XZuFi Validator Handbuch

verantwortlich:	Denis Degraf
Handbuch-Version:	2.11
Datum letzte Änderung:	30.09.2025
Status:	Abgeschlossen
Schutzstufe:	keine Schutzstufe

Änderungshistorie

In Tabelle 1 sind die letzten Versionen des Dokuments und die jeweils durchgeführten Änderungen aufgeführt.

Handbuch-Version	Datum	Author	Validator Release	Anpassungen
2.11	30.09.2025	Denis Degraf	./.	Validator VAL0050 auf Hinweis-Modus umgestellt.
2.10	15.09.2025	Denis Degraf	./.	Handbuch um Validator VAL0057 ergänzt.
2.9	29.08.2025	Norman Lierhaus	./.	Dokument um Validatorbegründungen und Inhaltsverzeichnis ergänzt.
2.8	26.08.2025	Denis Degraf		Validator VAL0036 angepasst. Volltextmodule müssen nicht an der Leistung übertragen werden und können durch eine Spezialisierung gesetzt werden.
2.7	15.08.2025	Niklas Vieland		Validatorenmodi aktualisiert.
2.6	05.08.2025	Jochen Hergeröder		Redaktionssystem-Konfiguration wird vom Sammlerdienst abgerufen.
2.5	04.08.2025	Christian Glazik	Validator-Lib 12.5.0	Anpassung der Validatoren Validator VAL0036 und VAL0065.
2.4	08.07.2025	Tiaden Beyl		Anpassung der XZuFi Versionen.
2.3	10.06.2025	Niklas Vieland		Validator VAL0065 hinzugefügt.
2.2	20.05.2025	Denis Degraf, Niklas Vieland		Validator VAL0030 wurde angepasst.

2.1	21.03.2025	Simone Höppner-Welcher		Leistungsbeschreibung in Leistung umbenannt
2.0	14.01.2025	Thomas Kreiß	Validator-Lib 11.0.0 Frontend 5.8.x	Dokument auf XZuFi 2.3 angepasst.
1.8	09.10.2024	Stephan Timm	Validator Frontend 5.8.x	Beschreibung des Webfrontend aktualisiert.
1.7	30.07.2024	Niklas Vieland	Validator-Lib 9.2.x	Neuer Validator VAL0064 und Aktualisierung der Beschreibung von Validator VAL0047.
1.6	10.01.2024	Thomas Kreiß	Validator-Lib 6.8.x	Neue Validatoren VAL0062 und VAL0063, sowie Einführen des Hinweis-Modus.
1.5	29.11.2023	Niklas Vieland	Validator-Lib 6.6.x	Protokollmodi von VAL0020, VAL0042 und VAL0043 auf die aktuelle Standardkonfiguration gesetzt. Versionsnummern der Anwendungen aktualisiert.
1.4	04.07.2023	Niklas Vieland	Validator-Lib 6.1.x	Ausnahmeregeln für VAL0012 und VAL0016 für Leistungen im gemeinsamen Vollzug hinzugefügt.
1.3	02.06.2023	Niklas Vieland	Validator-Lib 6.0.x	Der Validator VAL0060 wurde hinzugefügt.
1.2	25.05.2023	Denis Degraf, Niklas Vieland	Validator-Lib 5.6.x	Die Validatoren VAL0056 und VAL0059 wurden hinzugefügt.
1.1	15.02.2023	Denis Degraf, Niklas Vieland	Validator-Lib 5.2.x	Der Formularvalidator VAL0047 wurde angepasst. Die Felder 'Beschreibung', 'Kurzbezeichnung' und 'Hinweis' werden nicht mehr geprüft.
1.0	27.09.2022	Denis Degraf, Manfred Nentwig	Validator-Lib 3.2.x	Das Validator Handbuch wurde überarbeitet. Die einzelnen Validatoren haben jetzt eine eindeutige Nummer zugeordnet bekommen. Das Handbuch wurde mit den Beispielen ergänzt und neu strukturiert.

Chapter 1. Einleitung ins PVOG Validatorhandbuch

1.1. Kontext

Im Rahmen des Projekts Portalverbund Online Gateway (PVOG) werden die Verwaltungsleistungen des Bundes und teilnehmender Länder über mit dem PVOG verknüpfte Verwaltungsportale verfügbar gemacht. Dazu werden diese Verwaltungsleistungen von Bund und Ländern im XZuFi-Format zur Verfügung gestellt. Neben dem XML-Schema gemäß XZuFi-Version 2.2 und 2.3.1 stellt das PVOG noch zusätzliche, die fachliche Logik betreffende Anforderungen an die bereitgestellten XZuFi Dokumente. Die Überprüfung der fachlichen Anforderungen wird nicht von einer technischen Schema-Validierung der Dokumente abgedeckt und muss separat durchgeführt werden. Im Sammlerdienst des PVOG wird diese umfängliche Validierung mithilfe einer Validator Komponente umgesetzt. Um den Redaktionssystemen die Lieferung technisch und fachlich korrekter Verwaltungsleistungsdaten zu vereinfachen, wird derselbe Validator nun auch als separater Dienst zur Verfügung gestellt. Dabei entspricht der Validierungsumfang mindestens dem des im PVOG eingesetzten Sammlerdienstes. Das Validierungsergebnis wird verständlich formuliert in Form eines Berichts ausgegeben.

1.2. Zweck des Dokuments

Das vorliegende Dokument dokumentiert die eingesetzten Validatoren im einzelnen und beschreibt die Anwendung der Validator-Dienst Benutzungsoberfläche.

Chapter 2. Bedienung und Schnittstellen

2.1. REST API

Eine Dokumentation der REST-API des Validator-Backends ist als Open-API-Specification in einem separaten Dokument "REST API des Validator Backends" bereitgestellt.

2.2. Webfrontend

Das PVOG stellt ein Webfrontend als Benutzungsoberfläche für den Validator-Dienst zur Verfügung, welches mit Firefox und Chrome Browsern getestet wurde. Für die Nutzung des Webfrontend und der REST API ist ein Anwenderkonto erforderlich. Die Zugangsdaten können per Mail an dataportpvogsupport@dataport.de vom Support des PVOG erfragt werden. Im Konto hinterlegt sind unter anderem auch die Berechtigungen für den Zugriff auf die Daten einzelner Redaktionssysteme. So kann der Kontoinhaber z.B. auf die Daten des Redaktionssystems Hamburg zugreifen, aber nicht auf die Daten des Redaktionssystems Hessen.

Zur Anmeldung am Webfrontend muss ein Anwender seinen Kontonamen und sein Passwort angeben. Einmal angemeldet, gibt es im Webfrontend eine Schaltfläche zum Abmelden des Anwenders. Da die Sitzung nach kurzer Inaktivität automatisch beendet wird, ist eine manuelle Abmeldung in der Regel nicht notwendig. Sollte ein Anwender aber über mehrere Konten verfügen, kann er sich so von der aktuellen Sitzung ab- und dann mit einem anderen Konto wieder anmelden.

Nach dem Anmelden präsentiert das Webfrontend zuerst den Reiter zur Validation von Datenbeständen, die der Anwender bereitstellen muss.



Abbildung 1: Validator Frontend

2.2.1. Reiter Validation

In diesem Reiter können Datenbestände validiert werden, die unter einer zu spezifizierenden URL zur Verfügung gestellt wurden. Der Validator-Dienst wird diesen Datenbestand dann herunterladen und validieren und stellt dem Anwender dann das Prüfergebnis und einen Validationsreport zur Verfügung.

2.2.1.1. URL-Eingabe und Authentifizierung

Die zu überprüfenden Verwaltungsleistungen und Organisationseinheiten werden in der im PVOG etablierten Form zur Verfügung gestellt. Eine Index-Datei (z.B. im .txt-Format) verweist zeilenweise durch relative oder absolute Pfadangaben auf die zu validierenden Dokumente. Die URL zur Indexdatei kann dann in das dafür vorgesehene Textfeld eingetragen werden. Unterstützte Protokolle sind http und https.

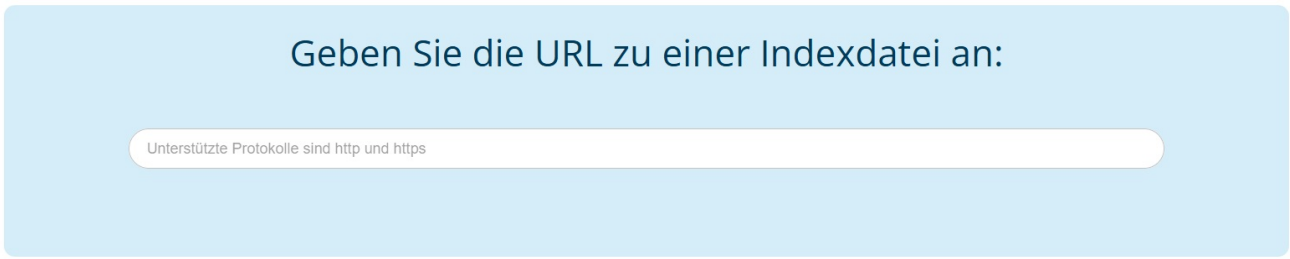


Abbildung 2: Die URL zu einer Indexdatei

Ist die Indexdatei passwortgeschützt, können Benutzername und Passwort zur Entschlüsselung dieser Datei im Bereich Authentifizierung angegeben werden. Dabei handelt es sich in der Regel nicht um die Kontoinformationen, mit denen sich der aktuelle Anwender am Webfrontend angemeldet hat.



Abbildung 3: Authentifizierung

2.2.1.2. Auswahl des Bundeslandes/Redaktionssystems

Die Zuständigkeiten von Organisationseinheiten für Verwaltungsleistungen sind immer an ein oder mehrere anzugebende Gebiete gebunden, die durch die Angabe der entsprechenden amtlichen Regionalschlüssel (ARS) festgelegt werden. Der Validator überprüft diese auf Konformität zum jeweils zuständigen Bundesland bzw. Redaktionssystem. Nach Beschluss des IT-Planungsrates wird diese Zuständigkeit auf der Länderebene geklärt (die zwei führenden Ziffern des ARS). Daher ist vor dem Start der Validierung das verantwortliche Bundesland bzw. Redaktionssystem aus einer Liste auszuwählen.

Unter 'Redaktionssystem spezifische Konfiguration' wird zu dem ausgewählten Bundesland bzw. Redaktionssystem das Gebiet (2-stelliger Länder-ARS bzw. bei einigen Redaktionssystemen auch mehrere ARS), die Redaktionssystem-ID (schemeAgencyId) sowie die erlaubten Leika Typen angezeigt. Diese Daten stammen vom Sammlerdienst und werden vom Validator automatisch abgerufen. Sollte der Sammlerdienst nicht erreichbar sein, wird eine Standard-Liste der Redaktionssysteme angezeigt, die keine Einschränkungen bei den Leika-Typen hat.



Abbildung 4: Redaktionssystem

2.2.1.3. Validierung abbrechen

Sobald die Validierung gestartet wurde, besteht die Möglichkeit mit dem Button 'Validierung abbrechen' den Validierungsprozess abzubrechen.

2.2.1.4. Fortschritt und Ergebnis der Validierung

Unterhalb des Eingabeformulars wird der aktuelle Validierungsfortschritt angezeigt. Bei abgeschlossener Validierung erscheint dort entweder eine Erfolgsmeldung

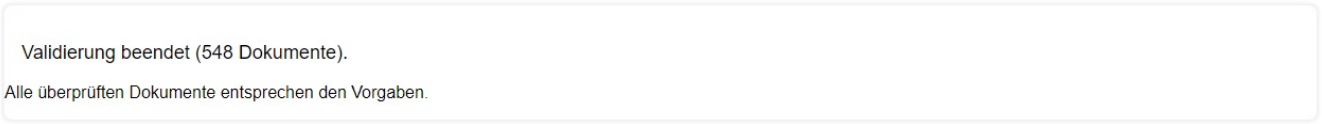


Abbildung 5: Erfolgsmeldung

oder eine Liste der gefundenen Validierungsfehler.

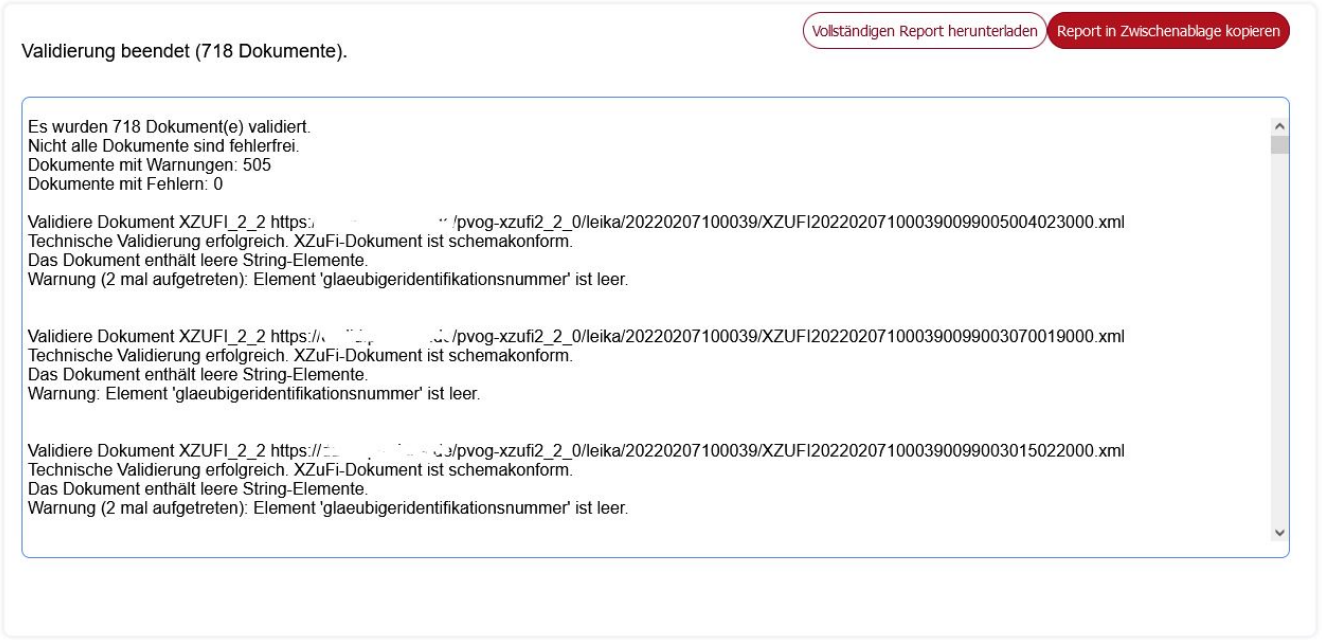


Abbildung 6: Validierungsfehler

Über die Schaltflächen am oberen rechten Rand kann der Report entweder in die Zwischenablage kopiert oder, falls der Report zu groß ist, als Text-Datei heruntergeladen werden.

Falls der Virenscanner in der Index-Datei einen Virus entdeckt, werden keine weiteren Dateien geprüft und die Validierung sofort abgebrochen. Bei einem Virusfund in einem XZuFi-Dokument, wird dies als Fehler dokumentiert, die Validierung dieses Dokuments abgebrochen und mit dem nächsten Dokument fortgefahren. Alle Virenfunde sind mit entsprechender Fehlermeldung in dem abgeschlossenen Validierungsreport aufgeführt.

2.2.2. Reiter Info: Anzeige der Fachvalidatorkonfiguration

Im Reiter Info sind alle Prüfregeln des Validator-Dienstes aufgelistet. Neben der Bezeichnung, einer Beschreibung und dem Datum des Inkrafttretens einer Prüfregel wird angegeben, ob sie aktiv ist und in welchem Modus die Regel läuft. Die Einhaltung von inaktiven Regeln wird bei der

Validierung des Datenbestandes nicht überprüft. Der Modus bestimmt, ob Verletzungen der Regel als Fehler, Warnung oder Hinweis gemeldet werden. Die Modi Warnung und Hinweis dienen dazu, die fachliche Qualität des Datenbestandes zu verbessern. Verletzungen einer Regel im Modus Fehler, die vom Validator-Dienst nur entsprechend protokolliert werden, würden im Sammlerdienst zu einer Ablehnung des Imports solcher Daten führen.

Zusätzlich finden sich im Reiter Info die unterstützten XZuFi-Versionen sowie die Versionen der aktuell aktiven Fachvalidator-Bibliothek und des Validator-Dienstes.

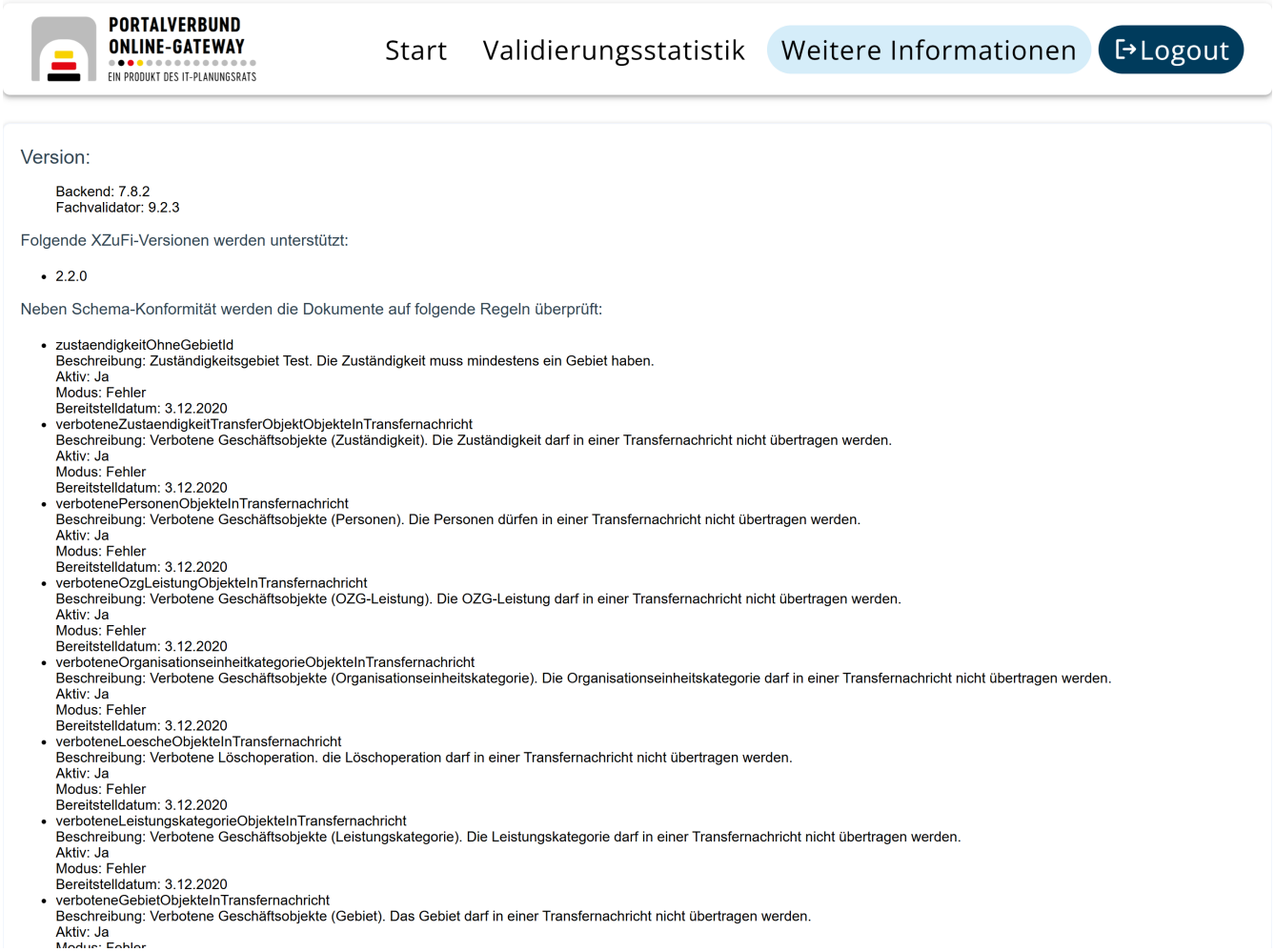


Abbildung 7: Anzeige der Validatorkonfiguration

2.2.3. Reiter Statistiken

Im Reiter Statistiken werden die Ergebnisse der täglichen Gesamtbestandsvalidierung angezeigt. Bei der Umstellung auf den Push-Betrieb war es nicht mehr möglich, die Validation jeder Push-Nachricht im gleichen Umfang durchzuführen wie beim bisher genutzten Pull-Betrieb oder wie bei der manuellen Validation aller Daten eines Redaktionssystems hier im Validator-Dienst. Deshalb werden im PVOG-System täglich die Datenbestände jedes Redaktionssystems automatisch validiert und die Ergebnisse hier dargestellt.

Hier kann ein Anwender einen statistischen Überblick über die Ergebnisse der Gesamtbestandsvalidierung erhalten sowie pro Redaktionssystem den letzten Report herunterladen. Dabei hat ein Anwender nur Zugriff auf die Daten derjenigen Redaktionssysteme, für die in seinem Konto die notwendigen Berechtigungen hinterlegt sind.

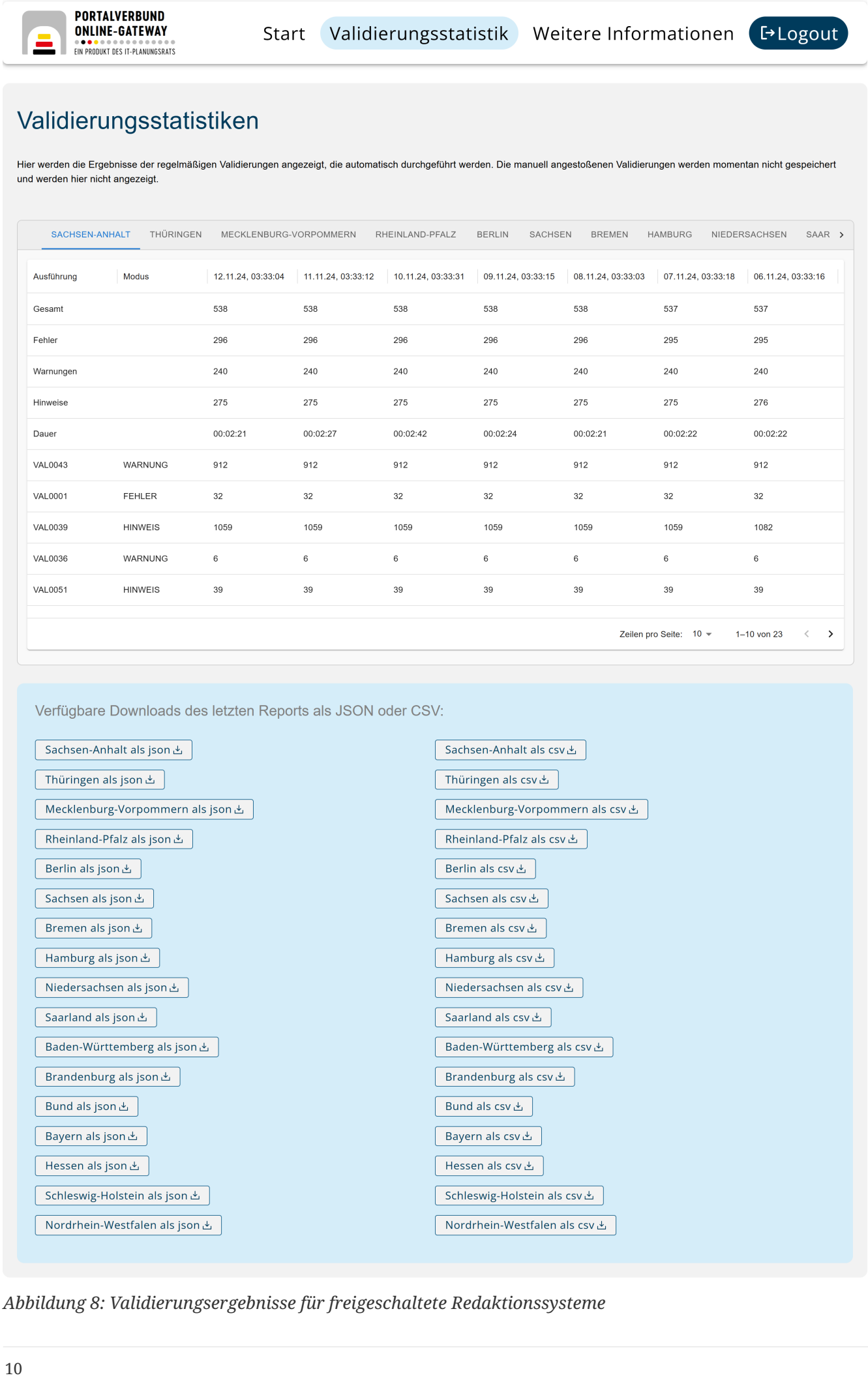


Abbildung 8: Validierungsergebnisse für freigeschaltete Redaktionssysteme

Chapter 3. Validierungsumfang XZuFi

Die Software führt sowohl eine technische als auch eine fachliche Validierung durch. Inhalte werden nicht validiert.

3.1. Technische Validierung

Die technischen Validierung prüft das Dokument auf Schema-Konformität zum XZuFi-Schema Version 2.2 und Version 2.3.1.
In der Übersicht werden sie markiert durch die Kürzel T (technisch notwendige Validierung) und ST (Standard-Konformität ohne unmittelbare technische Notwendigkeit). Validatoren der technischen Validierung sind für die Funktion des PVOG unerlässlich.

3.2. Fachliche Validierung

Die Fachliche Validierung überprüft das Dokument auf die im Folgenden aufgeführten Regeln und wird nur nach einer erfolgreichen technischen Validierung durchgeführt. Die meisten dieser Validierungsregeln sind mit DQ (Datenqualität) markiert, sie geben Aufschluss darüber, ob die gelieferten Daten den vereinbarten Regeln entsprechen oder sie verletzen. Die Grundlage für diese Regeln bildet das Sammlerdienst-Konzept. Neue Validatoren können gemeinschaftlich durch beschlussgebende Gremien (KG Portalverbund, UAG Leistungen) vorgeschlagen und durch die Fitko beauftragt werden. Grundsätzlich gelten für die bereitgestellten Dokumente die gleichen Anforderungen wie im PVOG. Das heißt, die Dokumente werden einzeln überprüft und müssen in sich konsistent sein. Zuständigkeiten dürfen nur auf die in dem XZuFi-Dokument gelieferte Leistung verweisen.

Als Ergebnis einer nicht fehlerfreien Validierung, werden die Fehler in die drei Schweregrade 'Error', 'Warnung' und 'Hinweis' kategorisiert und resultieren aus den konfigurierten Modus der Validatoren.

- **Error:** Es ist fachlicher Fehler aufgetreten, der eine Verarbeitung im Sammlerdienst verhindert.
- **Warnung:** Es ist fachlicher Fehler aufgetreten, der eine Verarbeitung im Sammlerdienst aktuell nicht verhindert, aber zukünftig verhindern wird. Der Fehler sollte kurzfristig behoben werden.
- **Hinweis:** Es wurde bei der Validierung ein fachlicher Mangel festgestellt. Eine Prüfung wird angeraten, eine Behebung ist aber nicht zwingend notwendig. Dieser Modus dient lediglich der Information.

3.3. Validator Modus Wechsel

Wie in dem Release-Konzept Version 2.1.0 vom 24.5.2023, Kapt. 3.2, dargelegt, ist die Umstellung der Validatoren von dem Modus 'Warnung' in den Modus 'Error' nach einem Übergangszeitraum von sechs Monaten nach Veröffentlichung des neuen Validators bzw. neuer Regeln im Validator auf allen Umgebungen vorgesehen. Nach der Umstellung ist eine weitere Verarbeitung der Daten nur noch dann möglich, wenn die bisherigen Warnungen behoben wurden. Die Umstellung erfolgt immer nach finaler Absprache mit den Vertretern der Länder und der Redaktionssysteme.

Bei Validatoren, die im Modus 'Hinweis' betrieben werden, erfolgt perspektivisch kein Wechsel in einen der anderen beiden Modi.

3.4. Übersicht der Validierungsregeln

Die Tabelle enthält eine Übersicht aller vorhandenen Validierungsregeln. Die Spalte **Validator-Id** enthält eine eindeutige Nummer der Validierungsregel. Über diese Nummer kann eine Fehler-, Warnungs- oder Hinweismeldung in einem Validierungsprotokoll einer Validierungsregel zugeordnet und somit die Fehler schneller identifiziert werden.
Die **Kurzbeschreibung** enthält Name sowie eine kurze Beschreibung der Validierungsregel. Eine ausführliche Beschreibung samt Beispiele ist in der Kapitel Beschreibung der Validierungsregeln enthalten.
Das Datum der ersten Veröffentlichung und das Datum der geplanter Umstellung in *Fehler*-Modus sind in den Spalten **Veröffentlicht** und **Geplante Umstellung** enthalten. Das Datum der Umstellung ist standardmäßig auf 6 Monate nach dem Datum der Veröffentlichung eingestellt. Das Umstellungsdatum kann nach Bedarf verschoben werden. Ist die geplante Umstellung mit keinem Datum versehen, so wird Validator über eine unbegrenzte Zeit im *Warnung*-Modus betrieben und dient ausschließlich zur Verbesserung der Datenqualität.
Der "Typ" zeigt an, ob es sich um einen Validator für die Datenqualität (DQ) handelt, oder ob er aus Erfordernissen der externen, verbindlichen Standards erwächst (ST) wie z.B. XZuFi-Standard oder Leika+. (T) markiert solche Validatoren, die für die Funktionsweise des PVOG technisch notwendig sind. Der Zusatz (RI) steht für "referenzielle Integrität" und bedeutet, dass die jeweilige Validierungsregel notwendig ist, damit die Daten mit einander verknüpft werden können, d.h. dass lose Daten, sogenannte "Karteileichen" vermieden werden.
Der **Modus** zeigt, ob der Validator im Sammlerdienst im *Fehler*- oder im *Warnung*-Modus betrieben wird. Alle Daten, welche Validierungsregeln im *Fehler*-Modus nicht erfüllen, werden vom Sammlerdienst abgelehnt.

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0065	Eindeutigkeit des Zugangskanals. In XZuFi-2.3.1 dürfen Leistungsmodule gleichen Typs nur mehrfach vorkommen, wenn sie sich in den Zugangskanälen unterscheiden. Gleiche Zugangskanal-Codes oder eine Kombination aus 'Online/Offline = 000' mit 'Online = 100' oder 'Offline = 200' ist nicht erlaubt. Nicht geprüft werden modulText, modulTextIndividuell, modulUrsprungsportal und modulBegriffeImKontext.	26.06.2025	31.12.2025	DQ	Warnung
VAL0064	Ein Formular in einer Organisationseinheit muss mindestens eine Zuständigkeit enthalten.	29.07.2024	10.02.2025	DQ (RI)	Warnung

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0063	Eine Leistung enthält nicht alle Typisierungen des Leistungsschlüssels, die vom FIM-Leistungskatalog vorgegeben sind.	--.--.----	--.--.----	DQ	Hinweis
VAL0062	Eine Leistung enthält Typisierungen des Leistungsschlüssels, die nicht vom FIM-Leistungskatalog vorgegeben sind.	--.--.----	--.--.----	DQ	Warnung
VAL0060	Eine Leistung darf keine Kategorie mit der Bezeichnung 'Mustertext' enthalten.	--.--.----	--.--.----	DQ	Fehler
VAL0059	Länderbezogene Leistungsschlüssel müssen zum Handbuch LeiKa-plus konform sein.	--.--.----	--.--.----	ST, DQ	Warnung
VAL0057	Zwei Leistungen dürfen nicht den gleichen Leistungsschlüssel referenzieren.	--.--.----	--.--.----	DQ	Hinweis
VAL0056	Leistungsschlüssel aus dem Element *'xzufi:referenzLeika'* müssen im *'FIM-Leistungskatalog'* existieren.	--.--.----	--.--.----	DQ	Hinweis
VAL0055	Onlinedienste müssen eine ID, sowie eine schemeAgencyID in gültiger Form enthalten.	03.12.2020	03.06.2021	T	Fehler
VAL0054	ARS-Schlüssel Test. Der ARS-Schlüssel muss in der Liste *'urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs'* vorhanden sein.	--.--.----	--.--.----	DQ	Warnung
VAL0053	Unerlaubte HTML-Tags. Das XZuFi-Dokument darf nur zugelassene HTML-Tags und nur an den zugelassenen Elementen enthalten.	--.--.----	--.--.----	ST, DQ	Warnung
VAL0052	Prüft die Codeliste im XRepository. Die Codeliste muss im XRepository existieren und der Code muss in der Liste vorhanden sein.	01.07.2022	01.01.2023	ST, T	Warnung
VAL0051	PV-Lagen Test. Prüfen der PV-Lagen Zuordnung gegen den FIM-Leistungskatalog.	01.07.2022	--.--.----	DQ	Hinweis
VAL0050	Zuordnung des Leistungsschlüssels. Die Leistung darf maximal einen Leistungsschlüssel referenzieren.	04.04.2022	--.--.----	DQ	Hinweis
VAL0049	Doppelte Zuständigkeiten im Formular. Ein Formular darf keine zwei identische Zuständigkeiten enthalten.	24.03.2022	24.09.2022	T	Fehler

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0048	Leistungsreferenz am Formular. Eine im Formular referenzierte Leistung muss existieren.	23.03.2022	23.09.2022	T (RI)	Fehler
VAL0047	Pflichtattribute am Formular. Ein Formular muss alle erforderlichen Attribute enthalten.	22.03.2022	22.09.2022	DQ	Fehler
VAL0046	Leistungsschlüssel Test. Der Leistungsschlüssel einer Leistung ist kein FIM-Schlüssel.	15.03.2022	15.09.2022	DQ	Warnung
VAL0045	TMP-Leistungsschlüssel Test. Der Leistungsschlüssel ist temporär.	15.03.2022	15.09.2022	DQ	Warnung
VAL0044	Rolle der Organisationseinheit. Die Rolle einer Organisationseinheit muss gesetzt sein.	28.01.2022	28.07.2022	DQ	Fehler
VAL0043	Vollständigkeit der Sprachversion. Alle übertragenen Sprachen müssen durch Sprachversion deklariert sein.	07.01.2022	07.07.2022	DQ	Warnung
VAL0042	Geo-Koordinaten Test. Die Geo-Koordinaten müssen im EPSG-Standard übertragen werden und müssen in Deutschland liegen.	01.09.2021	01.03.2022	DQ	Hinweis
VAL0041	Sprache des Moduls. Die Sprache des Modulinhaltes muss gekennzeichnet werden.	21.05.2021	21.11.2021	ST, DQ	Fehler
VAL0040	RedaktionssystemId Test. Die RedaktionssystemId muss mit den Daten mitübertragen werden.	21.05.2021	21.11.2021	T	Fehler
VAL0039	Leerer Inhalt Test. Ein XZuFi-Dokument darf keine String-Elemente mit leerem Inhalt haben.	21.05.2021	21.11.2021	DQ	Hinweis
VAL0038	Leistung mit unzulässiger Typisierung. Der Typ der Leistung muss im zulässigen Wertebereich liegen.	21.05.2021	21.11.2021	T, ST, DQ	Fehler
VAL0037	Leistung ohne Typisierung. Leistungen müssen mindestens eine Typisierung enthalten.	28.01.2021	28.07.2021	T, ST	Fehler

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0036	Pflicht Textmodule an der Leistung. Leistungen müssen ein nicht-leeres Textmodul mit Code 02 oder 03 enthalten. Enthält eine Leistung kein Textmodul mit Code 06 (oder nur eines mit leerem Inhalt), muss sie mindestens eine Spezialisierung enthalten und alle ihre Spezialisierungen müssen ein Textmodul mit Code 06 liefern. Generell dürfen Spezialisierungen keine Textmodule mit Code 02, 03 oder 06 mit leerem Inhalt haben, wenn die Spezialisierung die Inhalte ersetzt und nicht ergänzt. Für XZuFi 2.3.1 gelten diese Regeln je angegebenem Zugangskanal. Die Prüfung erfolgt pro in der Leistung definierte Sprachversion.	28.01.2021	28.07.2021	DQ	Hinweis
VAL0035	Minimale Zuständigkeit. Für eine Leistung muss mindestens eine Organisationseinheit oder ein Onlinedienst zuständig sein.	28.01.2021	28.07.2021	DQ	Hinweis
VAL0034	Anschrift Test. Die Anschrift einer Organisationseinheit muss vollständig sein.	28.01.2021	28.07.2021	DQ	Fehler
VAL0033	URI-Format Test. Das Format von URIs bzw. URLs muss dem 'RFC 2396' entsprechen.	28.01.2021	28.07.2021	T	Fehler
VAL0032	Quelle der Codeliste. URI und Version einer Codeliste müssen definiert sein.	03.12.2020	03.06.2021	T, ST, DQ	Fehler
VAL0031	Eindeutigkeit des Identifikators. Der Identifikator eines Geschäftsobjektes muss eindeutig sein.	03.12.2020	03.06.2021	T	Fehler
VAL0030	Eindeutigkeit des Textmoduls. Bei den Daten in XZuFi-2.2-Format dürfen zwei Textmodule einer Leistung nicht denselben Code haben. Bei den Daten in XZuFi 2.3.1-Format dürfen zwei Textmodule einer Leistung gleichen Code haben, wenn diese verschiedene Zugangskanäle besitzen. Eine Kombination aus 'Online/Offline = 000' und 'Online = 100' bzw. 'Offline = 200' ist nicht erlaubt.	03.12.2020	03.06.2021	DQ	Fehler

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0029	Doppelte Zuständigkeiten an der Organisationseinheit. Zwei Zuständigkeiten einer Organisationseinheit dürfen nicht identisch sein.	03.12.2020	03.06.2021	T	Fehler
VAL0028	Referenz einer Leistung an der Organisationseinheit. Eine in der Organisationseinheit referenzierte Leistung muss existieren.	03.12.2020	03.06.2021	DQ (RI)	Fehler
VAL0027	Gültige Identifikator Test. Ein Geschäftsobjekt muss einen Identifikator im vorgeschriebenen Format besitzen.	03.12.2020	03.06.2021	T, DQ	Fehler
VAL0026	Format des Gültigkeitsgebietes. Das Gültigkeitsgebiet einer Leistung muss ARS-Format haben.	03.12.2020	03.06.2021	T	Fehler
VAL0025	Gültigkeitszeitraum Test. Die bereitgestellten Daten dürfen keine zeitlichen Widersprüche enthalten.	03.12.2020	03.06.2021	DQ	Fehler
VAL0024	Keine Hierarchie der Organisationseinheiten. Organisationseinheiten dürfen weder übergeordnete noch untergeordnete Organisationseinheiten besitzen.	03.12.2020	03.06.2021	T	Fehler
VAL0023	Unterstützte Sprachen. Nicht unterstützte Sprachen dürfen nicht geliefert werden.	03.12.2020	03.06.2021	DQ	Fehler
VAL0022	Leistungsschlüssel-Format Test. Die Leistung muss mindestens einen Leistungsschlüssel im vorgegebenen Format enthalten.	03.12.2020	03.06.2021	T	Fehler
VAL0021	Gültigkeitsgebiet der Leistung. Die Leistung muss ein Gültigkeitsgebiet besitzen.	03.12.2020	03.06.2021	T	Fehler
VAL0020	Doppelte Zuständigkeiten am Onlinedienst. Zwei Zuständigkeiten am Onlinedienst dürfen nicht identisch sein.	03.12.2020	03.06.2021	T, DQ	Warnung
VAL0019	Zuständigkeit des Onlinedienstes. Der Onlinedienst muss mindestens eine Zuständigkeit besitzen.	03.12.2020	03.06.2021	DQ	Fehler
VAL0018	Format des Gebietes an der Onlinedienstzuständigkeit. Das Gebiet an der Onlinedienstzuständigkeit muss ARS-Format haben.	03.12.2020	03.06.2021	T, ST, DQ	Fehler

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0017	Referenz einer Leistung im Onlinedienst. Eine im Onlinedienst referenzierte Leistung muss existieren.	03.12.2020	03.06.2021	DQ	Fehler
VAL0016	Erlaubte Gebiete am Onlinedienst. Ein Redaktionssystem darf die Daten nur für erlaubte Gebiete liefern.	03.12.2020	03.06.2021	DQ	Fehler
VAL0015	Zuständigkeit der Organisationseinheit. Eine Organisationseinheit muss für mindestens eine Leistung zuständig sein.	03.12.2020	03.06.2021	DQ (RI)	Fehler
VAL0014	Spezialisierungsgebiet Test. Das Spezialisierungsgebiet muss im Gültigkeitsgebiet der Leistung liegen.	03.12.2020	03.06.2021	DQ	Fehler
VAL0013	Spezialisierungsmodul Test. Die Spezialisierung muss ein zu spezialisierendes Modul besitzen.	03.12.2020	03.06.2021	DQ	Fehler
VAL0012	Erlaubte Gebiete. Ein Redaktionssystem darf die Daten nur für erlaubte Gebiete liefern.	03.12.2020	03.06.2021	T	Fehler
VAL0011	Unerlaubte HTML-Tags. Das XZuFi-Dokument darf nur HTML-Tags enthalten, die in der XZuFi-Spezifikation zugelassen sind.	03.12.2020	03.06.2021	ST, DQ	Fehler
VAL0010	Format des Gebietes. Das Gebiet an den Geschäftsobjekten muss ARS-Format haben.	03.12.2020	03.06.2021	DQ, T	Fehler
VAL0009	Ursprungsportal Test. Die Leistung muss ein Ursprungsportal haben.	03.12.2020	03.06.2021	DQ	Fehler
VAL0008	Verbotene Geschäftsobjekte (Gebiet). Das Gebiet darf in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler
VAL0007	Verbotene Geschäftsobjekte (Leistungskategorie). Die Leistungskategorie darf in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler
VAL0006	Verbotene Löschoperation. die Löschoperation darf in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler
VAL0005	Verbotene Geschäftsobjekte (Organisationseinheitskategorie). Die Organisationseinheitskategorie darf in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler

Validator Id	Kurzbeschreibung	Veröffentlicht	Geplante Umstellung	Typ	Modus
VAL0004	Verbotene Geschäftsobjekte (OZG-Leistung). Die OZG-Leistung darf in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler
VAL0003	Verbotene Geschäftsobjekte (Personen). Die Personen dürfen in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler
VAL0002	Verbotene Geschäftsobjekte (Zuständigkeit). Die Zuständigkeit darf in einer Transfernachricht nicht übertragen werden.	03.12.2020	03.06.2021	T	Fehler
VAL0001	Zuständigkeitsgebiet Test. Die Zuständigkeit muss mindestens ein Gebiet haben.	03.12.2020	03.06.2021	DQ, T	Fehler

3.5. Beschreibung der Validierungsregeln

Validator-Id	VAL0065
Beschreibung	Zwei gleiche Leistungsmodule dürfen keine überschneidenden Zugangskanäle haben. Der Code im XZuFi-2.3.1 neu eingeführten Feld "zugangskanal" an Leistungsmodulen darf nicht doppelt oder überschneidend für denselben Modultyp an einer Leistung angegeben werden. Eine Kombination aus 'Online/Offline = 000' und 'Online = 100' bzw. 'Offline = 200' ist eine Überschneidung und daher nicht erlaubt. Die doppelte Angabe eines gleichen Zugangskanals ist ebenfalls nicht erlaubt. modulText(Individuell) benötigt eine erweiterte Prüfung und wird von VAL0065 nicht getestet. Siehe dafür VAL0030. modulUrsprungsportal und modulBegriffeImKontext werden nicht geprüft, da hier auch in XZuFi 2.2.0 schon mehrere Module erlaubt waren (bei Ursprungsportalmodul für Mehrsprachigkeit erforderlich).
Begründung	**Datenqualitäts- Validator:** Wenn eine Leistung pro Zugangskanal mehr als ein Pflichtmodul (z.B. Textmodul 06, Beschreibungstext/Volltext) enthält, dann herrscht Unklarheit darüber, welches Modul an APIs oder in Frontends von Portalen verwendet werden soll. Um Eindeutigkeit und damit Nutzbarkeit zu wahren, lehnt dieser Validator gleiche Module mit doppelten Zugangskanälen ab, vorerst auf Warnung.

Beispiel XZuFi 2.2

Das Feld "zugangskanal" existiert nicht in XZuFi-2.2. Daher hat der Validator hier keinen Effekt.

Beispiel XZuFi 2.3.1 (Falsch)

Zwei Kostenmodule mit Zugangskanal-Überschneidung. Der Zugangskanal 000 (Online/Offline) beinhaltet bereits den Zugangskanal 100 (Online).

```
<xzufi:leistung>
    ...
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>000</code>
        </xzufi:zugangskanal>
        ...
```

```
    </xzufi:modulKosten>
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>100</code>
        </xzufi:zugangskanal>
        ...
    </xzufi:modulKosten>
    ...
</xzufi:leistung>
```

Zwei Kostenmodule mit demselben Zugangskanal.

```
<xzufi:leistung>
    ...
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>200</code>
        </xzufi:zugangskanal>
        ...
    </xzufi:modulKosten>
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>200</code>
        </xzufi:zugangskanal>
        ...
    </xzufi:modulKosten>
    ...
</xzufi:leistung>
```

Beispiel XZuFi 2.3.1 (Richtig)

Ein Kostenmodul mit Zugangskanal 000 (Online/Offline).

```
<xzufi:leistung>
    ...
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>000</code>
        </xzufi:zugangskanal>
        ...
    </xzufi:modulKosten>
    ...
</xzufi:leistung>
```

Zwei Ursprungsportalmodule mit unterschiedlichen Zugangskanälen 100 (Online) und 200

(Offline).

```
<xzufi:leistung>
    ...
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>100</code>
        </xzufi:zugangskanal>
        ...
    </xzufi:modulKosten>
    <xzufi:modulKosten>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>200</code>
        </xzufi:zugangskanal>
        ...
    </xzufi:modulKosten>
    ...
</xzufi:leistung>
```

Validator-Id	VAL0064
Beschreibung	Ein Formular von einer Organisationseinheit muss eine Zuständigkeit enthalten. Dieser Validator gilt nicht für die Formulare, die als separate Geschäftsobjekte in einer Transfernachricht übertragen werden. Siehe dafür VAL0047.
Begründung	**Datenqualitäts- Validator:** Wenn eine Organisationseinheit direkt mit Formularen geliefert wird, müssen diese eine Zuständigkeit (für die sie umschließende Organisationseinheit) beinhalten. Fehlt diese, kann das Formular nicht mit der entsprechenden Leistung verknüpft werden (Referenz-Inkonsistenz). In diesem Falle gibt der Validator eine Warnung aus.

Beispiel (Falsch)

Beim Formular fehlt das Element *zustaendigkeit*.

```
<xzufi:transfer.operation.040502>
...
    <xzufi:schreibe>
        <xzufi:organisationseinheit>
            <xzufi:formular xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:Formular">
                <xzufi:id schemeAgencyID="L000000">10891301</xzufi:id>
                <xzufi:bezeichnung languageCode="de">Formular
extern</xzufi:bezeichnung>
                <xzufi:kurzbezeichnung
languageCode="de">Wohngeld</xzufi:kurzbezeichnung>
                <xzufi:beschreibung languageCode="de">Beispiel
Formular</xzufi:beschreibung>
                <xzufi:datei xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:FormulardateiLink">
                    <xzufi:uri>https://www.host.org</xzufi:uri>
                </xzufi:datei>
                <xzufi:versionsinformation>
                    <xzufi:geaendertDatumZeit>2021-10-
10T12:00:01</xzufi:geaendertDatumZeit>
                    <xzufi:geaendertDurch>Redaktionssystem</xzufi:geaendertDurch>
                </xzufi:versionsinformation>
                <xzufi:sprachversion>
                    <xzufi:languageCode>de</xzufi:languageCode>
                </xzufi:sprachversion>
            </xzufi:formular>
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
```

```
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Des Formular enthält alle benötigten Elemente.

```
<xzufi:transfer.operation.040502>
...
    <xzufi:schreibe>
        <xzufi:organisationseinheit>
            <xzufi:formular xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:Formular">
                <xzufi:id schemeAgencyID="L000000">10891301</xzufi:id>
                <xzufi:bezeichnung languageCode="de">Formular
extern</xzufi:bezeichnung>
                <xzufi:datei xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:FormulardateiLink">
                    <xzufi:uri>https://www.host.org</xzufi:uri>
                </xzufi:datei>
                <xzufi:zustaendigkeit>
                    <xzufi:leistungID
schemeAgencyID="L000000">27302</xzufi:leistungID>
                    <xzufi:gebietID
schemeAgencyID="L000000">090000000000</xzufi:gebietID>
                </xzufi:zustaendigkeit>
                <xzufi:versionsinformation>
                    <xzufi:geaendertDatumZeit>2021-10-
10T12:00:01</xzufi:geaendertDatumZeit>
                    <xzufi:geaendertDurch>Redaktionssystem</xzufi:geaendertDurch>
                </xzufi:versionsinformation>
                <xzufi:sprachversion>
                    <xzufi:languageCode>de</xzufi:languageCode>
                </xzufi:sprachversion>
            </xzufi:formular>
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0063
Beschreibung	Eine Leistung enthält nicht alle Typisierungen des Leistungsschlüssels, die vom FIM-Leistungskatalog vorgegeben sind. Die Zuordnung zum FIM-Leistungskatalog findet über dem Leistungsschlüssel aus dem Element *'referenzLeika'* der Leistung statt. Bei der Prüfung der Typisierung werden nur die Ziffern verglichen, etwaige Spezialisierungen werden dabei nicht betrachtet (*'2a'* wird wie *'2'* behandelt). Es werden nur Leistungen geprüft, deren Leistungsschlüssel im FIM-Leistungskatalog vorkommen. Dieser Validator hat nur informativen Charakter und wird perspektivisch nur im 'Hinweis'-Modus betrieben.
Begründung	**Datenqualitäts- Validator:** Der LeiKa gibt die zulässigen Typisierungen vor - also die Bezeichnung die festlegt, ob Bund, Land oder Kommune das letzte Wort bei der Gestaltung der Leistung hat. Wenn eine Leistung mit anderer Typisierung als erwartet geliefert wird, deutet das auf inhaltlich unzulässige Änderungen hin. Um darauf hinzuweisen, reagiert diese Validierungsregel mit einem Hinweis.

Beispiel (Falsch)

Der *'FIM-Leistungskatalog'* gibt zum Leistungsschlüssel '99001020008000' die Typisierungs-Codes *'1'* und *'2'* aus, die Leistung wird jedoch nur mit dem Code *'1'* geliefert. Es fehlt also der Typisierungs-Code *'2'* an der Leistung.

```
<xzufi:leistung>
    ...
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99001020008000</code>
    </xzufi:referenzLeiKa>
    ...
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>1</code>
    </xzufi:typisierung>
...

</xzufi:leistung>
```

Beispiel (Richtig)

Der *'FIM-Leistungskatalog'* gibt zum Leistungsschlüssel '99001020008000' die Typisierungs-Codes *'1'* und *'2'* vor, die Leistung wird ebenfalls mit den Codes *'1'* und *'2'* geliefert. Die Leistung ist damit vollständig typisiert.

```
<xzufi:leistung>
    ...
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99001020008000</code>
    </xzufi:referenzLeiKa>
    ...
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>1</code>
        <code>2</code>
    </xzufi:typisierung>
...

</xzufi:leistung>
```

Validator-Id	VAL0062
Beschreibung	Eine Leistung enthält Typisierungen des Leistungsschlüssels, die nicht vom FIM-Leistungskatalog vorgegeben sind. Die Zuordnung zum FIM-Leistungskatalog findet über dem Leistungsschlüssel aus dem Element *'referenzLeika'* der Leistung statt. Bei der Prüfung der Typisierung werden nur die Ziffern verglichen, etwaige Spezialisierungen werden dabei nicht betrachtet (*'2a'* wird wie *'2'* behandelt). Es werden nur Leistungen geprüft, deren Leistungsschlüssel im FIM-Leistungskatalog vorkommen.
Begründung	**Datenqualitäts- Validator:** Der LeiKa gibt die zulässigen Typisierungen vor - also die Bezeichnung die festlegt, ob Bund, Land oder Kommune das letzte Wort bei der Gestaltung der Leistung hat. Wenn eine Leistung mit anderer Typisierung als erwartet geliefert wird, deutet das auf inhaltlich unzulässige Änderungen hin. Um darauf hinzuweisen, reagiert diese Validierungsregel mit einer Warnung.

Beispiel (Falsch)

Der *'FIM-Leistungskatalog'* gibt zum Leistungsschlüssel '99001020008000' den Typisierungs-Code *'2'* vor, die Leistung wird jedoch mit dem Code *'3'* geliefert. Das ist nicht erlaubt.

```
<xzufi:leistung>
    ...
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99001020008000</code>
    </xzufi:referenzLeiKa>
    ...
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>3</code>
    </xzufi:typisierung>
...

</xzufi:leistung>
```

Beispiel (Richtig)

Der *'FIM-Leistungskatalog'* gibt zum Leistungsschlüssel '99001020008000' den Typisierungs-Code *'2'* vor, die Leistung wird jedoch mit dem Code *'2a'* geliefert. Der Code ist gültig, da nur die Ziffer geprüft wird und die Buchstaben (Spezialisierung durch die Landesredaktionen) ignoriert werden.

```
<xzufi:leistung>
    ...
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99001020008000</code>
    </xzufi:referenzLeiKa>
    ...
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>2a</code>
    </xzufi:typisierung>
...

</xzufi:leistung>
```

Validator-Id	VAL0060
Beschreibung	Eine Leistung darf keine Kategorie mit der Bezeichnung 'Mustertext' enthalten.
Begründung	**Datenqualitäts- Validator:** Mustertexte existieren im FIM-Lifecycle nur im FIM-Portal und dem Sammelrepository, nicht aber dem PVOG. Diese getrennten Systeme erfüllen unterschiedliche Funktionen. Um eine Vermischung zu vermeiden, beanstandet diese Validierungsregel Leistungen mit der Kategorie Mustertext per Hinweis.

Beispiel (Falsch)

Eine Leistungskategorie enthält die Bezeichnung "Mustertext":

```
<xzufi:leistung>
...
    <xzufi:kategorie>
        <xzufi:id schemeAgencyID="L100012" schemeID="OID">270357767</xzufi:id>
        <xzufi:bezeichnung languageCode="de">Mustertext</xzufi:bezeichnung>
    </xzufi:kategorie>
...
</xzufi:leistung>
```

Beispiel (Richtig)

Diese Leistung hat keine Kategorie mit der Bezeichnung 'Mustertext':

```
<xzufi:leistung>
...
    <xzufi:kategorie>
        <xzufi:id schemeAgencyID="L100012" schemeID="OID">270357767</xzufi:id>
        <xzufi:bezeichnung languageCode="de">Klima, Natur und
Arten</xzufi:bezeichnung>
    </xzufi:kategorie>
...
</xzufi:leistung>
```

Validator-Id	VAL0059
Beschreibung	Die landesbezogenen Leistungsschlüssel einer Leistung müssen zum Handbuch LeiKa-plus konform sein. Dazu gehört, dass sie mit einer Länderkennung des Amtlichen Gemeindeschlüssels (AGS) beginnen und in den folgenden Abschnitten nur Werte aus dem Bereich 900-999 enthalten dürfen.
Begründung	**Standard- und Datenqualitäts- Validator:** Das LeiKa-Plus gilt als verbindlicher Standard von Leistungsdaten. Diese sollen eingehalten werden, um die Datenqualität insgesamt zu wahren.

Beispiel (Falsch)

Ein Leistungsschlüssel der Form *40903911222000* verwendet eine falsche Länderkennung und eine Verrichtungskennung, die nicht aus dem Bereich 900-999 kommt.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>40903911222000</code>
    </xzufi:referenzLeiKa>
...
</xzufi:leistung>
```

Beispiel (Richtig)

Der Leistungsschlüssel *11903911922000* verwendet eine korrekte Länderkennung und in den folgenden Abschnitten Zahlen aus dem Bereich 900-999. Die Abschnitte Leistungskennung, Verrichtungskennung und Verrichtungsdetail dürfen den Platzhalter '000' verwenden.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>11903911922000</code>
    </xzufi:referenzLeiKa>
...
</xzufi:leistung>
```

Validator-Id	VAL0057
Beschreibung	Zwei Leistungen dürfen nicht den gleichen Leistungsschlüssel referenzieren.

Beispiel (Falsch)

Zwei Leistungen enthalten den gleichen Leistungsschlüssel (definiert im Feld *code* unter *referenzLeiKa*).

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L100012" schemeID="OID">1</xzufi:id>
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
    ...
</xzufi:leistung>

<xzufi:leistung>
    <xzufi:id schemeAgencyID="L100012" schemeID="OID">2</xzufi:id>
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
    ...
</xzufi:leistung>
```

Beispiel (XZuFi 2.3.1, Falsch)

Zwei Leistungen enthalten den gleichen Leistungsschlüssel.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L100012" schemeID="OID">1</xzufi:id>
    <xzufi:referenzLeistungsschluessel>
      99038009223001
    </xzufi:referenzLeistungsschluessel>
    ...
</xzufi:leistung>

<xzufi:leistung>
    <xzufi:id schemeAgencyID="L100012" schemeID="OID">2</xzufi:id>
    <xzufi:referenzLeistungsschluessel>
      99038009223001
    </xzufi:referenzLeistungsschluessel>
    ...
</xzufi:leistung>
```

Validator-Id	VAL0056
Beschreibung	Leistungsschlüssel aus dem Element *'referenzLeika'* müssen im *'FIM-Leistungskatalog'* existieren.
Begründung	**Datenqualitäts- Validator:** Die Referenz zum LeiKa ist notwendig, um Leistungen mit einander in Zusammenhang/Kontext zu bringen. Fehlt diese Information, leidet die Datenqualität. Der Validator 46 prüft lediglich die Form des Referenz-Leistungsschlüssels (99_XXXXXXXXXXXX), dieser Validator prüft darüberhinaus, ob der angegebene Schlüssel im LeiKa existiert.

Beispiel (Falsch)

Die Codes für die Leistungschlüssel im *'FIM-Leistungskatalog'* beginnen alle mit den Ziffern "99". Folgender Code ist im FIM-Leistungskatalog nicht vorhanden und ist fehlerhaft.

```
<xzufi:leistung>
    ...
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>00001020008001</code>
    </xzufi:referenzLeiKa>
    ...
</xzufi:leistung>
```

Beispiel (Richtig)

Folgender Code ist im *'FIM-Leistungskatalog'* vorhanden. Das XZuFi ist in diesem Fall gültig.

```
<xzufi:leistung>
    ...
    <xzufi:referenzLeiKa listVersionID="20180705" listURI="urn:de:fim:leika:leistung">
        <code>99001020008000</code>
    </xzufi:referenzLeiKa>
    ...
</xzufi:leistung>
```

Validator-Id	VAL0055
Beschreibung	Onlinedienste müssen eine ID, sowie eine schemeAgencyID in gültiger Form enthalten.
Begründung	**Technischer Validator:** Ohne eine individuelle ID und den Identifikator des liefernden Redaktionssystems (schemeAgencyID) kann ein Onlinedienstobjekt nicht korrekt verarbeitet werden. Folglich ist diese Validierungsregel technisch notwendig.

Beispiel (Falsch)

Das Element *xzufi:id* enthält kein Attribut *schemeAgencyID* und die ID selbst enthält das ungültige Zeichen ';'.

```
<xzufi:onlinedienst p3:type="xzufi:OnlinedienstErweitert"
xmlns:p3="http://www.w3.org/2001/XMLSchema-instance">
  <xzufi:id>ID;01</xzufi:id>
  ...
</xzufi:onlinedienst>
```

Beispiel (Richtig)

In diesem Fall sind sowohl das Attribut *schemeAgencyID*, sowie die ID im Element *xzufi:id* vorhanden und in gültiger Form.

```
<xzufi:onlinedienst p3:type="xzufi:OnlinedienstErweitert"
xmlns:p3="http://www.w3.org/2001/XMLSchema-instance">
  <xzufi:id schemeAgencyID="S100002">S1000020040000000130</xzufi:id>
  ...
</xzufi:onlinedienst>
```

Validator-Id	VAL0054
Beschreibung	ARS-Schlüssel müssen in der Codeliste 'urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs' existieren. Ist der ARS-Schlüssel mit Nullen aufgefüllt und stellt also eine höhere Ebene dar, so wird geprüft, ob mindestens ein Element aus der unteren Ebene existiert. Bsp.: Es wird der Schlüssel *010515163000* geprüft. Als Erstes werden die Nullen entfernt, bis eine höhere Ebene erreicht wurde. In unserem Beispiel werden 3 Nullen entfernt bis die Gemeindeverband-Ebene *010515163003* erreicht wurde. Jetzt wird geprüft, ob die ARS-Liste mindestens eine Gemeinde im Gemeindeverband *010515163* enthält. Ist in der Liste mindestens eine Gemeinde *010515163003*, *010515163010*, usw. enthalten, so wird der Gemeindeverband *010515163000* akzeptiert.
Begründung	**Datenqualitäts- Validator:** Wenn ARS-Schlüssel nicht mit der Codeliste übereinstimmen, kann die Zuständigkeitsfindung nicht korrekt funktionieren. Die Daten selbst können allerdings verarbeitet werden. Insofern erhöht dieser Validator lediglich die Datenqualität.

Beispiel (Voraussetzung)

Der ARS ist 12-stellig setzt sich wie folgt zusammen:

1.–2. Stelle	Bundesland
3. Stelle	Regierungsbezirk oder 0
4.–5. Stelle	Landkreise oder kreisfreie Stadt
6.–9. Stelle	Verbandsschlüssel
10.–12. Stelle	Gemeindekennzahl

Wird ein ARS mit weniger als 12 Stellen übertragen dann werden die restlichen Stellen vom PVOG mit "0" aufgefüllt. Die ARS-Liste *urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs* enthält die Elemente:
…
010040000000 Neumünster, Stadt
010510011011 Brunsbüttel, Stadt
010510044044 Heide, Stadt
010515163003 Averlak
010515163010 Brickeln
…

Beispiel (Falsch)

In der Liste ist der Gemeindeverband *010510030000* nicht präsent. Es gibt aber auch keine einzige Gemeinde, die mit *010510030* beginnen würde bzw. dem Gemeindeverband dazugehören würde. Der ARS *010510030000* wird nicht akzeptiert.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">010510100000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Beispiel (Richtig)

In der Liste ist der Gemeindeverband *010515163000 Burg-St. Michaelisdonn* nicht präsent. Es sind aber die Gemeinden *010515163003 Averlak* und *010515163010 Brickeln* enthalten. Der ARS *010515163000* wird akzeptiert.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">010515163000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Bei der Datenübertragung wird die Liste *urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:gemeindeverband* verwendet.
Die Liste wird vom Validator ignoriert und für die Prüfung die aktuellste ARS-Liste verwendet. Ist der Schlüssel in der Liste enthalten, so wird der Schlüssel akzeptiert, sonst nicht.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:gemeindeverband"
schemeVersionID="2022-05-31">010515163000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
```

```
</xzufi:organisationseinheit>
```

Validator-Id	VAL0053
Beschreibung	Das XZuFi-Dokument darf nur maskierte HTML-Tags enthalten, die in der XZuFi-Spezifikation zugelassen sind. Ausnahmen sind hier nur die HTML-Tags 'div' mit dem Attribut 'lang' oder Attribut 'href' am Attribut 'a'. Die HTML-Tags dürfen sich zusätzlich nur an den in der Spezifikation zugelassenen XZuFi-Elementen befinden.
Begründung	**Standard- und Datenqualitäts- Validator:** HTML-Tags sind aus Sicherheits- und Konformitätsgründen eingeschränkt worden. Es könnte z.B. ausführbarer (Schad-)Code eingebettet werden, oder aber Layout-Erfordernisse von nachnutzenden Systemen/Portalen erschwert werden. Um dies zu vermeiden, wurde bereits im XZuFi-Standard festgelegt, welche Tags an welchen Stellen erlaubt sind. Der Validator gibt eine Warnung aus, wenn unerlaubte Tags gefunden werden.

Beispiel (Falsch)

Das HTML-Tag "table" und die dazugehörigen Kind-Elemente sind laut xZuFi-Spezifikation nicht erlaubt.

```
<xzufi:modulText>
    <xzufi:positionDarstellung>3</xzufi:positionDarstellung>
    <xzufi:inhalt languageCode="de">
        <div> Hier kommen nicht erlaubte Tags vor </div>
        <table>
        <tr/>
          <th>Monat</th>
          <th>Datum</th>
        </tr>
        <tr>
          <td>Januar</td>
          <td>01.01.2022</td>
        <tr/>
      </table>
    </xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
                          listVersionID="20150601">
        <code>11</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Es wird im Element *infoOeffnungszeitenText* das HTML-Tag *
* verwendet. Nach XZuFi Standard

darf das Element *infoOeffnungszeitenText* keine HTML Tags enthalten.

```
<xzufi:organisationseinheit p3:type="xzufi:OrganisationseinheitErweitert"
xmlns:p3="http://www.w3.org/2001/XMLSchema-instance">
    <xzufi:id schemeAgencyID="S000000">00123</xzufi:id>
    <xzufi:infoOeffnungszeitenText languageCode="de">Den Kundenservice erreichen Sie
telefonisch. <br />
Ihre Anliegen erledigen Sie bitte weiterhin online. <br />
    </xzufi:infoOeffnungszeitenText>
</xzufi:organisationseinheit>
```

Beispiel (Richtig)

Ein Absatz *<p>* und eine Liste *
* sind im XZuFi Standard erlaubt.

```
<xzufi:modulText>
    <xzufi:positionDarstellung>3</xzufi:positionDarstellung>
    <xzufi:inhalt languageCode="de">
        <p> Hier kommen erlaubte Tags vor </p>
        <ul>
            <li>Ein Absatz und </li>
            <li> eine Liste sind erlaubt. </li>
        </ul>
    </xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
                          listVersionID="20150601">
        <code>11</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Das Element *infoOeffnungszeitenText* enthält keine HTML-Tags.

```
<xzufi:organisationseinheit p3:type="xzufi:OrganisationseinheitErweitert"
xmlns:p3="http://www.w3.org/2001/XMLSchema-instance">
    <xzufi:id schemeAgencyID="S000000">00123</xzufi:id>
    <xzufi:infoOeffnungszeitenText languageCode="de">Den Kundenservice erreichen Sie
telefonisch. Ihre Anliegen erledigen Sie bitte weiterhin online.
    </xzufi:infoOeffnungszeitenText>
</xzufi:organisationseinheit>
```

Validator-Id	VAL0052
Beschreibung	Der Validator prüft, ob die in den Elementen *listURI* und *listVersion* definierte Codeliste im XRepository https://www.xrepository.de existiert und die Codeliste den übertragenen Code enthält. Eine Ausnahme stellt *TextmodulIndividuell* dar. Das individuelle Textmodul wird nicht geprüft.
Begründung	**Technischer und Standard- Validator:** Codelisten sind für diverse Prüfungen und Verarbeitungsschritte im PVOG notwendig, daher gibt der Validator bei Verletzung des Standards eine Warnung aus. Die Datenverantwortlichen werden so darauf aufmerksam, dass es neue Codelisten gibt oder verwendete nicht mehr exestieren.

Beispiel (Falsch)

Die Liste *urn:de:xzufi:codeliste:leistungstextmodulleika* existiert in der Version *20200101* nicht.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20200101">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Der Code *2* existiert in der Liste *urn:de:xzufi:codeliste:leistungstextmodulleika* in der Version *20201201* nicht. Es existiert der Code *02*.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>2</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Beispiel (Richtig)

Die Liste *urn:de:xzufi:codeliste:leistungstextmodulleika* in der Version *20201201* sowie der Code *02* existieren.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Die Liste *urn:de:xzufi:codeliste:zustaendigkeitsrolle* in der Version *20210401* sowie der Code *03* existieren.

```
<xzufi:zustaendigkeit>
    ...
    <xzufi:rolle listURI="urn:de:xzufi:codeliste:zustaendigkeitsrolle"
listVersionID="20210401">
        <code>03</code>
        <name>Ansprechpunkt</name>
    </xzufi:rolle>
</xzufi:zustaendigkeit>
```

Validator-Id	VAL0051
Beschreibung	Die Leistungen enthalten eine Zuordnung zu den PV-Lagen. Die Zuordnung muss sich an der Zuordnung aus dem Leistungskatalog des FIM orientieren. PV-Lagen werden aus den Elementen *klassifizierung* mit dem Attribut *listUri="urn:xoev-de:fim:codeliste:pvlagen"* ermittelt und mit den Einträgen aus dem Leistungskatalog abgeglichen. Zum Ermitteln der PV-Lagen aus dem Leistungskatalog wird der Leistungsschlüssel aus dem Element *referenzLeiKa* der Leistung verwendet. Prüfregeln: - Der Leistungskatalog definiert PV-Lagen für die Leistung aber die Leistung enthält keine Lagen. - Der Leistungskatalog und die Leistung definieren die Lagen. Die Lagen sind aber nicht identisch.
Begründung	**Datenqualitäts- Validator:** Die PV-Lagen haben zum Ziel, fachliche und inhaltliche Verbindungen zwischen Leistungen aufzuzeigen und die Navigation zu vereinfachen. Das kann nur funktionieren, wenn das offizielle Schema eingehalten wird. Wenn Leistungen andere Lagen zugeordnet haben, als im LeiKa vorgesehen sind, führt das dazu, dass die Leistung den Zweck des Konzepts "Lebenslagen" nicht mehr erfüllt, allerdings ist die Nutzung und Anzeige in Verwaltungsportalen weiterhin problemlos möglich. Deshalb gibt der Validator lediglich einen Hinweis aus, sollten die Lagen nicht zum Standard passen, dh. wenn keine oder eine abweichende Lage angegeben ist.

Beispiel (Falsch)

Der Leistungskatalog für die Leistungsschlüssel *99038009223001* definiert zwei Lagen *2010100* und *2010400*. Die Leistung definiert nur eine Lage *2010100*. Die Lage *2010400* fehlt.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
...
    <xzufi:klassifizierung listURI="urn:xoev-de:fim:codeliste:pvlagen"
listVersionID="20211001">
```

```
        <code>2010100</code>
    </xzufi:klassifizierung>
<xzufi:leistung>
```

Beispiel (Richtig)

Der Leistungskatalog für die Leistungsschlüssel *99038009223001* definiert zwei Lagen *2010100* und *2010400*. Die Leistung definiert beide Lagen.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
...
    <xzufi:klassifizierung listURI="urn:xoev-de:fim:codeliste:pvlagen"
listVersionID="20211001">
        <code>2010100</code>
    </xzufi:klassifizierung>
    <xzufi:klassifizierung listURI="urn:xoev-de:fim:codeliste:pvlagen"
listVersionID="20211001">
        <code>2010400</code>
    </xzufi:klassifizierung>
<xzufi:leistung>
```

Validator-Id	VAL0050
Beschreibung	Die Leistung darf maximal einen Leistungsschlüssel enthalten. Das Element *referenzLeiKa* darf nur einmal vorkommen bzw. den gleichen Leistungsschlüssel *code* enthalten.
Begründung	**Datenqualitäts- Validator:** Wenn eine Leistung auf zwei Referenz-Leistungen verweist, kann ein nachnutzendes System, z.B. ein Verwaltungsportal, keinen Zusammenhang mit den definierten Leistungen des LeiKa herstellen. Die Folge wäre, dass Auswertungen oder Verknüpfungen nicht möglich wären, die grunsätzliche Verarbeitung und Anzeige in Verwaltungsportalen wäre nicht gefährdet. Aus diesem Grund gibt der Validator aus Datenqualitätsgründen lediglich eine Warnung aus.

Beispiel (Falsch)

Der Leistung sind zwei Leistungsschlüssel *99038009223001*, *99038009223002* zugeordnet.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223002</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Beispiel (Richtig)

Der Leistung ist nur ein Leistungsschlüssel zugeordnet.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Validator-Id	VAL0049
Beschreibung	Zwei Zuständigkeiten dürfen nicht im selben Formular auf dieselbe Menge von Gebiets-IDs (ARS) derselben Leistung verweisen. Die Listen der Gebiet-IDs und der LeistungIDs in zwei Zuständigkeiten müssen disjunkt sein.
Begründung	**Technischer Validator:** Es ist Konvention, dass pro Gebiet alle Zuständigkeiten in einem Objekt abgebildet werden sollen. Wenn ein Formular aber zwei Zuständigkeiten mit identischen Gebieten enthielte, könnte es zu Dopplungen kommen, die zu referenziellen Inkonsistenzen und faktischem Datenverlust oder -verfälschung führen würden. Um das zu vermeiden, lehnt der Validator Formulare mit gebiets-identischen Zuständigkeiten ab.

Beispiel (Falsch)

Die GebietID *090000000000* kommt in zwei Zuständigkeiten vor.

```
...
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
...
```

Beispiel (Richtig)

Die GebietID *090000000000* kommt in einer separaten Zuständigkeit nur einmal vor.

```
...
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
<xzufi:zustaendigkeit>
```

```
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
...
```

Validator-Id	VAL0048
Beschreibung	Die Zuständigkeit im Formular muss auf eine Leistung in demselben XZuFi-Dokument verweisen. Somit wird sichergestellt, dass eine Zuständigkeit existierende Geschäftsobjekte verknüpft.
Begründung	**Technischer Validator:** Wenn ein Formular eine Leistung referenziert, muss diese auch existieren. Da ohne diese Verknüpfung das Formularobjekt u.U. komplett unauffindbar sein könnte - es läge eine referenzielle Inkonsistenz vor - lehnt der Validator solche Formular aus technischen Gründen ab.

Beispiel (Falsch)

Die Zuständigkeit referenziert eine Leistung mit ID *4815*. Diese Leistung ist im Dokument nicht enthalten. Stattdessen gibt es eine Leistung mit ID *3815*.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L00000">3815</xzufi:id>
...
    <xzufi:leistung>
<xzufi:schreibe>
<xzufi:schreibe>
    <xzufi:formular>
        ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">4815</xzufi:leistungID>
            <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
        </xzufi:zustaendigkeit>
    </xzufi:formular>
</xzufi:schreibe>
```

Beispiel (Richtig)

Die Zuständigkeit referenziert eine Leistung mit ID *3815*. Die Leistung mit ID *3815* ist auch im Dokument enthalten.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L00000">3815</xzufi:id>
...
    <xzufi:leistung>
<xzufi:schreibe>
<xzufi:schreibe>
    <xzufi:formular>
        ...
```

```
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
            <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
        </xzufi:zustaendigkeit>
    </xzufi:formular>
</xzufi:schreibe>
```

Validator-Id	VAL0047
Beschreibung	Ein Formular muss eine Bezeichnung, eine Zuständigkeit und einen Link auf das Formular enthalten. Formulare, die als Teil der Organisationseinheit übertragen werden, werden nicht auf Zuständigkeiten geprüft. Das ist in VAL0064 ausgelagert. Formulare, die als separate Geschäftsobjekte in einer Transfernachricht übertragen werden, werden vollständig geprüft. Formulare, die in einem Textmodul übertragen werden, sind von der Prüfung ausgenommen.
Begründung	**Datenqualitäts-Validator:** Der Validator prüft, ob die als "fachlich notwendig" definierten Angaben gefüllt sind. Dies ist Teil der Datenqualität, da Formularobjekte ohne Titel oder Link zur Formulardatei ihren Zweck nicht erfüllen können.

Beispiel (Falsch)

Beim Formular fehlen die Elemente *bezeichnung* und *zustaendigkeit*.

```
<xzufi:transfer.operation.040502>
...
    <xzufi:schreibe>
        <xzufi:formular xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:Formular">
            <xzufi:id schemeAgencyID="L000000">10891301</xzufi:id>
            <xzufi:kurzbezeichnung languageCode="de">Wohngeld</xzufi:kurzbezeichnung>
            <xzufi:beschreibung languageCode="de">Beispiel
Formular</xzufi:beschreibung>
            <xzufi:datei xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:FormulardateiLink">
                <xzufi:uri>https://www.host.org</xzufi:uri>
            </xzufi:datei>
            <xzufi:versionsinformation>
                <xzufi:geaendertDatumZeit>2021-10-
10T12:00:01</xzufi:geaendertDatumZeit>
                <xzufi:geaendertDurch>Redaktionssystem</xzufi:geaendertDurch>
            </xzufi:versionsinformation>
            <xzufi:sprachversion>
                <xzufi:languageCode>de</xzufi:languageCode>
            </xzufi:sprachversion>
        </xzufi:formular>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Des Formular enthält alle benötigten Elemente.

```
<xzufi:transfer.operation.040502>
...
    <xzufi:schreibe>
        <xzufi:formular xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:Formular">
            <xzufi:id schemeAgencyID="L000000">10891301</xzufi:id>
            <xzufi:bezeichnung languageCode="de">Formular extern</xzufi:bezeichnung>
            <xzufi:datei xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xzufi:FormulardateiLink">
                <xzufi:uri>https://www.host.org</xzufi:uri>
            </xzufi:datei>
            <xzufi:zustaendigkeit>
                <xzufi:leistungID schemeAgencyID="L000000">27302</xzufi:leistungID>
                <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
            </xzufi:zustaendigkeit>
            <xzufi:versionsinformation>
                <xzufi:geaendertDatumZeit>2021-10-
10T12:00:01</xzufi:geaendertDatumZeit>
                <xzufi:geaendertDurch>Redaktionssystem</xzufi:geaendertDurch>
            </xzufi:versionsinformation>
            <xzufi:sprachversion>
                <xzufi:languageCode>de</xzufi:languageCode>
            </xzufi:sprachversion>
        </xzufi:formular>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0046
Beschreibung	Der Referenz-Leistungsschluessel einer Leistung muss ein FIM-Schlüssel sein, weshalb die ersten zwei Ziffern 99 (FIM) oder 77 (temporär) sein müssen.
Begründung	**Datenqualitäts-Validator:** Jede Leistung soll eine Referenz auf die Leika-Leistung bieten, die sie umsetzt. Damit dieser Bezug klar ist, muss der Referenz-Leistungsschlüssel standardkonform konstruiert sein, dh. mit den Ziffern 99 beginnen. Ist dies nicht der Fall, können Konsumenten der Daten nicht erkennen, zu welcher allgemein definierten Leika-Leistung diese Leistung gehört. Dies würde die Datenqualität und Nachnutzbarkeit der Daten stark reduzieren.

Beispiel (Falsch)

Der Leistungsschlüssel *09038009223001* fängt mit dem länderspezifischen Schlüssel *09* an.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>09038009223001</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Beispiel (Richtig)

Der Leistungsschlüssel *99038009223001* fängt mit *99* an.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Validator-Id	VAL0045
Beschreibung	Der Leistungsschlüssel eines Leistung ist von der Form TMP_XX_XXXXXXXXXXXXXX, wobei die X für beliebige Ziffern stehen. TMP-Schlüssel wurden im Sammlerdienst-Konzept nur temporär zugelassen und müssen durch *99er-Schlüssel* ersetzt werden.
Begründung	**Datenqualitäts-Validator:** Die Verwendung von Leistungsschlüsseln ist notwendig für die korrekte Funktionsweise des PVOG und aller anderen Systeme. Damit die Verwendung einfach und standardkonform bleibt, beginnen die von der zentralen Koordinierung vor- bzw. freigegebenen Schlüssel mit 99. Die Redaktionsysteme von Bund und Ländern sollen stattdessen ihr individuelles Kürzel verwenden. Stattdessen TMP zu verwenden, war für die schnelle Einführung neuer Leistungen hilfreich, aber diese Entwicklung ist nunmehr abgeschlossen. Der Validator lehnt TMP-Leistungen allerdings nicht ab, sondern gibt lediglich eine Warnung aus. Die Folge von Verwendung des TMP-Schlüssels könnte sein, dass jene Leistung in PVOG oder folgenden Systemen u.U. nicht aufgefunden würde.

Beispiel (Falsch)

Der Leistungsschlüssel der Form *TMP_02_020000000000* ist nur temporär einzusetzen und muss durch 99 Schlüssel ersetzt werden..

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>TMP_02_020000000000</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Beispiel (Richtig)

Der Leistungsschlüssel *99038009223001* fängt mit *99* an.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
```

```
...
<xzufi:leistung>
```

Validator-Id	VAL0044
Beschreibung	Die Zuständigkeit der Organisationseinheit muss ein Element *rolle* enthalten. Somit kann ermittelt werden, ob es sich bei der Organisationseinheit um einen Ansprechpunkt, eine zuständige Stelle oder um beides handelt.
Begründung	**Datenqualitäts-Validator:** Damit Bürger und Bürgerinnen unterscheiden können, ob eine Organisationseinheit fachlich verantwortlich (zuständige Stelle) oder lediglich für die Ausführung der Leistung zuständig ist (Ansprechpunkt), muss diese Rolle am Organisationseinheit-Objekt definiert sein. Fehlt dies, lehnt der Validator das Objekt entsprechend ab.

Beispiel (Falsch)

Das Element *rolle* ist nicht enthalten.

```
<xzufi:organisationseinheit>
...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
        ...
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Beispiel (Richtig)

Das Element *rolle* ist vorhanden.

```
<xzufi:organisationseinheit>
...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
        <xzufi:rolle listURI="urn:de:xzufi:codeliste:zustaendigkeitsrolle"
listVersionID="20210401">
            <code>01</code>
            <name>Es handelt sich um die sachlich und örtlich zuständige Stelle sowie
Ansprechpunkt für den Bürger.</name>
        </xzufi:rolle>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Validator-Id	VAL0043
Beschreibung	Der Sprachversion-Validator prüft, ob in dem XZuFi-Element 'Sprachversion' der Geschäftsobjekte Leistung, Organisationseinheit und Onlinedienst alle gelieferten Sprachen deklarieren sind. Werden zum Beispiel die Textmodule in englischer und deutscher Sprache geliefert, so muss Element *sprachversion* die Sprachen Deutsch (*de*) und Englisch (*en*) deklarieren.
Begründung	**Datenqualitäts-Validator:** Wenn Inhalte in verschiedenen Sprachversionen geliefert werden, prüft der Validator, ob jede Sprachversion an sich inhaltlich komplett ist. Dazu werden die gemeinschaftlich als Pflicht-Informationen festgelegten Felder geprüft. Ist eine Sprachversion unvollständig, z.B. weil der Titel nur auf Deutsch und nicht auf Englisch geliefert wird, lehnt der Validator das Objekt entsprechend ab.

Beispiel (Falsch)

Das Textmodul wird in englischer *en* und deutscher *de* Sprache geliefert. In 'Sprachversion' deklariert ist aber nur eine Sprache *de*.

```
...
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
    <xzufi:inhalt languageCode="en">Examinations for dental, oral and maxillofacial
diseases</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
....

<xzufi:sprachversion>
    <xzufi:languageCode>de</xzufi:languageCode>
    <xzufi:sprachbezeichnungDeutsch>Deutsch</xzufi:sprachbezeichnungDeutsch>
    <xzufi:sprachbezeichnungNativ>Deutsch</xzufi:sprachbezeichnungNativ>
</xzufi:sprachversion>
```

Beispiel (Richtig)

Das Textmodul wird in englischer *en* und deutscher *de* Sprache geliefert. Sprachversion deklariert ebenfalls zwei Sprachen *de* und *en*.

```
...
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
    <xzufi:inhalt languageCode="en">Examinations for dental, oral and maxillofacial
diseases</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
....

<xzufi:sprachversion>
    <xzufi:languageCode>de</xzufi:languageCode>
    <xzufi:sprachbezeichnungDeutsch>Deutsch</xzufi:sprachbezeichnungDeutsch>
    <xzufi:sprachbezeichnungNativ>Deutsch</xzufi:sprachbezeichnungNativ>
</xzufi:sprachversion>
<xzufi:sprachversion>
    <xzufi:languageCode>en</xzufi:languageCode>
    <xzufi:sprachbezeichnungDeutsch>Englisch</xzufi:sprachbezeichnungDeutsch>
    <xzufi:sprachbezeichnungNativ>English</xzufi:sprachbezeichnungNativ>
</xzufi:sprachversion>
```

Validator-Id	VAL0042
Beschreibung	Die "geokodierung" der Anschrift einer Organisationseinheit muss ein Element "Point" mit Attribut "srsName" enthalten. "srsName" muss dem Format "EPSG:[0-9]*" entsprechen. Das Attribut "axisLabels" muss die Information über Koordinatenachsen enthalten. Es werden die Achsen aus den Wertebereichen {northing, easting} und {latitude, longitude} unterstützt. Die Koordinaten müssen in Deutschland liegen.
Begründung	**Datenqualitäts-Validator:** Wenn an einer Organisationseinheit übermittelte Geokoordinaten nicht dem verwendeten Standard entsprechen oder außerhalb Deutschlands liegen, weist das auf einen redaktionellen Fehler hin. Entsprechend wird ein Hinweis ausgegeben.

Beispiel (Falsch)

Information über Koordinatenachsen wurde nicht übertragen. Attribut *axisLabels* fehlt.

```
<xzufi:geokodierung>
    <gml:Point srsName="EPSG:4326" xmlns:gml="http://www.opengis.net/gml/3.2">
        <gml:pos>9.768412 52.373543</gml:pos>
    </gml:Point>
</xzufi:geokodierung>
```

Beispiel (Richtig)

```
<xzufi:geokodierung>
    <gml:Point srsName="EPSG:4326" xmlns:gml="http://www.opengis.net/gml/3.2">
        <gml:pos axisLabels="longitude latitude">9.768412 52.373543</gml:pos>
    </gml:Point>
</xzufi:geokodierung>
```

Validator-Id	VAL0041
Beschreibung	Alle Module, die mehrsprachig geliefert werden, müssen das Attribut "languageCode" enthalten. Dieses darf nicht leer sein.
Begründung	**Standard- und Datenqualitäts-Validator:** Der XZuFi-Standard sieht vor, dass Geschäftsobjekte mehrere Sprachversionen enthalten können. Wenn mehr als nur eine (default: Deutsch) Sprachversion geliefert wird, muss an jedem Modul die Sprachversion (languageCode) definiert sein, um Eindeutigkeit für alle Konsumenten zu gewährleisten. Fehlt dies, lehnt der Validator entsprechend ab.

Beispiel (Falsch)

Das Attribut "languageCode" fehlt. Es ist nicht möglich, zu erkennen, um welche Sprache es sich handelt.

```
<xzufi:modulText>
    <xzufi:inhalt>Untersuchungen auf Zahn-, Mund- und Kieferkrankheiten</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Beispiel (Richtig)

Attribut *languageCode* ist vorhanden.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
    <xzufi:inhalt languageCode="en">Examinations for dental, oral and maxillofacial
diseases</xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Validator-Id	VAL0040
Beschreibung	Jedes an das PVOG angebundene Redaktionssysteme bekommt vor der Anbindung einen eindeutigen Identifikator 'RedaktionssystemID' zugeordnet. Diese RedaktionssystemID muss an das PVOG im Attribut 'SchemeAgencyId' der XZuFi-Elemente immer übertragen werden. Die RedaktionssystemID ist Teil des Geschäftsobjektidentifikators in PVOG. Der Validator prüft bei Leistungen, Onlinediensten, Organisationseinheiten, Spezialisierungen, Formularen und Zuständigkeiten, ob die SchemeAgencyId mit der vorgegebenen RedaktionssystemID übereinstimmt.
Begründung	**Technischer Validator:** Damit Geschäftsobjekte fehlerfrei verarbeitet und bereitgestellt werden können, brauchen sie einen eineindutigen Identifikator (ID). Da IDs von Redaktionssystemen individuell vergeben werden können, benötigt das PVOG zusätzlich den Identifikator des Redaktionssystems (SchemeAgencyID). Fehlt diese in einem übertragenen Objekt oder ist sie inkorrekt, lehnt der Validator dies entsprechend ab.

Beispiel (Falsch)

Die RedaktionssystemID ist *L000000*. Das Element *id* der Leistung enthält das Attribut *schemeAgencyID* mit dem Wert *L111111*.

```
<xzufi:leistung>
    <ns2:id schemeAgencyID="L111111">6006189</ns2:id>
...
</xzufi:leistung>
```

Beispiele (Richtig)

Die RedaktionssystemID ist *L000000*. Das Element *id* der Leistung enthält das Attribut *schemeAgencyID* mit dem Wert *L000000*.

```
<xzufi:leistung>
    <ns2:id schemeAgencyID="L000000">6006189</ns2:id>
...
</xzufi:leistung>
```

Die RedaktionssystemID ist *L000000*. Das Element *id* der Organisationseinheit enthält das Attribut

schemeAgencyID mit dem Wert *L000000*.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <xzufi:id schemeAgencyID="L000000">6003787</xzufi:id>
...
</xzufi:organisationseinheit>
```

Die RedaktionssystemID ist *L000000*. Das Element *leistungID* enthält das Attribut *schemeAgencyID* mit dem Wert *L000000*.

```
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">583</xzufi:leistungID>
...
</xzufi:zustaendigkeit>
```

Validator-Id	VAL0039
Beschreibung	XZuFi-Dokumente dürfen keine String-Elemente mit leerem Inhalt haben. Der Validator prüft bei allen Elementen vom Typ 'String.Localized', 'String.Latin', 'string' und 'normalizedString', ob mindestens ein Zeichen enthalten ist. Eine leere Zeichenkette wird als leerer Inhalt interpretiert. Eine Ausnahme stellt das Element 'Spezialisierung' dar. Durch eine Spezialisierung besteht weiterhin die Möglichkeit Inhalte auf 'leer' zu setzen.
Begründung	**Datenqualitäts-Validator:** Es ist Konvention, dass Texte (übertragen im String-Format) nur geliefert werden, wenn sie Inhalte enthalten. Das reduziert den Aufwand der Validierung. Werden leere Strings geliefert, wird es zwar verarbeitet, aber der Validator weist auf ein mögliches Versehen hin.

Beispiel (Falsch)

Das Element *inhalt* ist leer.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de"></xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de"/>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Das Element *inhalt* enthält nur Leerzeichen.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">   </xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
```

```
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Beispiel (Richtig)

Das Element *inhalt* enthält Text.

```
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
```

Das Element *inhalt* enthält Text. Der Text wird aber durch eine Spezialisierung mit leerem Inhalt ersetzt.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Untersuchungen auf Zahn-, Mund- und
Kieferkrankheiten</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>

...

<xzufi:spezialisierung>
...
    <xzufi:modulSpezialisiert>
        <xzufi:modul xsi:type="xzufi:Textmodul"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <xzufi:positionDarstellung>1</xzufi:positionDarstellung>
            <xzufi:inhalt languageCode="de"></xzufi:inhalt>
            <xzufi:leikaTextmodul
listURI="urn:de:xzufi:codeliste:leistungstextmodulleika" listVersionID="20190605">
                <code>02</code>
            </xzufi:leikaTextmodul>
        </xzufi:modul>
        <xzufi:ersetzung>true</xzufi:ersetzung>
    </xzufi:modulSpezialisiert>
<xzufi:spezialisierung>
```

Validator-Id	VAL0038
Beschreibung	Die Redaktionssysteme dürfen nur Leistungen von einem bestimmten Typ an das PVOG liefern. Welche Typen geliefert werden dürfen, ist im Sammlerdienstkonzept festgelegt. Der Validator prüft das Element 'typisierung' an der Leistung, ob dessen Werte im zulässigen Wertebereich liegt.
Begründung	**Technischer, Standard- und Datenqualitäts-Validator:** Die Typisierungen von Leistungen regeln ihre Schreiberechte, so liegen manche Leistungen im Verantwotungsbereich des Bundes, der Länder oder von Kommunen. Diese Verantwortlichkeiten sind gemeinschaftlich definiert worden. Der Validator lehnt Leistungen ab, die von nicht autorisierter Stelle (Datenquelle passt nicht zu Typisierung) oder mit falscher Typisierung (Typisierung der Leistung entspricht nicht dem LeiKa) übermittelt werden.

Beispiel (Falsch)

Eine Landesredaktion liefert eine Leistung vom Typ '1' = *Regelungs- und Vollzugskompetenz auf Bundesebene* im Element *typisierung*. Das ist nicht erlaubt. Leistungen vom Typ '1' werden vom Bundesredaktionssystem geliefert.

```
<xzufi:leistung>
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>1</code>
    </xzufi:typisierung>
...
</xzufi:leistung>
```

Beispiel (Richtig)

Eine Landesredaktion liefert eine Leistung vom Typ '4' = *Regelungskompetenz auf Landesebene* im Element *typisierung*. Das ist erlaubt.

```
<xzufi:leistung>
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>4</code>
    </xzufi:typisierung>
...
```

```
</xzufi:leistung>
```

Validator-Id	VAL0037
Beschreibung	Leistungen müssen mindestens eine Typisierung enthalten. Der Validator prüft das Element *typisierung* an der Leistung, ob das Element *code* nicht leer ist.
Begründung	**Technischer und Standard-Validator:** Die korrekte Typisierung ist notwendig für die Validierung im PVOG. Die Konventionen dazu werden gemeinschaftlich im Portalverbund getroffen.

Beispiel (Falsch)

Das Element *code* ist leer.

```
<xzufi:leistung>
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code></code>
    </xzufi:typisierung>
...
</xzufi:leistung>
```

```
<xzufi:leistung>
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code/>
    </xzufi:typisierung>
...
</xzufi:leistung>
```

Das Element *code* enthält nur die Leerzeichen.

```
<xzufi:leistung>
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>  </code>
    </xzufi:typisierung>
...
</xzufi:leistung>
```

Beispiel (Richtig)

Das Element *code* ist nicht leer.

```
<xzufi:leistung>
...
    <xzufi:typisierung listURI="urn:de:fim:leika:typisierung"
listVersionID="20201204">
        <code>4</code>
    </xzufi:typisierung>
...
</xzufi:leistung>
```

Validator-Id	VAL0036
Beschreibung	Leistungen müssen ein nicht-leeres Textmodul mit Code 02 oder 03 enthalten. Enthält eine Leistung kein Textmodul mit Code 06 (oder nur eines mit leerem Inhalt), muss sie mindestens eine Spezialisierung enthalten und alle ihre Spezialisierungen müssen ein Textmodul mit Code 06 liefern. Generell dürfen Spezialisierungen keine Textmodule mit Code 02, 03 oder 06 mit leerem Inhalt haben, wenn die Spezialisierung die Inhalte ersetzt und nicht ergänzt. Für XZuFi 2.3.1 gelten diese Regeln je angegebenem Zugangskanal. Die Prüfung erfolgt pro in der Leistung definierte Sprachversion. Für alle Sprachversionen außer Deutsch wird eine Meldung ausgegeben. Dadurch werden die Importe nicht blockiert, wenn die fremdsprachigen Texte fehlen.
Begründung	**Datenqualitäts- Validator:** Bezeichnung (Code 02 oder 03) und Volltext (Code 06) sollen geliefert werden, um den Nutzenden der Daten (Bürgerinnen, Bürger, Unternehmen und nachnutzende Systeme) das inhaltliche Verständnis der Daten zu ermöglichen. Es wäre auch möglich, dass leere Textmodule durch nachfolgende Spezialisierungen gefüllt werden könnten, anstatt auf ihre Lieferung im Leistungsobjekt zu bestehen. Diese Änderung ist von der UAG Leistungen im Juli 2024 beauftragt worden.

Beispiel (Falsch)

Es ist kein Textmodul mit dem Code *06* vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
```

```
</xzufi:leistung>
```

Es ist kein Textmodul mit dem Code *02* oder mit dem Code *03* vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Beispiel (Richtig)

Es sind Textmodule mit dem Code *02* und *06* vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:positionDarstellung>3</xzufi:positionDarstellung>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Es sind Textmodule mit dem Code *03* und *06* vorhanden.

```
<xzufi:leistung>
```

```
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>03</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:positionDarstellung>3</xzufi:positionDarstellung>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Es sind Textmodule mit dem Code *02*, *03* und *06* vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>03</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
```

```
    </xzufi:modulText>
...
</xzufi:leistung>
```

Beispiel (XZuFi 2.3.1, Falsch)

Es ist kein Textmodul mit dem Code *06* für Zugangskanal 100 vorhanden, obwohl Code *02* übergreifend (000) deklariert ist.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:zugangskanal>
            <code>000</code>
        </xzufi:zugangskanal>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:zugangskanal>
            <code>200</code>
        </xzufi:zugangskanal>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Beispiel (XZuFi 2.3.1, Richtig)

Es ist sind Textmodule mit dem Code *06* für Zugangskanäle 100 und 200 vorhanden, und Code *02* ist übergreifend (000) deklariert.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:zugangskanal>
            <code>000</code>
        </xzufi:zugangskanal>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
```

```
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:zugangskanal>
            <code>100</code>
        </xzufi:zugangskanal>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern. Verunreinigungen können online gemeldet werden.</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:zugangskanal>
            <code>200</code>
        </xzufi:zugangskanal>
        <xzufi:inhalt languageCode="de">Wasserversorgungsunternehmen sind dafür
verantwortlich, den Verbraucherinnen und Verbrauchern einwandfreies Trinkwasser zu
liefern</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>06</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Beispiel (Mehrsprahig, Falsch)

Es ist kein Textmodul mit dem Code *02* für die englische Sprache vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
    <xzufi:sprachversion>
        <xzufi:languageCode>de</xzufi:languageCode>
    </xzufi:sprachversion>
    <xzufi:sprachversion>
```

```
        <xzufi:languageCode>en</xzufi:languageCode>
    </xzufi:sprachversion>
</xzufi:leistung>
```

Beispiel (Mehrsprahig, Richtig)

Es ist ein Textmodul mit dem Code *02* für die englische und deutsche Sprache vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:inhalt languageCode="en">Drinking water - report
contamination</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
    <xzufi:sprachversion>
        <xzufi:languageCode>de</xzufi:languageCode>
    </xzufi:sprachversion>
    <xzufi:sprachversion>
        <xzufi:languageCode>en</xzufi:languageCode>
    </xzufi:sprachversion>
</xzufi:leistung>
```

Beispiel (Volltextmodul, Falsch)

Es ist kein Volltextmodul mit dem Code *06* an der Leistung und Spezialisierung vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:inhalt languageCode="en">Drinking water - report
contamination</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
    <xzufi:sprachversion>
        <xzufi:languageCode>de</xzufi:languageCode>
    </xzufi:sprachversion>
    <xzufi:sprachversion>
```

```
        <xzufi:languageCode>en</xzufi:languageCode>
    </xzufi:sprachversion>
</xzufi:leistung>
```

Beispiel (Volltextmodul, Richtig)

Es ist kein Volltextmodul mit dem Code *06* an der Leistung aber in einer Spezialisierung vorhanden. Alle Sprachversionen der Leistung sind auch in der Spezialisierung vorhanden.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen
melden</xzufi:inhalt>
        <xzufi:inhalt languageCode="en">Drinking water - report
contamination</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
    <xzufi:sprachversion>
        <xzufi:languageCode>de</xzufi:languageCode>
    </xzufi:sprachversion>
    <xzufi:sprachversion>
        <xzufi:languageCode>en</xzufi:languageCode>
    </xzufi:sprachversion>
</xzufi:leistung>

<xzufi:spezialisierung>
    <xzufi:idLeistung>8965820</xzufi:idLeistung>
    <xzufi:idGebiet>064390010010</xzufi:idGebiet>
    <xzufi:modulSpezialisiert>
        <xzufi:modul xsi:type="xzufi:Textmodul">
            <xzufi:inhalt languageCode="de">Trinkwasser - Verunreinigungen melden
Darmstadt</xzufi:inhalt>
            <xzufi:inhalt languageCode="en">Drinking water - report contamination
Darmstadt</xzufi:inhalt>
            <xzufi:leikaTextmodul
                listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
                listVersionID="20150601">
                <code>06</code>
            </xzufi:leikaTextmodul>
        </xzufi:modul>
        <xzufi:ersetzung>false</xzufi:ersetzung>
    </xzufi:modulSpezialisiert>
</xzufi:spezialisierung>
```

Validator-Id	VAL0035
Beschreibung	Für eine Leistung muss mindestens eine zuständige Stelle bzw. ein Ansprechpunkt existieren oder die Leistung muss über einen Onlinedienst angeboten werden. Der Validator prüft, ob mindestens eine Zuständigkeit im Dokument existiert, welche die Leistung mit einer Organisationseinheit oder einem Onlinedienst verknüpft.
Begründung	**Datenqualitäts-Validator:** Ziel des Portalverbund Onlinegateways ist es unter anderem und im besonderen, Informationen über Verwaltungsleistungen bereitzustellen. Dafür ist erforderlich, dass Leistungen, Organisationseinheiten und/oder Onlinedienste öffentlich angezeigt werden können. Fehlen diese Daten, hat z.B. eine Leistung in einem Gebiet keine zuständige Organisationseinheit oder Onlinedienst, sollte sie in diesem Gebiet auch nicht angezeigt bzw. zuständig sein. Der Validator prüft also, ob die Zuständigkeit mit Leistung und Interaktionsmöglichkeit für die Bürger komplett ist. Ist das nicht so, informiert der Validator mit einer Hinweismeldung.

Beispiele (Falsch)

Im Dokument gibt es weder eine Organisationseinheit noch einen Onlinedienst, welche die Leistung mit ID *280* in einer Zuständigkeit referenzieren.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L000000">280</xzufi:id>
    ...
    </xzufi:leistung>
</xzufi:schreibe>
```

Im Dokument gibt es eine Organisationseinheit, die eine andere Leistung mit ID *350* über die Zuständigkeit referenziert.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L000000">280</xzufi:id>
    ...
    </xzufi:leistung>
</xzufi:schreibe>
```

```
<xzufi:schreibe>
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">350</xzufi:leistungID>
            ...
        </xzufi:zustaendigkeit>
    </xzufi:organisationseinheit>
</xzufi:schreibe>
```

Beispiele (Richtig)

Die Leistung mit ID *280* wird in der Zuständigkeit eines Onlinedienstes im Element *leistungID* referenziert.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L000000">280</xzufi:id>
    ...
    </xzufi:leistung>
</xzufi:schreibe>

...

<xzufi:schreibe>
    <xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">280</xzufi:leistungID>
            ...
        </xzufi:zustaendigkeit>
    </xzufi:onlinedienst>
</xzufi:schreibe>
```

Die Leistung mit ID *280* wird in der Zuständigkeit einer Organisationseinheit im Element *leistungID* referenziert.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L000000">280</xzufi:id>
    ...
    </xzufi:leistung>
</xzufi:schreibe>

...

<xzufi:schreibe>
```

```
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">280</xzufi:leistungID>
            ...
        </xzufi:zustaendigkeit>
    </xzufi:organisationseinheit>
</xzufi:schreibe>
```

Validator-Id	VAL0034
Beschreibung	Die Anschrift einer Organisationseinheit muss einen Typ, eine PLZ und einen Ort enthalten, welche nicht leer sein dürfen. Wenn der Typ '003' ist, dann muss ein Postfach enthalten sein.
Begründung	**Datenqualitäts-Validator:** Der Geschäftsobjekttyp Organisationseinheit hat u.a. den Zweck, Informationen für die Organisationseinheit für Bürger und Bürgerinnen bereitzustellen. Wenn die Adresse einer Organisationseinheit unvollständig ist, erfüllt sie folglich den Zweck des Geschäftsobjekts nicht komplett und wird entsprechend vom Validator abgelehnt.

Beispiele (Falsch)

Es fehlen die Elemente *postleitzahl* und *ort*.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <xzufi:anschrift>
        <xzufi:typ listURI="urn:xoev-de:fim:codeliste:xzufi.anschrifttyp"
listVersionID="2.0">
            <code>001</code>
        </xzufi:typ>
        <xzufi:strasse>Musterstrasse</xzufi:strasse>
        <xzufi:hausnummer>1</xzufi:hausnummer>
    </xzufi:anschrift>
...
</xzufi:organisationseinheit>
```

Das Element *ort* ist leer.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <xzufi:anschrift>
        <xzufi:typ listURI="urn:xoev-de:fim:codeliste:xzufi.anschrifttyp"
listVersionID="2.0">
            <code>001</code>
        </xzufi:typ>
        <xzufi:strasse>Musterstrasse</xzufi:strasse>
        <xzufi:hausnummer>1</xzufi:hausnummer>
        <xzufi:postleitzahl>12345</xzufi:postleitzahl>
        <xzufi:ort></xzufi:ort>
    </xzufi:anschrift>
```

```
...
</xzufi:organisationseinheit>
```

Beispiele (Richtig)

Die Elemente *postleitzahl* und *ort* sind vorhanden und sind nicht leer.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <xzufi:anschrift>
        <xzufi:typ listURI="urn:xoev-de:fim:codeliste:xzufi.anschrifttyp"
listVersionID="2.0">
            <code>001</code>
        </xzufi:typ>
        <xzufi:strasse>Musterstrasse</xzufi:strasse>
        <xzufi:hausnummer>1</xzufi:hausnummer>
        <xzufi:postleitzahl>12345</xzufi:postleitzahl>
        <xzufi:ort>Musterort</xzufi:ort>
    </xzufi:anschrift>
...
</xzufi:organisationseinheit>
```

Die Adresse vom Typ *003* = *Postfach* enthält nicht leeres Element *postfach*

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <xzufi:anschrift>
        <xzufi:typ listURI="urn:xoev-de:fim:codeliste:xzufi.anschrifttyp"
listVersionID="2.0">
            <code>003</code>
        </xzufi:typ>
        <xzufi:postfach>815</xzufi:postfach>
        <xzufi:postleitzahl>12345</xzufi:postleitzahl>
        <xzufi:ort>Musterort</xzufi:ort>
    </xzufi:anschrift>
...
</xzufi:organisationseinheit>
```

Validator-Id	VAL0033
Beschreibung	Das Format von URIs und URLs muss dem 'RFC 2396' entsprechen. Der Validator überprüft, ob alle Elemente/Attribute vom Typ 'xs:anyURI' gemäß 'RFC 2396' syntaktisch korrekte URI sind. Zusätzlich müssen die Elemente 'linkKostenbildung' aus einem Kostenmodul und 'link' aus dem Onlinedienst syntaktisch korrekte URLs gemäß 'RFC 2396' enthalten.
Begründung	**Technischer Validator:** Es ist Konvention, sich an etablierte Standards zu halten, z.B. String.Localized. Der Standard für URL/URI stellt sicher, dass Hyperlinks verarbeitet und funktionstüchtig bereitgestellt werden können. Verstößt eine URI gegen den Standard, lehnt der Validator ab, um "kaputte Links" zu vermeiden.

Beispiel (Falsch)

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <xzufi:link languageCode="de">
        ...
        <xzufi:link>musterdomain.de/web/onlinedienst280</xzufi:link>
    </xzufi:link>
...
</xzufi:onlinedienst>
```

Beispiel (Richtig)

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <xzufi:link languageCode="de">
        ...
        <xzufi:link>https://www.musterdomain.de/web/onlinedienst280</xzufi:link>
    </xzufi:link>
...
</xzufi:onlinedienst>
```

Validator-Id	VAL0032
Beschreibung	Ein Element vom Type *Code* muss die Attribute *listURI* und *listVersionID* enthalten. Das Element *code* darf nicht leer sein.
Begründung	**Technischer, Standard- und Datenqualitäts-Validator:** Codelisten werden verwendet, um die Menge möglicher Auswahloptionen sinnvoll einzuschränken. Z.B. sieht die Codeliste für Textmodule vor, dass man sie für die Volltext-Beschreibung einer Leistung benutzen kann, oder ihren Titel, allerdings nicht für Link zum Onlinedienst (denn dafür gibt es den Geschäftsobjekttyp Onlinedienst). Kann der Validator nicht feststellen, dass bzw. welche gültige Codeliste verwendet wird, lehnt er die Daten ab.

Beispiel (Falsch)

Das Attribut *listURI* fehlt.
Das Element *leikaTextmodul* ist vom Typ *Code.Leistungstextmodul*. Obwohl das Attribut *listURI* in XZuFi-Standard optional ist, ist dieses Attribut für PVOG verpflichtend.

```
<xzufi:modulText>
...
    <xzufi:leikaTextmodul listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Das Attribut *listURI* ist leer.

```
<xzufi:modulText>
...
    <xzufi:leikaTextmodul listURI=" " listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Das Attribut *listVersionID* ist leer.

```
<xzufi:modulText>
...
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="">
        <code>02</code>
```

```
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Das Element *code* ist leer.

```
<xzufi:modulText>
...
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code> </code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Beispiel (Richtig)

Die Attribute *listURI* und *listVersionID* sind vorhanden. Das Element *code* ist nicht leer.

```
<xzufi:modulText>
...
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Validator-Id	VAL0031
Beschreibung	Die ID von den Leistungen, Spezialisierungen, Organisationseinheiten, Onlinediensten oder Formularen muss eindeutig sein, d.h. in demselben Dokument dürfen keine zwei Geschäftsobjekte des gleichen Typs mit demselben Identifikator enthalten sein. Werden im Rahmen der Validierung eines Redaktionssystems mehrere Dokumente verarbeitet, wird zusätzlich geprüft, dass Spezialisierung-IDs dokumentenübergreifend eindeutig sind.
Begründung	**Technischer Validator:** Damit Geschäftsobjekte fehlerfrei verarbeitet und bereitgestellt werden können, brauchen sie einen eineindutigen Identifikator (ID). Befinden sich in einem Dokument zwei Objekte mit derselben ID, lehnt der Validator dies entsprechend ab.

Beispiel (Falsch)

Das Dokument enthält das Geschäftsobjekt *Organisationseinheit* mit dem Identifikator *600* doppelt.

```
<xzufi:schreibe>
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <xzufi:id schemeAgencyID="L000000">600</xzufi:id>
        ...
    </xzufi:organisationseinheit>
</xzufi:schreibe>
...
<xzufi:schreibe>
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <xzufi:id schemeAgencyID="L000000">600</xzufi:id>
        ...
    </xzufi:organisationseinheit>
</xzufi:schreibe>
```

Beispiel (Richtig)

Die Organisationseinheit mit ID *600* ist im Dokument nur einmal vorhanden.

```
<xzufi:schreibe>
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <xzufi:id schemeAgencyID="L000000">600</xzufi:id>
        ...
```

```
    </xzufi:organisationseinheit>
</xzufi:schreibe>
```

Validator-Id	VAL0030
Beschreibung	Bei den Daten in XZuFi-2.2-Format dürfen zwei Textmodule einer Leistung nicht denselben Code haben. Bei den Daten in XZuFi 2.3.1-Format dürfen zwei Textmodule einer Leistung den gleichen Code haben, wenn diese verschiedene Zugangskanäle besitzen. Eine Kombination aus 'Online/Offline = 000' und 'Online = 100' bzw. 'Offline = 200' ist nicht erlaubt, da Textmodule eine Überschneidung beim Zugangskanal haben.
Begründung	**Datenqualitäts- Validator:** Wenn eine Leistung Textmodule mit demselben Code enthält, legt das einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt. Andernfalls könnte es konkurrierende Volltextbeschreibungen (Code 006) einer Leistung geben, oder konkurrierende Titel (Code 001). Im Zuge der Einführung von XZuFi 2.3.1 wird ein Anwendungsfall für doppelte Textmodule geschaffen: unterschiedliche Gebühren o.ä. für eine Leistung abhängig davon ob sie online oder in Person im Amt in Anspruch genommen wird.

Beispiel XZuFi 2.2 (Falsch)

Das Textmodul mit dem Code *02* kommt zweimal vor.

```
<xzufi:leistung>
...
    <xzufi:modulText>
    ...
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
    ...
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Beispiel XZuFi 2.2 (Richtig)

Das Textmodul mit dem Code *02* kommt einmal vor.

```
<xzufi:modulText>
...
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Beispiel XZuFi 2.3.1 (Falsch)

Das Textmodul mit dem Code *02* kommt zweimal mit Zugangskanal-Überschneidung vor. Der Zugangskanal 000 (Online/Offline) beinhaltet den Zugangskanal 100 (Online).

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>000</code>
        </xzufi:zugangskanal>
    ...
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>100</code>
        </xzufi:zugangskanal>
    ...
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
            <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Das Textmodul mit dem Code 02 kommt zweimal mit gleichem Zugangskanal 200 (Offline) vor.

```
<xzufi:leistung>
...
    <xzufi:modulText>
```

```
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>200</code>
        </xzufi:zugangskanal>
    ...
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
           <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
    <xzufi:modulText>
        <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>200</code>
        </xzufi:zugangskanal>
    ...
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
           <code>02</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
</xzufi:leistung>
```

Beispiel XZuFi 2.3.1 (Richtig)

Das Textmodul mit dem Code *02* kommt einmal pro Zugangskanal 000 (Online/Offline) vor.

```
<xzufi:modulText>
     <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>000</code>
     </xzufi:zugangskanal>
...
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Das Textmodul mit dem Code *02* kommt einmal pro Zugangskanal 100 (Online) und 200 (Offline) vor.

```
<xzufi:modulText>
     <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>100</code>
     </xzufi:zugangskanal>
...
```

```
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
<xzufi:modulText>
     <xzufi:zugangskanal listURI="urn:xoev-de:fim:codeliste:xzufi.zugangskanal"
listVersionID="1">
          <code>200</code>
     </xzufi:zugangskanal>
...
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20201201">
        <code>02</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Validator-Id	VAL0029
Beschreibung	Zwei Zuständigkeiten einer Organisationseinheit dürfen nicht auf dieselbe Menge von Gebiets-Ids (ARS) und auf dieselbe Leistung verweisen. Die Listen der GebietIDs und der LeistungIDs in zwei Zuständigkeiten müssen disjunkt sein.
Begründung	**Technischer Validator:** Es ist Konvention, dass eine Organisationseinheit pro Gebiet alle Zuständigkeiten in einem Objekt abbildet. Wenn eine Organisationseinheit aber zwei Zuständigkeiten mit identischen Gebieten enthielte, könnte es zu Dopplungen kommen, die zu referenziellen Inkonsistenzen und faktischem Datenverlust oder -verfälschung führen würden. Um das zu vermeiden, lehnt der Validator Organisationseinheiten mit gebiets-identischen Zuständigkeiten ab.

Beispiel (Falsch)

Die GebietID *090000000000* kommt in zwei Zuständigkeiten vor.

```
...
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
...
```

Beispiel (Richtig)

Die GebietID *090000000000* kommt in einer separaten Zuständigkeit nur einmal vor.

```
...
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
```

```
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
...
```

Validator-Id	VAL0028
Beschreibung	Alle in der Zuständigkeit referenzierten Leistungen müssen im selben Dokument existieren.
Begründung	**Datenqualitäts-Validator:** Wenn eine Zuständigkeit eine referenzierte Leistung nicht enthält, kann die referenzielle Integrität nicht geprüft werden. Es könnte also zu Dopplungen kommen, wenn dieselbe Leistung mit anderen Inhalten von anderen Zuständigkeiten übertragen würde. Um dem vorzubeugen, lehnt der Validator Zuständigkeiten mit "leeren Referenzen" ab.

Beispiel (Falsch)

Die Zuständigkeit referenziert eine Leistung mit ID *4815*. Diese Leistung ist im Dokument nicht enthalten. Es gibt eine Leistung mit ID *3815*.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L00000">3815</xzufi:id>
...
    <xzufi:leistung>
<xzufi:schreibe>
<xzufi:schreibe>
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
        ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">4815</xzufi:leistungID>
            <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
        </xzufi:zustaendigkeit>
    </xzufi:organisationseinheit>
</xzufi:schreibe>
```

Beispiel (Richtig)

Die Zuständigkeit referenziert eine Leistung mit ID *3815*. Die Leistung mit ID *3815* ist auch im Dokument enthalten.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L00000">3815</xzufi:id>
...
    <xzufi:leistung>
<xzufi:schreibe>
<xzufi:schreibe>
    <xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
        ...
```

```
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
            <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
        </xzufi:zustaendigkeit>
    </xzufi:organisationseinheit>
</xzufi:schreibe>
```

Validator-Id	VAL0027
Beschreibung	Die Geschäftsobjekte Leistung, Organisationseinheit, Spezialisierung, Onlinedienst und Formular müssen eine gültige ID und SchemeAgencyId besitzen. Der Validator prüft, ob das Element *id* mit dem Attribut *schemeAgencyID* existiert. Das Element und das Attribut dürfen nicht leer sein. Die *id* und *schemeAgencyID* dürfen nur Buchstaben, Ziffern, Bindestriche und Unterstriche enthalten.
Begründung	**Technischer und Datenqualitäts-Validator:** Die ID ist der notwendige Identifikator jedes Objekts. Die SchemeAgencyId ist der notwendige Identifikator für den jeweiligen Datenbestand (z.B. Bund, Land ABC, Redaktionssystem XYZ). Ohne diese eindeutigen und gültigen IDs können die Daten nicht referenziert und bereitgestellt werden. Daher käme es zu faktischem Datenverlust.

Beispiele (Falsch)

Das Attribut *schemeAgencyID* fehlt.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id>327954_23-78</xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Das Attribut *schemeAgencyID* ist leer.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID=" ">327954_23-78</xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Die *id* ist leer.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">  </xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Die *id* und *schemeAgencyID* enthalten unerlaubte Zeichen Punkt . und Raute #

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L.000#000">327954#23.78</xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Beispiele (Richtig)

Die *id* enthält das Attribut *schemeAgencyID*. Die *id* und *schemeAgencyID* enthalten nur Buchstaben und Ziffern.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">327954</xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Die *id* enthält das Attribut *schemeAgencyID*. Die *id* und *schemeAgencyID* enthalten nur Buchstaben, Ziffern, Bindestrich und Unterstrich.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">L327954_23-78</xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Validator-Id	VAL0026
Beschreibung	Der ARS-Schlüssel im Element „gueltigkeitGebietID“ einer Leistung muss 12-stellig sein.
Begründung	**Technischer Validator:** Wenn eine Leistung keine "gueltigkeitGebietID" enthält, legt das einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt. Der XZuFi-Standard 2.2 definiert auf Seite 81 "gueltigkeitGebietID" wie folgt: "Gibt an, für welche Gebiete der Inhalt der Leistung gültig ist. Für bundesland-weite Versionen einer Leistung die ID des Gebiets des Bundeslandes. Für untergeordnete Gebiete (z.B.Städte) dann die IDs der entsprechenden Gebiete. Es soll vermieden werden, hier kein Gebiet anzugeben. Wenn dies der Fall sein sollte, wird eine globale Gültigkeit in einem zu vereinbarendem Kontext definiert. Wichtig: Es handelt sich hierbei nicht um eine Definition der Zuständigkeit!"

Beispiel (Falsch)

Wert des Elementes *gueltigkeitGebietID* ist 2-stellig.

```
<xzufi:leistung>
    ...
    <xzufi:gueltigkeitGebietID schemeAgencyID="L000000">11</xzufi:gueltigkeitGebietID>
    ...
<xzufi:leistung>
```

Beispiel (Richtig)

Wert des Elementes *gueltigkeitGebietID* ist 12-stellig.

```
<xzufi:leistung>
    ...
    <xzufi:gueltigkeitGebietID
schemeAgencyID="L000000">110000000000</xzufi:gueltigkeitGebietID>
    ...
<xzufi:leistung>
```

Validator-Id	VAL0025
Beschreibung	Die bereitgestellten Daten dürfen keine zeitlichen Widersprüche enthalten. Der Beginn eines Gültigkeitszeitraums im Element „gueltigkeit“ darf nicht nach dem Ende des Gültigkeitszeitraums liegen. Das Ende eines Gültigkeitszeitraums im Element „gueltigkeit“ darf nicht in der Vergangenheit liegen. Es wird eine Warnung produziert, falls der Gültigkeitszeitraum in weniger als eine Woche abläuft. Elemente ohne festgelegte Gültigkeitszeiträume gelten ohne Einschränkung. Es werden die Geschäftsobjekte Leistung inklusive Spezialisierungen, Textmodule, Organisationseinheiten, Onlinedienste und Formulare inklusive Zuständigkeiten geprüft.+ Die Gültigkeit der Pflichttextmodule mit dem Code *02* oder *03* und *06* dürfen nicht vor der Leistung ablaufen. Ansonsten kann die Situation entstehen, dass der Leistung die Pflichtinformationen fehlen.
Begründung	**Datenqualitäts- Validator:** Um sicher zu stellen, dass Leistungen nicht durch Ablauf ihrer Gültigkeit aus dem Datenbestand fallen, prüft diese Validierungsregel die Ablaufdaten. Ohne diese Prüfung könnte es zu "unsichtbaren" Zuständigkeiten kommen, da die referenzierten Leistungen, Onlinedienste und Organisationseinheiten nicht im aktiven Datenbestand des Suchdienstes (API oder Portal) gefunden werden könnten.

Beispiel (Falsch)

Beginndatum liegt nach dem Enddatum.

```
<xzufi:leistung>
...
    <xzufi:gueltigkeit>
        <xzufi:beginn>2050-06-16</xzufi:beginn>
        <xzufi:ende>2022-01-01</xzufi:ende>
    </xzufi:gueltigkeit>
</xzufi:leistung>
```

Die Gültigkeit des Textmoduls mit dem Code *03* läuft am 2022-12-30 ab, obwohl die Gültigkeit der Leistung erst am 2050-01-01 abläuft. Dies kann dazu führen, dass die Leistung ab dem 2022-12-30

die Information über den Namen der Leistung fehlt.

```
<xzufi:leistung>
...
    <xzufi:modulText>
        <xzufi:gueltigkeit>
            <xzufi:beginn>2022-06-16</xzufi:beginn>
            <xzufi:ende>2022-12-30</xzufi:ende>
        </xzufi:gueltigkeit>
        <xzufi:inhalt languageCode="de">Freiwillige Feuerwehr - Mitglied
werden</xzufi:inhalt>
        <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
listVersionID="20190605">
            <code>03</code>
        </xzufi:leikaTextmodul>
    </xzufi:modulText>
...
    <xzufi:gueltigkeit>
        <xzufi:beginn>2022-06-16</xzufi:beginn>
        <xzufi:ende>2050-01-01</xzufi:ende>
    </xzufi:gueltigkeit>
</xzufi:leistung>
```

Beispiele (Richtig)

Eine Leistung mit dem Gültigkeitszeitraum.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000">2010</xzufi:id>
...
    <xzufi:gueltigkeit>
        <xzufi:beginn>2022-06-16</xzufi:beginn>
        <xzufi:ende>2050-01-01</xzufi:ende>
    </xzufi:gueltigkeit>
</xzufi:leistung>
```

Eine Leistung ohne Gültigkeitszeitraum. Leistung gilt ohne zeitliche Einschränkung.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000">2010</xzufi:id>
...
</xzufi:leistung>
```

Validator-Id	VAL0024
Beschreibung	Eine Organisationseinheit darf keine Verweise auf die übergeordnete oder auf die untergeordnete Organisationseinheiten enthalten. Eine Organisationseinheit-Hierarchie ist nicht erlaubt.
Begründung	**Technischer Validator:** Der XZufi-Standard 2.2 sieht die Möglichkeit von Organisationseinheit-Hierarchien vor. Bei der ersten Umsetzung des PVOG wurden die Architekturentscheidung getroffen, die dieses Feature zugunsten erhöhter Effizienz hinten an zu stellen. Das Feature ist bisher nicht beauftragt, folglich erfüllen über- oder untergeordneteOrganisationseinheitID keinen Zweck und werden abgelehnt.

Beispiel (Falsch)

Die Organisationseinheit enthält Verweis auf eine übergeordnete *uebergeordneteOrganisationseinheitID* sowie auf eine untergeordnete *untergeordneteOrganisationseinheitID* Organisationseinheit.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">3001</xzufi:id>
    ...
    <xzufi:uebergeordneteOrganisationseinheitID
schemeAgencyID="L000000">3000</xzufi:uebergeordneteOrganisationseinheitID>

    <xzufi:untergeordneteOrganisationseinheitID
schemeAgencyID="L000000">3002</xzufi:untergeordneteOrganisationseinheitID>
...
<xzufi:organisationseinheit>
```

Beispiel (Richtig)

Es sind keine Elemente *uebergeordneteOrganisationseinheitID* bzw. *untergeordneteOrganisationseinheitID* enthalten und es ist keine Hierarchie abgebildet.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">3001</xzufi:id>
    ...
<xzufi:organisationseinheit>
```

Validator-Id	VAL0023
Beschreibung	Dokumente welche Sprachcodes außerhalb des definierten Wertebereichs enthalten, sind nicht valide. Der Inhalt des Attributes *languageCode* darf nicht leer sein und muss im Wertebereich {de, en} liegen.
Begründung	**Datenqualitäts- Validator:** Diese Validierungsregel stellt sicher, dass gelieferte Sprachen im vereinbarten Umfang liegen.

Beispiele (Falsch)

Der Wert *de-DE* liegt nicht im Wertebereich.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de-DE">Freiwillige Feuerwehr - Mitglied
werden</xzufi:inhalt>
    ...
</xzufi:modulText>
```

Der Wert *fr* liegt nicht im Wertebereich. Französisch *fr* ist momentan nicht erlaubt.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="fr">Freiwillige Feuerwehr - Mitglied
werden</xzufi:inhalt>
    ...
</xzufi:modulText>
```

Attribut *languageCode* ist leer.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode=" ">Freiwillige Feuerwehr - Mitglied
werden</xzufi:inhalt>
    ...
</xzufi:modulText>
```

Beispiel (Richtig)

Der Wert *de* kommt aus dem Wertebereich.

```
<xzufi:modulText>
    <xzufi:inhalt languageCode="de">Freiwillige Feuerwehr - Mitglied
werden</xzufi:inhalt>
    ...
</xzufi:modulText>
```

Validator-Id	VAL0022
Beschreibung	Jede Leistung muss mindestens einen Referenz-Leistungsschlüssel besitzen. Der Validator prüft das Element *referenzLeiKa*. Der Leistungsschlüssel einer Leistung muss entweder aus 14 Ziffern bestehen oder von der Form TMP_XX_XXXXXXXXXXXXX sein, wobei die X für Ziffern stehen.
Begründung	**Technischer und Datenqualitäts- Validator:** Wenn eine Leistung einen ungültigen Leistungsschlüssel aufweist, kann ein nachnutzendes System, z.B. ein Verwaltungsportal, keinen Zusammenhang mit den definierten Leistungen des LeiKa herstellen.

Beispiel (Falsch)

Der Leistungsschlüssel *99* ist nicht 14-stellig.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Der Leistungsschlüssel ist leer.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code> </code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Beispiele (Richtig)

Der 14-stellige FIM-Leistungsschlüssel *99038009223001*

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>99038009223001</code>
    </xzufi:referenzLeiKa>
...
```

```
<xzufi:leistung>
```

Der 14-stellige länderspezifische Leistungsschlüssel *11038009223001*.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>11038009223001</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Der temporäre Leistungsschlüssel *TMP_11_99038009223001* darf nicht 14-stellig sein.

```
<xzufi:leistung>
...
    <xzufi:referenzLeiKa listURI="urn:de:fim:leika:leistung" listVersionID="20210114">
        <code>TMP_11_99038009223001</code>
    </xzufi:referenzLeiKa>
...
<xzufi:leistung>
```

Validator-Id	VAL0021
Beschreibung	Eine Leistung muss ein Element „gueltigkeitGebietID“ mit dem ARS-Schlüssel enthalten.
Begründung	**Technischer Validator:** Wenn eine Leistung keine "gueltigkeitGebietID" enthält, legt das einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt. Der XZuFi-Standard 2.2 definiert auf Seite 81 "gueltigkeitGebietID" wie folgt: "Gibt an, für welche Gebiete der Inhalt der Leistung gültig ist. Für bundesland-weite Versionen einer Leistung die ID des Gebiets des Bundeslandes. Für untergeordnete Gebiete (z.B.Städte) dann die IDs der entsprechenden Gebiete. Es soll vermieden werden, hier kein Gebiet anzugeben. Wenn dies der Fall sein sollte, wird eine globale Gültigkeit in einem zu vereinbarendem Kontext definiert. Wichtig: Es handelt sich hierbei nicht um eine Definition der Zuständigkeit!"

Beispiel (Falsch)

Das Element *gueltigkeitGebietID* ist nicht vorhanden.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000" schemeID="OID">8964020</xzufi:id>
    ...
<xzufi:leistung>
```

Beispiel (Richtig)

Das Element *gueltigkeitGebietID* ist nicht vorhanden.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000" schemeID="OID">8964020</xzufi:id>
    ...
    <xzufi:gueltigkeitGebietID
schemeID="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">060000000000</xzufi:gueltigkeitGebietID>
    ...
<xzufi:leistung>
```

Validator-Id	VAL0020
Beschreibung	Zwei Zuständigkeiten eines Onlinedienstes dürfen nicht auf dieselbe Menge von Gebiets-Ids (ARS) derselben Leistung verweisen. Die Liste an GebietIDs und der LeistungIDs in den Zuständigkeiten müssen disjunkt sein.
Begründung	**Technischer und Datenqualitäts-Validator:** Wenn Zuständigkeiten am Onlinedienst exakt gleich sind, legt dies einen Fehler seitens der Redaktion nahe. Würden sie in dieser Form angenommen, könnten die entstandenen Duplikate zu inkonsistenten Referenzen führen. Um dies zu vermeiden, wird ein solcher Onlinedienst vom Validatordienst abgelehnt.

Beispiel (Falsch)

Die GebietID *090000000000* kommt in zwei Zuständigkeiten vor.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
        <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
        <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
```

Beispiel (Richtig)

Die GebietID *090000000000* kommt in einer separaten Zuständigkeit nur einmal vor.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
```

```
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
...
```

Validator-Id	VAL0019
Beschreibung	Ein Onlinedienst muss mindestens eine Zuständigkeit besitzen.
Begründung	**Datenqualitäts- Validator:** Wenn ein Onlinedienst keine Zuständigkeiten enthält, kann er nicht zur Zuständigkeitsfindung verwendet werden. Dies legt einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt.

Beispiel (Falsch)

Der Onlinedienst enthält keine Zuständigkeit.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
</xzufi:onlinedienst>
```

Beispiel (Richtig)

Der Onlinedienst enthält zwei Zuständigkeiten.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091620000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
```

Validator-Id	VAL0018
Beschreibung	Der ARS-Schlüssel im Element *gebietID* einer Zuständigkeit des Onlinedienstes müssen 12-stellig sein.
Begründung	**Technischer, Standard- und Datenqualitäts-Validator:** Wohlgeformte ARS-Schlüssel sind 12-stellig. Damit Zuständigkeiten auch von Onlinediensten funktionieren können, ist die standardkonforme Verwendung der ARS-Schlüssel notwendig. Sollten ARS-Schlüssel nicht korrekt sein, z.B. und u.a. wenn sie nicht 12 Stellen aufweisen, wird die Zuständigkeitsfindung in Mitleidenschaft gezogen. In der Folge werden Daten nicht verknüpft und stehen für die Verwendung effektiv nicht zur Verfügung, sie gehen verloren.

Beispiel (Falsch)

Der ARS-Schlüssel im Element *gebietID* ist *09* besteht aus nur zwei Ziffern.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">09</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
```

Beispiel (Richtig)

Der ARS-Schlüssel im Element *gebietID* ist *090000000000* 12-stellig.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">090000000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
```

Validator-Id	VAL0017
Beschreibung	Alle in der Zuständigkeit eines Onlinedienstes referenzierten Leistungen müssen im selben XZuFi-Dokument existieren. Somit wird sichergestellt, dass eine Zuständigkeit existierende Geschäftsobjekte verknüpft.
Begründung	**Datenqualitäts- Validator:** Diese Validierungsregel stellt sicher, dass kein Onlinedienst ohne entsprechende Verwaltungsleistung existieren kann. Sollte ein Onlinedienst Leistungen referenzieren, die aber nicht in derselben Lieferung enthalten sind, kann der Validatordienst deren Existenz nicht verifizieren und lehnt das Objekt deshalb ab.

Beispiel (Falsch)

Die Zuständigkeit referenziert eine Leistung mit ID *4815*. Diese Leistung ist in dem Dokument nicht enthalten. Stattdessen gibt es in dem Dokument eine Leistung mit der ID *3815*.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L00000">3815</xzufi:id>
...
    <xzufi:leistung>
<xzufi:schreibe>
<xzufi:schreibe>
    <xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">4815</xzufi:leistungID>
            ...
        </xzufi:zustaendigkeit>
    </xzufi:onlinedienst>
</xzufi:schreibe>
```

Beispiel (Richtig)

Die Zuständigkeit referenziert eine Leistung mit ID *3815*. Die Leistung mit ID *3815* ist auch im Dokument enthalten.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L00000">3815</xzufi:id>
...
    <xzufi:leistung>
<xzufi:schreibe>
```

```
<xzufi:schreibe>
    <xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        ...
        <xzufi:zustaendigkeit>
            <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
            ...
        </xzufi:zustaendigkeit>
    </xzufi:onlinedienst>
</xzufi:schreibe>
```

Validator-Id	VAL0016
Beschreibung	Das Redaktionssystem darf nur die Daten des Landes übertragen, für das es zuständig ist. Der ARS-Schlüssel im Element „gebietID“ an den Zuständigkeiten von Onlinediensten muss mit dem Präfix des zuständigen Landes beginnen. ARS-Schlüssel in Zuständigkeiten zu Leistungen im gemeinsamen Vollzug sind von dieser Prüfung ausgenommen. Solche Leistungen haben eine Kategorie mit der Bezeichnung "Leistung im gemeinsamen Vollzug".
Begründung	**Datenqualitäts- Validator:** Wenn ein Online-Dienst geliefert wird, dessen Zuständigkeitsgebiet keine Überschneidung mit dem der verknüpften Leistung aufweist, ist er faktisch leer bzw. "unsichtbar". Dies legt einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt.

Beispiel (Falsch)

Die Daten werden vom Redaktionssystem *Bayern* geliefert. Das Redaktionssystem *Bayern* ist nach Sammlerdienst-Konfiguration für die Gebiete, die mit dem Präfix *09* beginnen, zuständig. In den Daten wird versucht, eine Zuständigkeit des Landes Schleswig-Holstein *011630000000* mit dem Präfix *01* zu manipulieren.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">011630000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
```

Beispiel (Richtig)

Die Daten werden vom Redaktionssystem *Bayern* geliefert. Das Redaktionssystem *Bayern* ist nach Sammlerdienst-Konfiguration für die Gebiete, die mit dem Präfix *09* beginnen, zuständig. In den Daten wird eine Zuständigkeit *091630000000* mit dem Präfix *09* übermittelt.

```
<xzufi:onlinedienst xsi:type="xzufi:OnlinedienstErweitert"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    ...
    <xzufi:zustaendigkeit>
        ...
```

```
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">091630000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:onlinedienst>
```

Validator-Id	VAL0015
Beschreibung	Eine Organisationseinheit muss mindestens eine Zuständigkeit besitzen.
Begründung	**Datenqualitäts- Validator:** Wenn eine Organisationseinheit keine Zuständigkeiten enthält, kann sie nicht zur Zuständigkeitsfindung verwendet werden. Dies legt einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt.

Beispiel (Falsch)

Die Organisationseinheit enthält keine Zuständigkeit.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">3001</xzufi:id>
    ...
</xzufi:organisationseinheit>
```

Beispiel (Richtig)

Die Organisationseinheit enthält zwei Zuständigkeiten.

```
<xzufi:organisationseinheit xsi:type="xzufi:OrganisationseinheitErweitert">
    <xzufi:id schemeAgencyID="L000000">3001</xzufi:id>
    ...
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        ...
    </xzufi:zustaendigkeit>
    <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
        ...
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Validator-Id	VAL0014
Beschreibung	Die Gebiet-ID einer Spezialisierung muss in das Gebiet der zugehörigen Leistung fallen. Die ARS-Hierarchie muss berücksichtigt werden.
Begründung	**Datenqualitäts- Validator:** Wenn eine Spezialisierung Änderungen an einer Leistung vornimmt, die aber nicht in deren Zuständigkeitsgebiet fallen, ist die Spezialisierung faktisch leer und erfüllt den Zweck der Spezialisierung von allgemeinen Leistungen nicht. Dies legt einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt.

Beispiel (Falsch)

Die Leistung mit *gebietID* *010510000000* ist als eine Leistung des Kreises *51* definiert. In der Spezialisierung mit dem *gebietID* *010530090090* wird die Leistung für die Gemeinde *090* eines anderen Kreises *53* spezialisiert. In dem Kreis *53* wird die Leistung per Definition nicht angeboten.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000" schemeID="OID">8964020</xzufi:id>
    ...
    <xzufi:gueltigkeitGebietID
schemeID="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">010510000000</xzufi:gueltigkeitGebietID>
    ...
<xzufi:leistung>

...

<xzufi:spezialisierung>
    <xzufi:idLeistung schemeAgencyID="L000000"
schemeID="OID">8964020</xzufi:idLeistung>
    <xzufi:idGebiet
schemeID="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2018-07-31">010530090090</xzufi:idGebiet>
    ...
</xzufi:spezialisierung>
```

Beispiel (Richtig)

Die Leistung ist mit *gebietID* *010510000000* als eine Leistung des Kreises *51* definiert. In der Spezialisierung mit dem *gebietID* *010515175074* wird die Leistung für die Gemeinde *074* spezialisiert.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000" schemeID="OID">8964020</xzufi:id>
```

```
    ...
    <xzufi:gueltigkeitGebietID
schemeID="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">010510000000</xzufi:gueltigkeitGebietID>
    ...
<xzufi:leistung>

...

<xzufi:spezialisierung>
    <xzufi:idLeistung schemeAgencyID="L000000"
schemeID="OID">8964020</xzufi:idLeistung>
    <xzufi:idGebiet
schemeID="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2018-07-31">010515175074</xzufi:idGebiet>
    ...
</xzufi:spezialisierung>
```

Validator-Id	VAL0013
Beschreibung	Die Spezialisierung muss ein zu spezialisierendes Leistungsmodul enthalten.
Begründung	**Datenqualitäts- Validator:** Spezialisierungen dienen zur Individualisierung und Anpassung von Leistungen. Wenn sie ohne tatsächliche Änderungen geliefert werden, sind sie faktisch leer und erfüllen den Zweck der Spezialisierung von allgemeinen Leistungen nicht. Dies legt einen Fehler seitens der Redaktion nahe und wird entsprechend vom Validatordienst abgelehnt.

Beispiele (Falsch)

Das Element *spezialisierung* enthält kein Element *modulSpezialisiert*.

```
<xzufi:spezialisierung>
    <xzufi:idLeistung schemeAgencyID="L000000">57</xzufi:idLeistung>
    <xzufi:idGebiet schemeAgencyID="L000000">091610000000</xzufi:idGebiet>
</xzufi:spezialisierung>
```

Das Element *modulSpezialisiert* enthält kein Element *modul*. Es ist keine Information vorhanden, welches Leistungsmodul spezialisiert wird.

```
<xzufi:spezialisierung>
    <xzufi:idLeistung schemeAgencyID="L000000">57</xzufi:idLeistung>
    <xzufi:idGebiet schemeAgencyID="L000000">091610000000</xzufi:idGebiet>
    ...
    <xzufi:modulSpezialisiert>
        <xzufi:ersetzung>false</xzufi:ersetzung>
    </xzufi:modulSpezialisiert>
</xzufi:spezialisierung>
```

Beispiel (Richtig)

Das Element *modulSpezialisiert* enthält ein Element *modul*. Im *modul* wird definiert, dass ein Textmodul mit dem Code *06* spezialisiert wird.

```
<xzufi:spezialisierung>
    <xzufi:idLeistung schemeAgencyID="L000000">57</xzufi:idLeistung>
    <xzufi:idGebiet schemeAgencyID="L000000">091610000000</xzufi:idGebiet>
    ...
    <xzufi:modulSpezialisiert>
        <xzufi:modul xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="xzufi:Textmodul">
            ...
```

```
            <xzufi:leikaTextmodul
listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
                                listVersionID="20201201">
                <code>06</code>
                <name>Volltext</name>
            </xzufi:leikaTextmodul>
        </xzufi:modul>
        <xzufi:ersetzung>false</xzufi:ersetzung>
    </xzufi:modulSpezialisiert>
...
</xzufi:spezialisierung>
```

Validator-Id	VAL0012
Beschreibung	Ein Redaktionssystem darf nur die Daten des Landes übertragen, für das es zuständig ist. Der ARS-Schlüssel der Gebiete muss mit dem Präfix des zuständigen Landes beginnen. ARS-Schlüssel im Zusammenhang mit Leistungen im gemeinsamen Vollzug sind von dieser Prüfung ausgenommen. Solche Leistungen haben eine Kategorie mit der Bezeichnung "Leistung im gemeinsamen Vollzug". Der Validator prüft ARS-Schlüssel in den Elementen *gueltigkeitGebietID* an der Leistung, *gebietID* an den Zuständigkeiten (auch von Kontaktpersonen) der Organisationseinheit und des Formulars sowie *idGebiet* an der Spezialisierung.
Begründung	**Datenqualitäts- Validator:** Diese Validierungsregel stellt sicher, dass Redaktionssysteme nur die Daten innerhalb ihres originären Zuständigkeitsgebietes liefern und ändern dürfen. Damit werden versehentliche Änderungen und Dopplungen im Datenbestand verhindert.

Beispiele (Falsch)

Die Daten werden von dem Redaktionssystem *Bayern* geliefert. Das Redaktionssystem *Bayern* ist nach Sammlerdienst-Konfiguration für die Gebiete, die mit dem Präfix *09* beginnen, zuständig.

In den Daten wird eine Zuständigkeit des Landes Schleswig-Holstein *011630000000* mit dem Präfix *01* übertragen.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">011630000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Das Gültigkeitsgebiet der Leistung *gueltigkeitGebietID* ist *01*. Das bedeutet, dass das Redaktionssystem *Bayern* eine Leistung des Landes Schleswig-Holstein ausliefert.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000">57</xzufi:id>
    ...
    <xzufi:gueltigkeitGebietID
schemeAgencyID="L000000">010000000000</xzufi:gueltigkeitGebietID>
    ...
</xzufi:leistung>
```

Beispiele (Richtig)

Die Daten werden vom Redaktionssystem *Bayern* geliefert. Das Redaktionssystem *Bayern* ist nach Sammlerdienst-Konfiguration für die Gebiete, die mit dem Präfix *09* beginnen, zuständig.

In den Daten wird eine Zuständigkeit *091630000000* mit dem Präfix *09* übertragen.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">091630000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Es wird ein Leistungsmodul für das Gebiet *091630000000* mit dem Präfix *09* spezialisiert.

```
<xzufi:spezialisierung>
    <xzufi:idLeistung schemeAgencyID="L000000">57</xzufi:idLeistung>
    <xzufi:idGebiet schemeAgencyID="L000000">091630000000</xzufi:idGebiet>
    ...
</xzufi:spezialisierung>
```

Das Gültigkeitsgebiet der Leistung *gueltigkeitGebietID* ist *09*.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000">57</xzufi:id>
    ...
    <xzufi:gueltigkeitGebietID
schemeAgencyID="L000000">090000000000</xzufi:gueltigkeitGebietID>
    ...
</xzufi:leistung>
```

Validator-Id	VAL0011
Beschreibung	Das XZuFi-Dokument darf nur maskierte HTML-Tags enthalten, die in der XZuFi-Spezifikation zugelassen sind. Eine Ausnahme stellen die HTML-Tags 'div' mit dem Attribut 'lang' und Attribut 'href' am Tag 'a' dar. Auch darin dürfen nur Protokolle auf der Whitelist verwendet werden, z.B. "mailto:" oder "sms:". Diese Einschränkung ist nötig, um die Verwendung von dort evtl. hinterlegtem Code zu unterbinden.+
Begründung	**Standard- und Datenqualitäts- Validator:** Der XZuFi-Standard schränkt ein, welche HTML-Tags an welchen Objekten und Attributen verwendet werden dürfen. Diese Einschränkungen sind inhaltlich sowohl an technischen als auch Datenqualitätsgründen orientiert. Diese Validierungsregel hat durch die Verbindlichkeit des Standards ebenfalls verbindlichen Charakter.

Beispiel (Falsch)

HTML Tags * * ohne Attribut *lang* und Tabelle *<table>* sind nicht erlaubt.

```
<xzufi:modulText>
    <xzufi:positionDarstellung>3</xzufi:positionDarstellung>
    <xzufi:inhalt languageCode="de">
    &lt;div&gt; Hier kommen nicht erlaubte Tags vor &lt;/div&gt;
    &lt;table&gt;
    &lt;tr&gt;
      &lt;th&gt;Monat&lt;/th&gt;
      &lt;th&gt;Datum&lt;/th&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
      &lt;td&gt;Januar&lt;/td&gt;
      &lt;td&gt;01.01.2022&lt;/td&gt;
    &lt;/tr&gt;
  &lt;/table&gt;
    </xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
                          listVersionID="20150601">
        <code>11</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Beispiel (Richtig)

Ein Absatz *<p>* und eine Liste *
* sind im XZuFi-Standard erlaubt.

```
<xzufi:modulText>
    <xzufi:positionDarstellung>3</xzufi:positionDarstellung>
    <xzufi:inhalt languageCode="de">
        &lt;p&gt; Hier kommen erlaubte Tags vor &lt;/p&gt;
        &lt;ul&gt;
            &lt;li&gt;Ein Absatz und &lt;/li&gt;
            &lt;li&gt; eine Liste sind erlaubt. &lt;/li&gt;
        &lt;/ul&gt;
    </xzufi:inhalt>
    <xzufi:leikaTextmodul listURI="urn:de:xzufi:codeliste:leistungstextmodulleika"
                          listVersionID="20150601">
        <code>11</code>
    </xzufi:leikaTextmodul>
</xzufi:modulText>
```

Validator-Id	VAL0010
Beschreibung	Der ARS-Schlüssel bei den Gebieten muss 12-stellig sein. Der Validator prüft ARS-Schlüsseln im Element *gueltigkeitGebietID* an der Leistung, im Element *gebietID* an den Zuständigkeiten der Organisationseinheit, des Formulars und des Onlinedienstes sowie *idGebiet* an der Spezialisierung.
Begründung	**Technischer, Standard- und Datenqualitäts-Validator:** Wohlgeformte ARS-Schlüssel sind 12-stellig. Damit Zuständigkeiten funktionieren können, ist die standardkonforme Verwendung der ARS-Schlüssel notwendig. Sollten ARS-Schlüssel nicht korrekt sein, z.B. und u.a. wenn sie nicht 12 Stellen aufweisen, wird die Zuständigkeitsfindung in Mitleidenschaft gezogen. In der Folge werden Daten nicht verknüpft und stehen für die Verwendung effektiv nicht zur Verfügung, sie gehen verloren.

Beispiel (Falsch)

Der ARS-Schlüssel *01* besteht aus zwei Ziffern.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000">57</xzufi:id>
    ...
    <xzufi:gueltigkeitGebietID schemeAgencyID="L000000">01</xzufi:gueltigkeitGebietID>
    ...
</xzufi:leistung>
```

Der ARS-Schlüssel *01004* besteht aus fünf Ziffern.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">01004</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Beispiel (Richtig)

Der ARS-Schlüssel *010000000000* ist zwölfstellig.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000">57</xzufi:id>
    ...
    <xzufi:gueltigkeitGebietID
schemeAgencyID="L000000">010000000000</xzufi:gueltigkeitGebietID>
    ...
</xzufi:leistung>
```

Der ARS-Schlüssel *010040000000* ist zwölfstellig.

```
<xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:type="xzufi:OrganisationseinheitErweitert">
    ...
    <xzufi:zustaendigkeit>
        ...
        <xzufi:gebietID schemeAgencyID="DESTATIS"
schemeDataURI="urn:de:bund:destatis:bevoelkerungsstatistik:schluessel:rs"
schemeVersionID="2022-09-30">010040000000</xzufi:gebietID>
    </xzufi:zustaendigkeit>
</xzufi:organisationseinheit>
```

Validator-Id	VAL0009
Beschreibung	An der Leistung oder, wenn nicht an der Leistung, an allen dazugehörigen Spezialisierungen muss mindestens ein UrsprungsportalModul existieren. **1)** Existiert ein XZuFi-Element *modulUrsprungsportal* an der Leistung, so gilt diese für alle Gebiete und die Leistung ist valide. **2)** Existiert kein XZuFi-Element *modulUrsprungsportal* an der Leistung, so muss sichergestellt werden, dass für jedes Gebiet, das spezialisiert wird, ein UrsprungsportalModul in der Spezialisierung existiert.
Begründung	**Datenqualitäts- Validator:** Der Verweis (ggf. inklusive URI/Link) zum Ursprungsportal bzw. -system dient der Nachvollziehbarkeit von Daten innerhalb des Portalverbundes. Zielgruppe sind nicht Bürgerinnen und Bürger, sondern die Teilnehmer des PV. Dies ist eine fachliche Konvention zwischen den Redaktionssystemen und ohne technische Implikationen.

Beispiel (Falsch)

Das Element *modulUrsprungsportal* ist an der Leistung nicht vorhanden.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000" schemeID="OID">8965820</xzufi:id>
    ...
</xzufi:leistung>
```

Beispiele (Richtig)

Das Element *modulUrsprungsportal* ist an der Leistung vorhanden.

```
<xzufi:leistung>
    <xzufi:id schemeAgencyID="L000000" schemeID="OID">8965820</xzufi:id>
    ...
    <xzufi:modulUrsprungsportal
        languageCode="de">
        <xzufi:positionDarstellung>1</xzufi:positionDarstellung>

<xzufi:uri>https://www.muster.de/portal/?tsa_leistung_id=8965820&amp;tsa_sprache=de_DE
</xzufi:uri>
        <xzufi:titel>Abfallrechtliches Nachweisverfahren</xzufi:titel>
    </xzufi:modulUrsprungsportal>
    ...
```

```
</xzufi:leistung>
```

Das Element *modulUrsprungsportal* ist an der Leistung nicht vorhanden aber es existiert eine Spezialisierung mit dem *UrsprungsportalModul*.

```
<xzufi:schreibe>
    <xzufi:leistung>
        <xzufi:id schemeAgencyID="L000000" schemeID="OID">8965820</xzufi:id>
        ...
    </xzufi:leistung>
</xzufi:schreibe>

...

<xzufi:schreibe>
    <xzufi:spezialisierung>
        <xzufi:idLeistung schemeAgencyID="L000000"
schemeID="OID">8965820</xzufi:idLeistung>
        ...
        <xzufi:modulSpezialisiert>
            <xzufi:modul
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:type="xzufi:UrsprungsportalModul" languageCode="de">
                <xzufi:positionDarstellung>0</xzufi:positionDarstellung>

<xzufi:uri>https://www.muster.de/portal/?tsa_leistung_id=8965820&amp;tsa_gebiet_id=895
5323&amptsa_sprache=de_DE</xzufi:uri>
                <xzufi:titel>Abfallrechtliches Nachweisverfahren</xzufi:titel>
            </xzufi:modul>
            <xzufi:ersetzung>false</xzufi:ersetzung>
        </xzufi:modulSpezialisiert>
        <xzufi:id schemeAgencyID="L000000" schemeID="OID">358628793_8955323</xzufi:id>
    </xzufi:spezialisierung>
</xzufi:schreibe>
```

Validator-Id	VAL0008
Beschreibung	Das Übertragen von Gebieten in einer Transfernachricht ist verboten.
Begründung	**Technischer Validator:** Das Gebiet ist "geschachtelt" in einer Organisationseinheit zu übertragen. Dies ist eine grundlegende Architekturentscheidung aus den Jahren 2019/20.

Beispiel (Falsch)

Es wird ein Geschäftsobjekt *gebiet* übertragen. Das ist nicht erlaubt.

```
<xzufi:transfer.operation.040502>
 <xzufi:schreibe>
    <xzufi:gebiet>
      <xzufi:id schemeAgencyID="8cde6870-fe84-4341-bc9f-e65232481186"
schemeAgencyName="SchemeAgencyName" schemeDataURI="https://example.com"
schemeID="8cde6870-fe84-4341-bc9f-e65232481186" schemeName="SchemeName"
schemeVersionID="1.0.0">8cde6870-fe84-4341-bc9f-e65232481186</xzufi:id>
      <xzufi:typ>
        <xzufi:id schemeAgencyID="c1fed905-c30d-47bb-ac1b-ff2a3db172c7"
schemeAgencyName="SchemeAgencyName" schemeDataURI="https://example.com"
schemeID="c1fed905-c30d-47bb-ac1b-ff2a3db172c7" schemeName="SchemeName"
schemeVersionID="1.0.0">c1fed905-c30d-47bb-ac1b-ff2a3db172c7</xzufi:id>
        <xzufi:klasse>
          <xzufi:id schemeAgencyID="d5d650d4-09f4-4e56-9b09-fef69a974651"
schemeAgencyName="SchemeAgencyName" schemeDataURI="https://example.com"
schemeID="d5d650d4-09f4-4e56-9b09-fef69a974651" schemeName="SchemeName"
schemeVersionID="1.0.0">d5d650d4-09f4-4e56-9b09-fef69a974651</xzufi:id>
          <xzufi:bezeichnung languageCode="de">MeineGebietsklasse</xzufi:bezeichnung>
        </xzufi:klasse>
        <xzufi:bezeichnung languageCode="de">MeineGebietstyp</xzufi:bezeichnung>
      </xzufi:typ>
      <xzufi:bezeichnung languageCode="de">MeineGebiet</xzufi:bezeichnung>
    </xzufi:gebiet>
  </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Es wird eine Leistung mit einer Organisationseinheit übertragen. Das Gebiet wird nicht übertragen.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">LeistungsId_01</xzufi:id>
            ...
        </xzufi:leistung>
```

```
    </xzufi:schreibe>

    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">OEID_TC_010_OE01</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0007
Beschreibung	Das Übertragen von Leistungskategorien in einer Transfernachricht ist verboten.
Begründung	**Technischer Validator:** Die Leistungskategorie ist "geschachtelt" in einer Leistung zu übertragen. Dies ist eine grundlegende Architekturentscheidung aus den Jahren 2019/20.

Beispiel (Falsch)

Es wird ein Geschäftsobjekt *leistungskategorie* übertragen. Das ist nicht erlaubt.

```
<xzufi:transfer.operation.040502>
 <xzufi:schreibe>
    <xzufi:leistungskategorie>
      <xzufi:klasse>
        <xzufi:id schemeAgencyID="L000000">45000</xzufi:id>
        <xzufi:bezeichnung languageCode="de">Klasse</xzufi:bezeichnung>
      </xzufi:klasse>
      <xzufi:id schemeAgencyID="L000000">12300</xzufi:id>
      <xzufi:bezeichnung languageCode="de">Bezeichnung</xzufi:bezeichnung>
    </xzufi:leistungskategorie>
  </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Es wird eine Leistung mit einer Organisationseinheit übertragen. Die Leistungskategorie wird nicht übertragen.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">012345</xzufi:id>
        ...
        </xzufi:leistung>
    </xzufi:schreibe>

    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">9867</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0006
Beschreibung	Eine Löschoperation in einer Transfernachricht ist verboten. Eine Ausnahme stellt die PUSH-Schnittstelle des Sammlerdienstes dar. Über die PUSH-Schnittstelle übertragene Löschoperationen werden akzeptiert.
Begründung	**Technischer Validator:** Eine Löschoperation ist "geschachtelt" in dem Objekt zu übertragen, welches aktualisiert/gelöscht werden soll. Dies ist eine grundlegende Architekturentscheidung aus den Jahren 2019/20.

Beispiel (Falsch)

Die Transfernachricht enthält eine Löschoperation *loesche*.

```
<xzufi:transfer.operation.040502>
  <xzufi:loesche klasse="xzufi:Leistung">
    <xzufi:id schemeAgencyID="L000000">01232</xzufi:id>
  </xzufi:loesche>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

In der Transfernachricht ist keine Löschoperation enthalten.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">012345</xzufi:id>
        ...
        </xzufi:leistung>
    </xzufi:schreibe>

    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">9867</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0005
Beschreibung	Das Übertragen von Geschäftsobjekt 'Organisationseinheitskategorie' in einer Transfernachricht ist verboten.
Begründung	**Technischer Validator:** Die Organisationseinheitskategorie ist "geschachtelt" in einer Organisationseinheit zu übertragen. Dies ist eine grundlegende Architekturentscheidung aus den Jahren 2019/20.

Beispiel (Falsch)

Es wird ein Geschäftsobjekt *organisationseinheitskategorie* übertragen.

```
<xzufi:transfer.operation.040502>
 <xzufi:schreibe>
    <xzufi:organisationseinheitskategorie>
      <xzufi:klasse>
        <xzufi:id schemeAgencyID="L000000">12329</xzufi:id>
        <xzufi:bezeichnung languageCode="de">Klasse</xzufi:bezeichnung>
      </xzufi:klasse>
      <xzufi:id schemeAgencyID="L000000">023451</xzufi:id>
      <xzufi:bezeichnung languageCode="de">Bezeichnung</xzufi:bezeichnung>
    </xzufi:organisationseinheitskategorie>
  </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Es wird eine Leistung mit einer Organisationseinheit übertragen. Die Organisationseinheitskategorie wird nicht übertragen.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">012345</xzufi:id>
        ...
        </xzufi:leistung>
    </xzufi:schreibe>

    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">9867</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
```

```
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0004
Beschreibung	Das Übertragen von dem Geschäftsobjekt OZG-Leistungen in einer Transfernachricht ist verboten.
Begründung	**Technischer Validator:** Die ozgLeistung ist "geschachtelt" in einer Organisationseinheit zu übertragen. Dies ist eine grundlegende Architekturentscheidung aus den Jahren 2019/20.

Beispiel (Falsch)

Es wird ein Geschäftsobjekt *ozgLeistung* übertragen.

```
<xzufi:transfer.operation.040502>
 <xzufi:schreibe>
    <xzufi:ozgLeistung>
      <xzufi:id listURI="urn:xoev-de:fim:codeliste:xzufi.ozgleistung"
listVersionID="1.0.0">
        <code>001</code>
      </xzufi:id>
      <xzufi:themenfeld listURI="urn:xoev-de:fim:codeliste:xzufi.ozgthemenfeld"
listVersionID="1.0.0">
        <code>002</code>
      </xzufi:themenfeld>
      <xzufi:lage listURI="urn:xoev-de:fim:codeliste:xzufi.ozglage"
listVersionID="1.0.0">
        <code>002</code>
      </xzufi:lage>
      <xzufi:zielgruppe listURI="urn:xoev-de:fim:codeliste:xzufi.leistungsadressat"
listVersionID="1.0">
        <code>001</code>
      </xzufi:zielgruppe>
    </xzufi:ozgLeistung>
  </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Es wird eine Leistung mit einer Organisationseinheit übertragen. Die OZG-Leistung wird nicht übertragen.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">012345</xzufi:id>
        ...
        </xzufi:leistung>
    </xzufi:schreibe>
```

```
    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">9867</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0003
Beschreibung	Das Übertragen des Geschäftsobjektes 'person' in einer Transfernachricht ist nicht erlaubt.
Begründung	**Technischer Validator:** Die Kontaktperson ist "geschachtelt" in einer Organisationseinheit zu übertragen. Dies ist eine grundlegende Architekturentscheidung aus den Jahren 2019/20.

Beispiel (Falsch)

Es wird ein Geschäftsobjekt *person* übertragen. Das ist nicht erlaubt.

```
<xzufi:transfer.operation.040502>
 <xzufi:schreibe>
    <xzufi:person>
      <xzufi:organisationseinheitID
schemeAgencyID="L000000">12303</xzufi:organisationseinheitID>
      <xzufi:kontaktperson>
        <xzufi:id schemeAgencyID="L000000">1292</xzufi:id>
        <xzufi:familienname>Muster-mann/frau/diverse</xzufi:familienname>
      </xzufi:kontaktperson>
    </xzufi:person>
  </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Es wird eine Leistung mit einer Organisationseinheit übertragen. Das Geschäftsobjekt 'person' wird nicht übertragen.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">012345</xzufi:id>
        ...
        </xzufi:leistung>
    </xzufi:schreibe>

    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">9867</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0002
Beschreibung	Das Übertragen des Geschäftsobjekts 'zuständigkeit' in einer Transfernachricht ist nicht erlaubt.
Begründung	**Technischer Validator:** Zuständigkeitsobjekte sind "geschachtelt" zu übermitteln, jeweils in den Organisationseinheit-, Online-Dienst- oder Formular-Objekten. Dies sind grundlegende Architekturentscheidungen aus dem Jahr 2019/20.

Beispiel (Falsch)

Es wird ein Geschäftsobjekt *zustaendigkeitTransferObjekt* übertragen.

```
<xzufi:transfer.operation.040502>
 <xzufi:schreibe>
    <xzufi:zustaendigkeitTransferObjekt>
      <xzufi:zustaendigkeit>
        <xzufi:leistungID schemeAgencyID="L000000">120200</xzufi:leistungID>
      </xzufi:zustaendigkeit>
    </xzufi:zustaendigkeitTransferObjekt>
  </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Beispiel (Richtig)

Es wird eine Leistung mit einer Organisationseinheit übertragen. Das Geschäftsobjekt 'zuständigkeit' wird nicht übertragen.

```
<xzufi:transfer.operation.040502>
    <xzufi:schreibe>
        <xzufi:leistung>
            <xzufi:id schemeAgencyID="L000000">012345</xzufi:id>
        ...
        </xzufi:leistung>
    </xzufi:schreibe>

    <xzufi:schreibe>
        <xzufi:organisationseinheit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="xzufi:OrganisationseinheitErweitert">
            <xzufi:id schemeAgencyID="L000000">9867</xzufi:id>
            ...
        </xzufi:organisationseinheit>
    </xzufi:schreibe>
</xzufi:transfer.operation.040502>
```

Validator-Id	VAL0001
Beschreibung	An der Zuständigkeit muss mindestens eine GebietID existieren.
Begründung	**Technischer und Standard-Validator:** ohne die Verknüpfung mit einer GebietID kann die Zuständigkeit ihren Zweck lt. XZuFi-Standard nicht erfüllen. Zitat S. 113 XZuFi 2.2-Standard: "Zuständigkeit definiert die fachliche und räumliche Zuständigkeit unter optionaler Einbeziehung weiterer Kriterien. Es sollen Leistungen und Gebiete mit gleichen Zuständigkeitskriterien in einem Zustaendigkeitsobjekt zusammengefasst werden." Die GebietID ist damit zwingende Voraussetzung für jegliche Zuständigkeitsfindung innerhalb des PVOG oder bei nachnutzenden Systemen.

Beispiel (Falsch)

Die Zuständigkeit besitzt kein Element *gebietID*.

```
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
</xzufi:zustaendigkeit>
```

Beispiel (Richtig)

Die Zuständigkeit besitzt zwei Elemente *gebietID*.

```
<xzufi:zustaendigkeit>
    <xzufi:leistungID schemeAgencyID="L000000">3815</xzufi:leistungID>
    <xzufi:gebietID schemeAgencyID="L000000">091630000000</xzufi:gebietID>
    <xzufi:gebietID schemeAgencyID="L000000">090000000000</xzufi:gebietID>
</xzufi:zustaendigkeit>
```