



**FIM**

Föderales  
Informationsmanagement

## Produkthandbuch Kapitel 3.3

Technische Rahmenbedingungen für den Aufbau  
einer FIM-Redaktion (Konformität)

*Datum: 24.01.2022*

*Version: 1.0 Release Candidate*

## Änderungsverzeichnis

Lfd. Nr.	Datum	Version/ Status	Änderungen
1	14.09.2021	Entwurf	Erstellung und Abstimmung in der GK FIM.
2	15.10.2021	0.9	Zur Abstimmung an die FIM Fachgruppe verteilt.
3	24.01.2022	1.0 Release Candidate	<p>Kleinere sprachliche Anpassungen nach Rückmeldungen durch die FIM FG-Mitglieder sowie:</p> <ul style="list-style-type: none"> <li>• Kapitel 3.3.1.1 „Editor“ zu dem Redaktionssystem im Bausteins Leistungen: Der Punkt „Import der Dokumentsteckbriefe des FIM-Bausteins Datenfelder, die als Formular referenziert werden“ wird von einer Muss- zu einer Soll-Anforderung.</li> <li>• Kapitel 3.3.2.2 „Editor“ zu dem Redaktionssystem im Baustein Prozesse: Das Anlegen von Prozessbibliotheken im Redaktionssystem durch die Modellierenden wird von einer Muss- zu einer Kann-Anforderung.</li> <li>• Kapitel 3.3.5.1 „FIM-Portal“: Streichen von „zentralen“ in dem Satz „Dazu ist im FIM-Portal eine Suche integriert, die auf die Inhalte der <del>zentralen</del> Repositorys der drei Bausteine (Leistungen, Prozesse und Datenfelder) über die JSON-Schnittelle zugreift.“</li> <li>• Kapitel 3.3.5.3 „Zentrale Repositorys der Bausteine“: Entfernen „Zentrale“ aus der Überschrift und in dem Text des Kapitels im Zusammenhang mit den Repositorys. Hintergrund: Der Baustein Datenfelder verfügt aktuell über kein zentrales sondern verteilte Repositorys in den Redaktionen.</li> </ul>

## Inhaltsverzeichnis

3.3	Aufbau einer FIM-Redaktion.....	4
3.3.1	Redaktionssystem im Baustein Leistungen.....	4
3.3.1.1	<i>Editor</i> .....	4
3.3.1.2	<i>Repository</i> .....	5
3.3.2	Redaktionssystem im Baustein Prozesse .....	6
3.3.2.1	<i>Prozessbaukasten</i> .....	6
3.3.2.2	<i>Editor</i> .....	6
3.3.2.3	<i>Repository</i> .....	7
3.3.3	Redaktionssystem im Baustein Datenfelder .....	8
3.3.3.1	<i>Editor</i> .....	8
3.3.3.2	<i>Repository</i> .....	9
3.3.4	Datenaustauschformate .....	10
3.3.4.1	<i>XZuFi</i> .....	10
3.3.4.2	<i>XProzess</i> .....	10
3.3.4.3	<i>XDatenfelder</i> .....	10
3.3.5	Anbindung an Querschnittssysteme .....	11
3.3.5.1	<i>FIM-Portal</i> .....	11
3.3.5.2	<i>Ticket-System</i> .....	11
3.3.5.3	<i>Repositorys der Bausteine</i> .....	11

### 3.3 Aufbau einer FIM-Redaktion

Um am Austausch von FIM-Ergebnissen partizipieren zu können, müssen diese mit entsprechend ausgerichteten Werkzeugen erstellt oder übernommen werden können. Dabei soll ein redaktioneller Nachbearbeitungsaufwand möglichst vermieden und der automatisierte elektronische Austausch unterstützt werden.

Diese Werkzeuge benötigen ein FIM-konformes Funktionsspektrum, um FIM-Ergebnisse so zu erstellen, dass sie vom nachnutzungswilligen Produkt unabhängig fehlerfrei importiert oder von Redaktionen z. B. im FIM-Portal publiziert werden können. FIM-Ergebnisse sollen dabei so interpretiert werden können, dass die erforderlichen Informationen zum Lebenszyklus und der eigentlichen Ausprägung, Struktur und Festlegung zu den entsprechenden Elementen ausgewertet und für die eigene Systemumgebung gemäß XÖV-Standard übernommen werden können. Werkzeuge, die dies im Rahmen der Erstellung oder Übernahme von FIM-Ergebnissen ermöglichen, gelten als FIM-konform.

Für die jeweiligen Werkzeuge, z. B. Editoren/Redaktionsumgebungen, gelten die im Folgenden beschriebenen bausteinspezifischen Konformitätsanforderungen.

#### 3.3.1 Redaktionssystem im Baustein Leistungen

Ein Redaktionssystem **muss** nachfolgend genannte Funktionalitäten aufweisen. Diese Funktionalitäten gewährleisten das Erstellen und den Austausch von FIM-Kataloginformationen (Leistungssteckbriefe) und Bibliotheksinformationen (Stammtexte).

Darüber hinaus kann ein Redaktionssystem weitere Funktionalitäten zur Verfügung stellen, die z. B. das redaktionelle Arbeiten erleichtern oder die Qualitätssicherung ermöglichen, jedoch nicht zwingend für die Erfassung und den Austausch (Im-/Export) sind.

Das Redaktionssystem muss

- die jeweils geltende Version des XZuFi-Standards und
- die im Folgenden geschilderten Anforderungen des Fach- und des QS-Konzepts FIM-Baustein Leistungen unterstützen.

##### 3.3.1.1 Editor

Der Editor **muss** folgende Funktionen unterstützen:

- Im-/Export
  - einzelner Katalogelemente (Leistungssteckbriefe) im XZuFi-Format,
  - einzelner Leistungen (Stammtexte und Leistungsbeschreibungen) im XZuFi-Format und
  - über die XZuFi-Schnittstelle zum Austausch von Katalog- und Bibliothekseinträge mit anderen Redaktionssystemen (Redaktionssystem des Bausteins Leistungen und dezentrale Redaktionssysteme).
- Bearbeiten (Erfassung)

- Anlegen neuer sowie Bearbeiten, offline-/online-Setzen, Versionieren und Löschen vorhandener Katalog- und Bibliothekseinträge (Leistung, Organisationseinheit, Online-Dienst, Formular).
- Anzeige und Bearbeitung
  - der Metadaten des Leistungssteckbriefs,
  - der Module des Leistungssteckbriefs,
  - der Module des Bibliothekseintrags und
  - Validierung der Daten bei Bearbeitung gemäß QS-Handbuch des FIM Bausteins Leistungen.
- Mehrsprachigkeit  
 Bearbeitung deutsch- und englischsprachiger Katalog- und Bibliothekseinträge sowie die Möglichkeit zur Erweiterung auf weitere Sprachen inklusive weitere Zeichensätze der europäischen Sonderzeichen.

Der Editor **sollte** den Import der Dokumentsteckbriefe des FIM-Bausteins Datenfelder, die als Formular referenziert werden, unterstützen können.

### 3.3.1.2 Repository

Die Minimalanforderungen an das Repository sind:

- Parallele Unterstützung mehrerer Versionen des XZuFi-Austauschformates (aktuelle Version und deren Vorläufer).
- Im-/Export
  - Unterstützung der aktuellen Version des XZuFi Datenaustauschstandards,
  - Kopplung zum Datenaustausch mit dem übergeordneten zentralen Repository des FIM-Bausteins Leistungen gemäß XZuFi-Standard
    - Leistungsbibliothek,
    - Leistungskatalog,
    - Filterung des Datenaustauschs.
  - Import-Schnittstelle zum XRepository zur Einbindung der Codelisten des Leistungsbaukastens (z. B. für Leistungsgruppierungen/Verrichtungskennungen/etc.), die direkt aus dem XRepository bezogen werden.

### 3.3.2 Redaktionssystem im Baustein Prozesse

Die in den nachfolgenden Kapiteln beschriebenen funktionalen Anforderungen sind für das Redaktionssystem des FIM-Bausteins Prozesse (Editor und Repository) relevant.

#### 3.3.2.1 Prozessbaukasten

- Die Modellierenden **müssen** im Redaktionssystem auf einen **Prozessbaukasten** zurückgreifen können. In diesem Prozessbaukasten befinden sich zumindest die in FIM erlaubten BPMN-Elemente und die Referenzaktivitätengruppen, perspektivisch auch die Musterprozesse und Standardaktivitätengruppen.
- Die Modellierenden **müssen** im Redaktionssystem zumindest die **BPMN Elemente**, die in FIM erlaubt sind, für die Modellierung verwenden können.

#### 3.3.2.2 Editor

An den einzusetzenden Editor werden folgende Anforderungen gestellt:

- Das Redaktionssystem kann die BPMN-Elemente über eine Art FIM-Filter einschränken, sodass den Modellierenden nur die FIM-BPMN-Elemente angezeigt werden.
- Die Modellierenden sollen im Redaktionssystem die BPMN-Prozessmodelle in der senkrechten Modellierungsrichtung darstellen können.
- Die Modellierenden **müssen** im Redaktionssystem im BPMN-Element „Teilprozess“ alle Metadaten gemäß den QS-Kriterien erfassen können. Hierbei handelt es sich um die Metadaten der Referenzaktivitätengruppe (RAG).
- Das Redaktionssystem **muss** die Piktogramme der jeweiligen RAG-Typen oben rechts im Teilprozess-Element visualisieren können.
- Das Redaktionssystem **kann** die Aktivitätengruppen-ID im Teilprozess-Element visualisieren.
- Die Modellierenden **müssen** im Redaktionssystem zum BPMN-Prozessmodell alle Metadaten gemäß den QS-Kriterien erfassen können. Hierbei handelt es sich um die Metadaten des Prozesssteckbriefs.
- Die Modellierenden **müssen** im Redaktionssystem den Prozess (=BPMN-Prozessmodell und Prozesssteckbrief) einer Prozessbibliothek zuordnen können. Die Modellierenden **können** im Redaktionssystem mehrere Prozessbibliotheken anlegen. Die Metadaten der Prozessbibliothek werden direkt im XProzess-2.0 Datenaustauschformat beschrieben.
- Die Modellierenden **müssen** im Redaktionssystem den Prozess genau einer Prozessklasse zuordnen können. Dafür **müssen** die Modellierenden im Redaktionssystem einen Prozesskatalog anlegen und mit Prozessklassen füllen können.
- Die Modellierenden **müssen** im Redaktionssystem die Metadaten des Prozesskatalogs und der Prozessklasse erfassen können. Die Metadaten des Prozesskatalogs werden direkt im XProzess-2.0 Datenaustauschformat beschrieben. Die Metadaten der Prozessklasse werden in den QS-Kriterien beschrieben.

- In den Metadaten der Prozessklasse, des Prozesssteckbriefs und der Referenzaktivitätengruppen werden zum Teil Codelisten verwendet. Das Redaktionssystem soll die Codelisten direkt aus dem XRepository beziehen können (siehe Codelisten zum XProzess-Standard: <https://www.xrepository.de/details/urn:xoev-de:mv:em:standard:xprozess>)
- Das Redaktionssystem kann auf Pflicht-Metadaten im Rahmen einer Validierung hinweisen.
- Das Redaktionssystem kann die Beschreibung zu den Metadaten, z. B. per Tooltip/Mouseover anzeigen.

### 3.3.2.3 Repository

An ein Repository im Baustein Prozesse werden folgende Anforderungen gestellt:

- Das Redaktionssystem soll das XProzess 2.0 Datenaustauschformat unterstützen. Dazu gehören die XProzess 2.0 Importschnittstelle und die XProzess 2.0 Exportschnittstelle. Die Import- und Exportschnittstelle **müssen** zumindest die Nachrichtentypen „alleInhalte.anfrage 0301“ und „alleInhalte.antwort 0302“ unterstützen. Auf diese Weise können der **Prozesskatalog** inklusive Prozessklassen sowie die **Prozessbibliothek** inklusive Prozesse importiert und exportiert werden. Das Redaktionssystem soll an die FIM-Backend-Infrastruktur des Bausteins Prozesse per Maschine-zu-Maschine-Kommunikation angebunden werden können.
- Das Redaktionssystem soll auch zukünftige Versionen des XProzess-Austauschformats unterstützen können. Die Version 2.01. ist bereits veröffentlicht, sodass das Redaktionssystem zusätzlich zu den XProzess 2.0 Schnittstellen auch die XProzess 2.01 Schnittstellen umsetzen kann.
- Das Redaktionssystem kann die Dokumentsteckbriefe des FIM-Bausteins Datenfelder importieren, sodass die Modellierenden von den Prozessen auf die Dokumentsteckbriefe referenzieren können.
- Das Redaktionssystem kann die Leistungssteckbriefe des FIM-Bausteins Leistungen importieren, sodass die Modellierenden von den Prozessen und Prozessklassen auf die Leistungssteckbriefe referenzieren können.

### 3.3.3 Redaktionssystem im Baustein Datenfelder

Ein Redaktionssystem muss die in den nachfolgenden Kapiteln beschriebenen Funktionalitäten aufweisen.

#### 3.3.3.1 Editor

Ein entsprechendes Werkzeug ermöglicht den FIM-Datenfeldredaktionen FIM-Ergebnisse zu erstellen, zu ändern, zu ergänzen und in einer jeweils geltenden respektive noch unterstützten Version des Standards XDatenfelder zu im-/exportieren. Idealerweise werden die Ergebnisse in ein Repository geschrieben, welches die strukturierte Ablage innerhalb einer Bibliothek anhand eines Katalogs ermöglicht und über Schnittstellen zum Datenaustausch mit dem FIM-Portal und zu anderen Repositories wie BOB und Sammelrepository verfügt.

Das Werkzeug räumt die Möglichkeiten ein, die zwingend gemäß Fachkonzept FIM-Baustein Datenfelder adäquat vorhanden sind.

Zusammenfassung der Anforderungen zur Erzeugung von FIM Ergebnissen:

- **Dokumentsteckbriefe**  
Der Editor verfügt über entsprechende Funktionen, um Dokumentsteckbriefe mit Identifier und allen erforderlichen Meta-Angaben (Pflichtfelder) zur Beschreibung des Sinns und Zwecks eines Dokuments XDatenfelder und konform den Qualitätskriterien zu erzeugen.
- **Datenschemata**  
Der Editor dient der strukturierten Erstellung von Datenschemata (Stamm- oder Referenzdatenschemata) konform den Qualitätskriterien basierend auf Feldgruppen, Feldern mit Bezug auf intern erzeugte oder extern referenzierte Codelisten und mit der Möglichkeit, Regeln gemäß Regelwerk FIM-Datenfelder zu hinterlegen. Datenschemata enthalten Metadaten, die den Einsatzzweck bestimmen, Auskunft über die Handlungsgrundlage, Ersteller sowie Versionierung und Status geben.
  - Elemente müssen dabei nach einem Baukastenprinzip wiederverwertbar sein.
  - Elemente müssen mit Metadaten beschrieben und mit einem Identifier versehen sein, die Auskunft über die Version, die Redaktion, den Status und die Ausprägung sowie Handlungsgrundlage geben und XDatenfelder konform sein.
  - Elemente können aus den zentralen Repositories, z. B. BOB, übernommen werden.
- **Codelisten**  
Der Editor ermöglicht die Erstellung von XDatenfelder konformen Codelisten. Diese verfügen über Meta-Angaben, die der Identifikation und Versionierung dienen.
- **Regeln**  
Felder und Gruppen können Regeln zugewiesen werden, die der Steuerung und Auswertung dienen, z. B. als Prüfroutine. Diese Regeln müssen zumindest verwortet erfasst werden können. In Erweiterung ist die Verwendung einer API Regelwerk FIM empfohlen, die die Script-Hinterlegung einräumt. Entsprechende XDatenfelder konforme Meta-Angaben ermöglichen die Referenzierung und Identifikation.

- **Mehrsprachigkeit**  
Der Editor unterstützt die Mehrsprachigkeit der Elemente. Dies betrifft alle textuellen Inhalte, die ein Schema betreffen, z. B. Labels, Ausgaben statischer Daten, Hilfetexte/Ausfüllhinweise, Codelisteneinträge und deren Label.
- **Im- und Export aus dem Repository über den Editor**  
Eine Schnittstelle ermöglicht die Im- und Exportfunktion
  - einzelner Stammdatenschemata (SDS) über XDatenfelder-Dateien,
  - aller Metadaten,
  - des gesamten Datenfeldbaukastens im Format XDatenfelder,
  - der Datenfeldbibliothek (mehrere Stammdatenschemata) im Format XDatenfelder,
  - des Datenfeldkatalogs (mehrere Dokumentensteckbriefe) im Format XDatenfelder,
  - der Import-Schnittstelle vom XRepository zur Implementierung von Codelisten auf Feldebene.
 Dabei wird der direkte Maschine:Maschine-Austausch mit anderen Datenfeld-Repositorys unterstützt. Für FIM-Landesredaktionen ist die Übermittlung der Daten in das FIM-Portal möglich.
- **Unterstützung Redaktionsprozess**
  - Der Editor ermöglicht die Hinterlegung von Statusinformation zum Entwurf, der fachlichen Freigabe, der methodischen Freigabe, der Veröffentlichung. Die Publikation der Inhalte, z. B. in das FIM-Portal oder zur Weitergabe an andere Repositorys, muss redaktionell gesteuert ermöglicht werden können. Dabei sollte eine Automation bei Status „Veröffentlichen nach Freigabe“ vorhanden sein.
  - Der Editor verfügt im Zusammenhang mit einem Repository über ein Rollen- und Rechtemanagement, welches den Redaktionsworkflow unterstützt.

### 3.3.3.2 *Repository*

Inhalte werden in einem strukturierten Repository zur Datenhaltung und zwecks Datenabruf geführt, welches durchsuchbar ist und verwaltet werden kann. Dieses besteht aus einem Katalog, einer Bibliothek und für die Redaktionstätigkeit abrufbaren Elementen (Bausteine), sowie den daraus erzeugten FIM-Informationen (Dokumentsteckbriefe, Stamm- oder Referenzdatenschemata, Codelisten, Regeln).

### **3.3.4 Datenaustauschformate**

Ein wesentlicher Bestandteil von FIM ist der Austausch der FIM-Informationen auf allen föderalen Ebenen. Um dies zu gewährleisten, wurden für jeden Baustein standardisierte Datenaustauschformate entwickelt, die gemäß dem XÖV-Standardisierungsrahmen zertifiziert sind.

#### *3.3.4.1 XZuFi*

Die aktuelle und alle vorherigen Versionen und Codelisten müssen im Editor lesbar sein und bearbeitet werden können.

Alle weiteren Informationen zu dem Standard und den Codelisten finden Sie unter folgendem Link im XRepository: <https://www.xrepository.de/details/urn:xoev-de:fim:standard:xzufi>

#### *3.3.4.2 XProzess*

Die aktuelle und alle vorherigen Versionen und Codelisten müssen im Editor lesbar sein und bearbeitet werden können.

Alle weiteren Informationen zu dem Standard und den Codelisten finden Sie unter folgendem Link im XRepository: <https://www.xrepository.de/details/urn:xoev-de:mv:em:standard:xprozess>

#### *3.3.4.3 XDatenfelder*

Die aktuelle und alle vorherigen Versionen und Codelisten müssen im Editor lesbar sein und bearbeitet werden können.

Alle weiteren Informationen zu dem Standard und den Codelisten finden Sie unter folgendem Link im XRepository: <https://www.xrepository.de/details/urn:xoev-de:fim:standard:xdatenfelder>

### **3.3.5 Anbindung an Querschnittssysteme**

#### *3.3.5.1 FIM-Portal*

Die zentrale Funktion des FIM-Portals (<https://fimportal.de/>) ist die Bereitstellung aller FIM-Stamminformationen. Dazu ist im FIM-Portal eine Suche integriert, die auf die Inhalte der Repositorys der drei Bausteine (Leistungen, Prozesse und Datenfelder) über die JSON-Schnittstelle zugreift. Das FIM-Portal selbst hält keine FIM-Inhalte vor. Über die Suche können sowohl auf die Inhalte der Katalog- wie auch auf die Inhalte der Bibliothekselemente zugegriffen werden. Zusätzlich dient das FIM-Portal als zentrale Informationsquelle für alle Informationen rund um FIM, wie beispielsweise News, Dokumente, Schulungen usw.

#### *3.3.5.2 Ticket-System*

Das FIM Ticketsystem ist über das FIM-Portal (über [ticket@fimportal.de](mailto:ticket@fimportal.de)) erreichbar, deshalb ist eine technische Anbindung nicht notwendig.

#### *3.3.5.3 Repositorys der Bausteine*

Alle drei Baustein verfügen über Repositorys, auf die die Suche des FIM-Portals zugreift. Auf die Katalog- und Bibliothekseinträge in diesen Repositorys kann gemäß den Festlegungen in Kapitel 5.6. „Status von FIM-Informationen“ des Produkthandbuchs und dem „Konzept zur Integration von Landesrecht und kommunalem Satzungsrecht“ über die FIM-Suche zugegriffen werden. Der Datenaustausch mit den Baustein-Repositorys ist in den Baustein-spezifischen Spezifikationen in den jeweiligen XRepositorys beschrieben.