateninstanzen auslösen können.

Beispiel: Eine Software bietet in der Benutzeroberfläche ein Formular zur Erfassung von Jahresfrachten, sowie die Möglichkeit des Erstellens bzw. Übermittels einer EmRegV-OW-XML-Dateninstanz. Ein Software-Benutzer hat einen Parameter (z.B. eine Fracht) ausgewählt, dazu aber noch keinen Wert angegeben, und wählt nun die Funktion „XML-Export/Übermittlung“. Gemäß XML-Datenformat wird für den Parameter auch der Wert benötigt. Würde die Software eine XML-Dateninstanz erstellen, in welcher der Wert fehlt, so handelte es sich um eine ungültige Dateninstanz, und die Software wäre mangelhaft. Es ist also Aufgabe der Software, den Benutzer darauf aufmerksam zu machen, dass die vorliegenden Angaben unzureichend sind um eine gültige Dateninstanz zu erzeugen. ■

Vorgabe 2 (ID 628): Bei der Zeichencodierung generierter Dateninstanzen MUSS es sich um UTF-8 handeln. ■

Vorgabe 3 (ID 705): Sofern ergänzend zum Datenformat Datenanforderungen spezifiziert sind: Generierte Dateninstanzen MÜSSEN allen der Datenanforderungen entsprechen, die als verpflichtend gekennzeichnet sind. ■

Vorgabe 4 (ID 394): Sofern ergänzend zum Datenformat Datenanforderungen spezifiziert sind: Software SOLL so implementiert sein, dass generierte Dateninstanzen ALLEN veröffentlichten Datenanforderungen entsprechen.

Anmerkung: Diese Vorgabe impliziert unter anderem das folgende Verhalten von Software: Sind von Benutzern stammende Angaben auf eine Weise unvollständig oder inkonsistent, die das Erstellen einer allen Datenanforderungen genügenden Dateninstanz verhindert, dann soll durch die Software keine Dateninstanz erstellt oder übermittelt werden, sondern stattdessen der Software-Benutzer auf das Fehlen oder die Inkonsistenz von Daten aufmerksam gemacht werden. ■

Vorgabe 5 (ID 816): Bei der Verarbeitung von Dateninstanzen MÜSSEN Dateninstanzen, welche sämtliche der als verpflichtend gekennzeichneten Datenvorgaben – insbesondere Gültigkeit bezüglich des XML Schemas und, sofern Datenanforderungen spezifiziert sind, Einhaltung aller als verpflichtend gekennzeichneten Datenanforderungen – erfüllen, akzeptiert werden und dürfen nicht automatisch zurückgewiesen werden.

Anmerkung: Eine bei der Verarbeitung akzeptierte Dateninstanz kann in Folge von einem Menschen inhaltlich nicht akzeptiert bzw. zurückgewiesen werden. Diese Vorgabe bezieht sich lediglich auf ein automatisches Zurückweisen, welches es ausschließlich unter den genannten Voraussetzungen geben darf. ■

Vorgabe 6 (ID 549): Soll ein Softwareprodukt das Entgegennehmen und (ggf. spätere) „Weiterleiten“ von zuvor empfangenen EMREG-XML-Dateninstanze ermöglichen, dann MUSS die Speicherung in Bezug auf die Inhalte (XML-Element- und Attributwerte) abwandlungsfrei und verlustfrei sein. Datenabwandlungen und Verluste MÜSSEN für die gesamte Dauer der Persistierung ausgeschlossen sein.

Anmerkungen:

- Dateninstanzen MÜSSEN so persistiert werden, dass die persistierten Daten für das Erstellen einer XML-Instanz geeignet sind, die sich von der entgegengenommenen XML-Instanz in den Inhalten (XML-Element- und Attribut-Inhalte) nicht unterscheidet.

- Dateninstanzen brauchen NICHT so persitiert zu werden, dass eine exakte Reproduktion der entgegengenommenen XML-Instanz grundsätzlich möglich ist. Unterschiede zwischen entgegengenommener XML-Instanz und aus persistierten Daten generierter bzw. generierbarer XML-Instanz sind zulässig, sofern sie nicht die Inhalte betreffen. Beispiele für solche nicht die Inhalte betreffenden Unterschiede zwischen XML-Instanzen sind in der W3C Empfehlung „Canonical XML“ beschrieben. Werden etwa in einer XML-Instanz Tabulatorzeichen zur Einrückung von XML-Tags verwendet, und in der anderen stattdessen Leerzeichen, so sind die beiden XML-Instanzen zwar als Byte- oder Zeichenfolge nicht exakt übereinstimmend, inhaltlich aber dennoch äquivalent.
- De facto bedeutet diese Vorgabe auch, dass entgegengenommene und verarbeitete XML-Element- und Attributwerte allesamt einzeln für sich persistiert werden müssen. Es ist insbesondere die folgende Arten der Persistierung nicht geeignet:
 - Entgegengenommene Element- oder Attributwerte stimmen mit Werten aus Stammdaten überein, auf welche die Daten entgegennehmende Software Zugriff hat. Anstelle die Element- und Attributwerte einzeln für sich zu persistieren wird lediglich ein Verweis auf den Stammdateneintrag gespeichert;

Diese Arten der Persistierung sind aus dem folgenden Grund nicht geeignet: Für Anpassungen von Stammdaten soll unabhängig von deren Historisierung jedenfalls sichergestellt sein, dass deren Anpassungen keine automatischen (und im Allgemeinen unbeabsichtigten) Änderungen von Dateninstanzinhalten nach sich ziehen. ■

3.3 Vorgaben in Bezug auf den Umgang mit Codelisten

Vorgabe 7 (ID 216): Auslegung auf Codelisten-Änderungen

Software MUSS so implementiert werden, dass eine Aktualisierung von Codelisten bzw. ein Verwenden der Software mit aktualisierten Codelisten ohne neues Kompilieren, Ausrollen und Installieren der Software möglich ist. ■

Vorgabe 8 (ID 481): Verwendung des EDM-Codelisten-Webservice

Im EDM werden über ein Webservice Codelisten zum Abruf angeboten. Software DARF NICHT so implementiert werden, dass jeder Zugriff auf Codelisten ad hoc und unmittelbar über das EDM Webservice erfolgt. Stattdessen MUSS Software mit „lokalen Kopien“ der Codelisten arbeiten. Das EDM Webservice zum Bezug von Codelisten DARF NICHT für andere Zwecke verwendet werden als das Initialisieren und Aktualisieren solcher „lokaler Codelisten-Kopien“. ■

Vorgabe 9 (ID 634): Automatische Aktualisierung von Codelisten

Es wird EMPFOHLEN, Software so zu implementieren, dass die Verfügbarkeit aktualisierter Codelisten in regelmäßigen Abständen automatisch geprüft wird.

Anmerkung: Eine solche Überprüfung der Verfügbarkeit aktualisierter Codelisten ist durch Implementierung einer Anbindung an das EDM Codelisten-Webservice möglich. ■

Vorgabe 10 (ID 788): Mindest-Häufigkeit für die Prüfung auf Codelisten-Aktualität

Wird von Software in regelmäßigen Abständen automatisiert die Verfügbarkeit aktualisierter Codelisten geprüft, dann SOLL die Prüfung auf die Verfügbarkeit aktualisierter Codelisten zumindest alle 30 Tage erfolgen. ■

Vorgabe 11 (ID 580): Maximale Häufigkeit für die Prüfung auf Codelisten-Aktualität

Wird von Software in regelmäßigen Abständen automatisiert die Verfügbarkeit aktualisierter Codelisten unter Verwendung des EDM Webservice für Codelisten geprüft, dann DARF die Prüfung auf die Verfügbarkeit einer aktualisierten Liste nicht öfter als einmal alle 12 Stunden erfolgen, und SOLL nicht öfter als einmal alle 24 Stunden erfolgen. ■

Vorgabe 12 (ID 909): Voraussetzung für die Zurückweisung von Dateninstanzen aufgrund „unbekannter“ Bezüge auf Codelisten-Einträge

Wenn bei der Entgegennahme bzw. Verarbeitung von Daten geprüft wird, ob in den entgegengenommenen Daten enthaltene Identifikationszeichenketten gültig in dem Sinn sind, dass sie mit der zu einem Codelisten-Eintrag gehörigen Identifikationszeichenkette übereinstimmen, dann MUSS folgende Bedingung eingehalten werden: Eine automatisierte Zurückweisung darf nur dann erfolgen, wenn sichergestellt ist, dass die von der empfangenden Software genutzten Codelisten-Kopien mindestens so aktuell sind wie die vom Dokumentersteller bzw. der dokumenterstellenden Software genutzten Codelistenkopien. ■

Vorgabe 13 (ID 327): Zurückweisung von Dateninstanzen auf Basis von Codelisteneintrags-Gültigkeitszeiträumen

Für Datenelemente und dazugehörige Codelisten, für die nichts Anderes definiert ist, gilt Folgendes: Die Gültigkeit eines in der Dateninstanz referenzierten Codelisteneintrags in Bezug auf den in der Codeliste definierten Gültigkeitszeitraum DARF bei der Entgegennahme bzw. Verarbeitung einer Dateninstanz NICHT so geprüft werden, dass eine Verletzung zu einer automatischen Zurückweisung führt.

Begründung: In Bezug auf die Prüfung der zeitlichen Gültigkeit eines Codelisteneintrags ist das Heranziehen eines richtigen Bezugszeitpunkts bzw. Bezugszeitraums wichtig. Als abstraktes Beispiel: Für die Angabe eines Geburtsorts/Geburtsstaats ist das Geburtsdatum der passende Bezugszeitraum. D.h. es sind genau solche Orte/Staaten zulässig, die es zum Zeitpunkt der Geburt gegeben hat. Beispielsweise sind Staaten zulässig, die es zum Zeitpunkt der Geburt gegeben hat, später aber nicht mehr (z.B. DDR). Der passende Bezugszeitpunkt bzw. Bezugszeitraum kann von Datenelement zu Datenelement unterschiedlich sein. Den passenden Bezugszeitpunkt bzw. Bezugszeitraum für automatische Prüfungen heranzuziehen, kann daher eine vergleichsweise komplexe Umsetzung erfordern, mit einem vergleichsweise hohen Fehlerrisiko, und somit einem vergleichsweise hohen Risiko, aufgrund nicht gerechtfertigter automatischer Zurückweisungen die Interoperabilität zu beeinträchtigen. Aus diesem Grund sollen Dateninstanzen auch bei (vermeintlicher) Ungültigkeit eines Codelisteneintrags bzgl. eines Gültigkeitsraums technisch akzeptiert werden. Automatische Prüfungen sind dennoch möglich, solange sie keine automatischen technischen Zurückweisungen zur Folge haben (sondern z.B. nur Hinweise an Benutzer). ■