

ELAK-Transaktionen Spezifikation		Konvention
		ELAK-Trans 3.0.1
		Empfehlung
Kurzbeschreibung	<p>Folgende Use-Cases werden abgedeckt:</p> <ul style="list-style-type: none"> • Primär: Bidirektionale Übermittlung von Akten, Geschäftsfällen, Geschäftsstücken (und Dokumenten) zwischen Akten- und Fachinformationssystemen. • Sekundär: Übermittlung von Akten, Geschäftsfällen, Geschäftsstücken (und Dokumenten) zwischen Aktensystemen. <p>Die Anbindung dezentraler Aktensysteme steht immer öfter im Pflichtenheft zentraler Fachinformationssysteme (FIS). Gleichzeitig wird getrieben von behördenspezifischen FIS eine möglichst nahtlose, bidirektionale Integration beider Systeme erwartet, um die effiziente Bearbeitung von Verwaltungsprozessen in Zeiten verknappender Ressourcen zu gewährleisten.</p> <p>ELAK-Trans 3 definiert, aufbauend auf bestehenden Standards wie EDIDOC, einheitliche Schnittstellen und Strukturen zur Umsetzung dieser Anforderungen.</p>	
Autor(en):	<p>Michael Freund, Gerhard Hartmann, Robin Heilig, Gerhard Huber, Christoph Kaiser, Johann Reithmayer, Martin Scheuchenstuhl, Christoph Straßer</p> <p>Gesamtverantwortlich: Christoph Straßer</p>	<p>Projektteam / Arbeitsgruppe:</p> <hr/> <p>AG-II; ELAK-Trans-Projektteam</p>
	<p>Beiträge von: Stefan Arnold, Thomas Kuhta</p>	

Version 3.0.0.: **06.05.2016**

Fristablauf: **05.08.2016**

Abgelehnt von:

(Länderangabe bei ablehnender Stellungnahme)

Unter-Version 3.0.1.: **08.08.2016**

Fristablauf: ---

Abgelehnt von: ---

(Länderangabe bei ablehnender Stellungnahme)

Detail-Version ... : **TT.MM.JJJJ**

Fristablauf: **TT.MM.JJJJ**

Anmerkungen:

(Detailangaben zur Freigabe)

ELAK-Transaktion 3.0.0

Inhaltsverzeichnis

0. Einleitung	6
0.1. Umfang der Umsetzung dieser Schnittstelle	7
0.1.1. Schnittstellenmethoden (bzw. UseCases)	7
0.1.2. EDIDOC-Paket	7
1. Architektur	9
1.1. Portalverbundprotokoll	9
1.2. ELAK-Trans Komponenten	9
1.2.1. ELAK-Trans FIS-2-ELAK	10
1.2.2. ELAK-Trans ELAK-2-FIS	10
1.3. Schema Vereinfachung	10
1.4. Allgemeine Architekturmerkmale	10
1.4.1. Webservice	10
1.4.2. Online Betrieb	10
1.4.3. Batch Betrieb	11
2. Beziehungen zwischen Fachinformationssystem und ELAK	12
2.1. Schlüssel	12
2.2. Welche Daten werden wo abgelegt?	12
2.3. In welchem Umfang steuert das FIS die Aktenstruktur?	12
2.4. Adressierung von FIS bzw. ELAK-Systemen	13
3. Allgemeine technische Festlegungen	14
3.1. EDIDOC	14
3.2. Synchrone Aufrufe	14
3.3. Schlüsselhandling/Schlüsseleindeutigkeit	15
3.3.1. Wie sind die Referenced-Identifizier der Objekte der vier EDIDOC-Ebenen zu interpretieren?	15
3.3.2. Ausnahmen	16
3.3.3. Was muss alles einen Referenced-Identifizier haben?	16
3.3.4. Wo werden die Referenced-Identifizier beim sendData übergeben?	16
3.3.5. Wo werden FIS-ID's von Personen, Adressen und Standorten übergeben?	16
3.4. PVP Application Chaining	17
3.5. Fehlerbehandlung	17
4. Schema-Beschreibung	19
4.1. EDIDOC - reduziertes Schema	19
4.1.1. EdiaktType	19
4.1.2. HeaderType	20
4.1.3. MetaDataType	21
4.1.4. IdentificationType	25
4.1.5. ParticipantsType	26
4.1.6. LocationsType	26
4.1.7. LocationType	27
4.1.8. Layer3Type, Layer2Type	27
4.1.9. Layer1Type, Layer0Type, PayloadType	29
4.1.10. SpecialDataType	31

4.2. PersonData - reduziertes Schema plus Erweiterungen	31
4.2.1. PersonDataType	31
4.2.2. PersonExtensionType	32
4.2.3. PhysicalPersonType, PersonNameType, NationalityType	33
4.2.4. CorporateBodyType	36
4.2.5. BankConnectionType	37
4.2.6. TypedPostalAddressType	37
4.2.7. PostalAddressType	39
4.2.8. InternetAddressType	41
4.2.9. TelephoneAddressType	41
4.2.10. IdentificationType	42
4.2.11. DateOfBirthType	42
4.2.12. DateOfDeathType	42
4.2.13. Wo im PersonData legt das FIS die ID's ab?	42
4.2.14. Welche Adresse der Person wird für die Zustellung verwendet?	44
4.3. ELAK-Trans-spezifische Datenstrukturen	44
4.3.1. CommonInputParameterType	44
4.3.2. CommonOutputParameterType	46
4.3.3. RequestIdentifierType	47
4.3.4. UserMessageType	47
4.3.5. LayerControlType	48
4.3.6. LayerSendControlType	54
4.3.7. LayerResponseControlType	56
5. WS Methoden	60
5.1. Einleitung	60
5.2. FIS -> ELAK Kommunikation (FIS-2-ELAK)	60
5.2.1. SendData	60
5.2.1.1. SendDataInputObject	61
5.2.1.2. SendDataOutputObject	63
5.2.2. ReadData	64
5.2.2.1. ReadDataInputObject	64
5.2.2.2. ReadDataOutputObject	65
5.2.3. Unbind	65
5.2.3.1. UnbindDataInputObject	66
5.2.3.2. UnbindDataOutputObject	66
5.3. ELAK -> FIS Kommunikation (ELAK-2-FIS)	66
5.3.1. SendData	66
5.3.1.1. SendDataInputObject	67
5.3.1.2. SendDataOutputObject	67
5.3.2. ReadData	67
5.3.2.1. ReadDataInputObject	67
5.3.2.2. ReadDataOutputObject	67
6. Anwendungsfälle	68
6.1. ELAK-Trans ELAK-2-FIS Client	68
6.1.1. Daten aktualisieren (L3-L0)	68
6.1.2. Geschäftsstück/ "Aufgabe" übermitteln (L1)	69
6.1.3. Benachrichtigung über Status-Aktualisierung (L3-L0)	70
6.1.4. Benachrichtigung über Zustellstatus-Aktualisierung (L1)	70
6.1.5. URI der FIS-Entität ermitteln (L3-L1)	70
6.1.6. FIS-ID's ermitteln	71
6.2. ELAK-Trans ELAK-2-FIS Server	71

6.2.1. FIS-L3-L0-Entität suchen	72
6.2.2. ReadData verarbeiten	72
6.2.3. SendData verarbeiten	72
6.2.4. Geschäftsstück/"Aufgabe" erzeugen (L1)	73
6.2.5. Status-Aktualisierung verarbeiten (L3-L0)	73
6.2.6. Zustellstatus-Aktualisierung verarbeiten (L1)	74
6.2.7. FIS-L3-L0-Entität aktualisieren	74
6.3. ELAK-Trans FIS-2-ELAK Client	74
6.3.1. Aktenstruktur (Geschäftsobjekte+Dokumente) erzeugen	75
6.3.2. Aktenstruktur (Geschäftsobjekte+Dokumente) aktualisieren	75
6.3.3. Geschäftsstück verschieben	76
6.3.4. Aktenstruktur (Geschäftsobjekte+Dokumente) lesen	76
6.3.5. Aktenstruktur-Verknüpfung lösen	76
6.4. ELAK-Trans FIS-2-ELAK Server	76
6.4.1. Akt suchen	77
6.4.2. Geschäftsfall suchen	78
6.4.3. Geschäftsstück suchen	78
6.4.4. Dokument suchen	78
6.4.5. ELAK-Ziel ermitteln	79
6.4.6. ReadData verarbeiten	79
6.4.7. SendData verarbeiten	81
6.4.8. UnbindData verarbeiten	84
6.4.9. Objekt Mapping aktualisieren	85
6.4.10. Akt erzeugen	89
6.4.11. Akt aktualisieren	89
6.4.12. Geschäftsfall erzeugen	90
6.4.13. Geschäftsfall aktualisieren	91
6.4.14. Geschäftsstück erzeugen	92
6.4.15. Geschäftsstück aktualisieren	93
6.4.16. Dokument erzeugen	94
6.4.17. Dokument aktualisieren	96
7. Abgrenzung	98
7.1. Was ist nicht Scope von ELAK-Trans?	98
7.2. Was ist nicht Scope von ELAK-Trans 3.0 (evtl. spätere Revisionen)	99
7.3. Sonstiges	100
8. Änderungsbeschreibung von ELAKTrans 2.0 zu ELAKTrans 3.0	101
8.1. Methodenzusammenführung	101
8.2. Implizite (Um-)Protokollierung	101
8.3. Objektidentifizierung	101
8.4. Geänderte Elemente	101
8.4.1. Basiselement EDIAKT wird zu EDIDOC	101
8.4.2. Basiselement PersonData	102
8.4.3. CommonParameter	102
8.4.4. LayerUID	102
9. Glossar	103
A. Anhang - Abbildungsverzeichnis	106
B. Anhang - Tabellenverzeichnis	107
C. Anhang - Änderungsprotokoll	108

0. Einleitung

EDIDOC legt als Nachfolger von EDIAKT II die Basis für den Austausch von elektronischen Akteninformationen zwischen Systemen der Verwaltung. EDIDOC konzentriert sich dabei auf die Aspekte der Archivierung und des Offline-Datenaustausches. Die von EDIAKT II / EDIDOC offen gelassene Lücke des Online-Datenaustausches wird seit 2007 durch ELAK-Trans gefüllt.¹

ELAK-Trans regelt den Online-Datenaustausch zwischen Systemen folgender Klassen:

- **ELAK-System:**
Generalisiertes elektronisches System zur Verarbeitung von aktenrelevanten Informationen. Ein wesentliches Merkmal ist die Strukturierung aller Informationen und die Gestaltung der Schlüsselsysteme nach Kriterien der Aktenführung. Die Regeln dafür sind meist verfahrensunabhängig und für eine Organisation oder Organisationseinheit einheitlich gestaltet.
- **Fachinformationssystem (FIS):**
Auf ein bestimmtes Verfahren bzw. eine bestimmte Verfahrensgruppe hin ausgerichtetes System. Die Strukturierung der Informationen und die Gestaltung der Schlüsselsysteme sind abhängig vom jeweiligen Verfahren und kann organisationsunabhängig gestaltet sein (Führerscheinregister, Identitätsregister, GISA, SOTRA, etc.).

Als „Brücke“ zwischen diesen Systemen dient der:

- **ELAK-Trans-Server:**
Im Regelfall vor dem ELAK-System vorgelagerte Implementierung der ELAK-Trans-Spezifikation. Übersetzt standardisierte Aufrufe der ELAK-Trans-Spezifikation in für das jeweilige ELAK-System spezifische Aufrufe. Bei dieser "Übersetzung" kann es sich sowohl um produktspezifische, technische Protokoll-Übersetzungen als auch behörden-spezifische, fachliche Übersetzungen handeln.
Der ELAK-Trans-Server kann auch direkt im ELAK-System integriert sein.
Die Summe aus den in „1 Architektur“ angeführten Komponenten "ELAK-Trans FIS-2-ELAK Server" und "ELAK-Trans ELAK-2-FIS Client".

Der Schwerpunkt der ELAK-Trans-Spezifikation liegt auf der bidirektionalen Kommunikation zwischen ELAK- und Fachinformationssystemen. Organisationsübergreifende (zentrale) Fachinformationssysteme werden dabei über die produktunabhängige ELAK-Trans-Schnittstelle mit dezentralen, organisationsspezifischen ELAK-Systemen gekoppelt. ELAK-Trans erlaubt dabei zu den fachlichen Anforderungen des jeweiligen FIS passende Integrationsgrade bzw. -tiefen.

Darüber hinaus können Aktenstrukturen zwischen ELAK-Systemen automatisiert ausgetauscht werden, was die manuellen Schritte des Exports im Quell-ELAK, die Übermittlung und den Import im Ziel-ELAK erspart. Dazu ist anzumerken, dass die ELAK-zu-ELAK-Kommunikation in ELAK-Trans seit 2007 (damals noch expliziter) Teil des Standards ist, bis 2016 aber aufgrund einer Vielzahl damit verbundener Herausforderungen bzw. negativer Kosten-Nutzen-Überlegungen nie implementiert wurde.

Die in diesem Dokument definierten Methoden des ELAK-Trans-Service werden auf Basis synchroner Web-Services spezifiziert. Anforderungen der Massenverarbeitung wurden ebenso bedacht und sind mit konkreten Lösungsansätzen im Best Practice - Dokument beschrieben.

¹ ELAK-Trans 3 verwendet die EDIDOC-Struktur in der Version 1.0. Diese ELAK-Trans 3 - Spezifikation referenziert, wo sinnvoll und nötig, EDIAKT-Elemente der EDIDOC-Spezifikation.

Die Spezifikation versucht folgende (tlw. einander entgegenstehende) Anforderungen bestmöglich abzudecken:

- Entkoppelung zwischen FIS und ELAK (FIS implementiert die Schnittstelle zu den ELAKs einmal, ohne sich um Spezifika der jeweiligen ELAKs kümmern zu müssen. Versions- oder Produktwechsel der ELAKs dürfen das FIS nicht betreffen. Im FIS können Konfigurationsmöglichkeiten bezüglich der Befüllung des EDIDOC-Pakets vorgesehen sein.²
- Bestmögliche Performance
- Bestmögliches Benutzererlebnis (Sowohl in Szenarien loser Koppelung als auch tiefgehender Integration)

Warum wird die Anbindung dezentraler ELAKs an zentrale FIS angestrebt?

- Zentrale FIS helfen den Verwaltungen, Kosten zu sparen und Prozesse zu vereinheitlichen.
- Die zentralen FIS sollten aber zu keinen "Datensilos" werden, die jeweils auch ihr eigenes Aktensystem (=Fachelak) mitbringen
 - ...da sich die Akten einer Verwaltung dann auf den ELAK und n FIS verteilen.
 - ...da in jedem zentralen FIS die Kosten für die Implementierung der Aktenverwaltungs-Funktionalitäten anfallen.
- Nutzung der Stärken der jeweiligen Systeme.
 - FIS ist auf Fachdaten und Fachprozesse spezialisiert.
 - ELAK auf Aktenverwaltung.

0.1. Umfang der Umsetzung dieser Schnittstelle

0.1.1. Schnittstellenmethoden (bzw. UseCases)

Welche einzelnen Schnittstellenmethoden (bzw. UseCases) aufgegriffen und tatsächlich umgesetzt werden, ist den einzelnen Systemanbietern überlassen. Im Rahmen des Projekt-Kontrakts (siehe Best Practice - Dokument; „Projekt-Kontrakt“) beschreibt das jeweilige FIS den Umsetzungsumfang. Aus diesem leitet sich ab, welche Schnittstellenmethoden (bzw. UseCases) vom ELAK(-Trans-Server) für die Anbindung des jeweiligen FIS implementiert sein müssen und welche implementiert sein können.

0.1.2. EDIDOC-Paket³

Es muss gewährleistet werden, dass alle 4 Ebenen des EDIDOC-Pakets verarbeitet werden. Dies bedeutet, dass – falls auf einer Seite (ELAK-System, Fachinformationssystem) weniger als 4 Ebenen vorhanden sind – vor der Überführung in das EDIDOC-Paket eine Erweiterung stattzufinden hat.

- ELAK-2-FIS-Kommunikation:
Wenn es zB im ELAK-System keinen Geschäftsfall (Layer2) gibt, könnte das Geschäftsstück (Layer1) auf den Layer1 und den Layer2 abgebildet werden.
- FIS-2-ELAK-Kommunikation:
Umgekehrt würde für ein EDIDOC-Paket aus dem Fachinformationssystem für jede

² zB a) wie wird der Betreff gebildet? b) wird die Ebene 2 (Geschäftsfall) genutzt c) welche Teile des EDIDOC-Pakets werden vom Empfänger verarbeitet (kann zB ein zeitintensiver Aufbau gewisser Teile entfallen)

³ EDIDOC-Spezifikation: <http://reference.e-government.gv.at/EP-EA-XML-Schema-zu-EDIDOC-1-0.3107.0.html>

Kombination aus Layer1 und Layer2 ein Geschäftsstück im ELAK-System erzeugt werden.

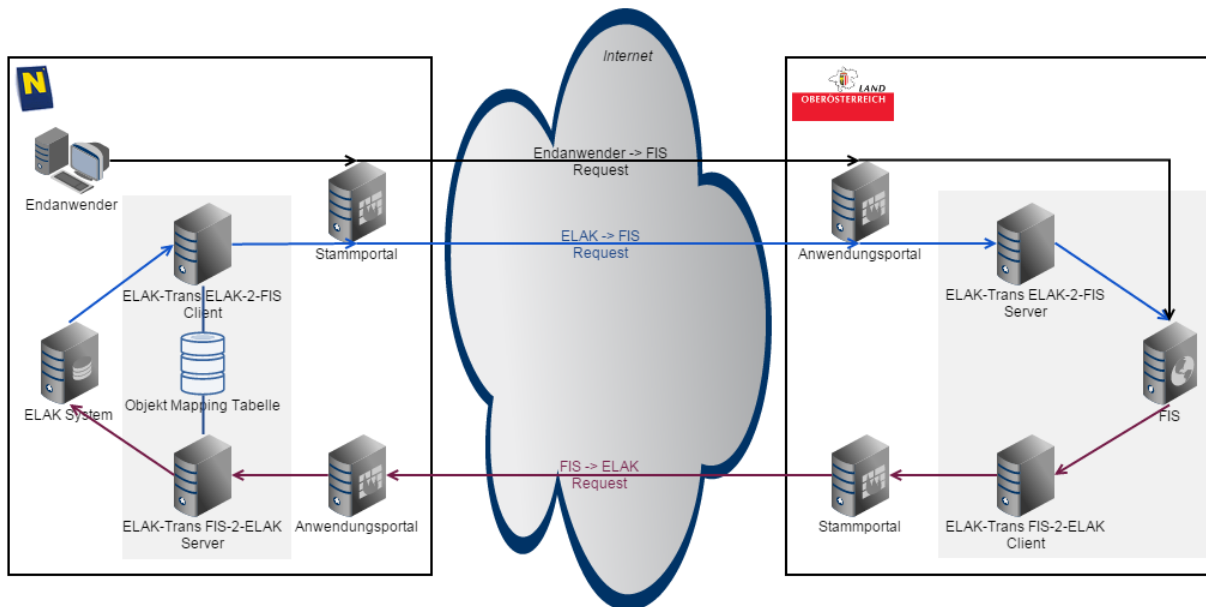
Beim Einsatz von ELAK-Systemen mit 4 Ebenen entfallen die meisten dieser potentiellen Probleme.

Beachten Sie dazu bitte auch

- die Abschnitte „3.1 EDIDOC“ und „4.1 EDIDOC - reduziertes Schema“ in diesem Dokument
- den Abschnitt „Aktenführung, Umgang mit den 4 Ebenen des EDIDOC-Pakets“ im Best Practice- Dokument

1. Architektur

Abbildung 1 - Verteilte ELAK-Trans Architektur mit PVP



1.1. Portalverbundprotokoll⁴

Das ELAK-Trans Protokoll ermöglicht die Kommunikation zwischen zentralen FIS und verteilten ELAK Systemen. Das Protokoll selbst beinhaltet jedoch keine Mechanismen zur Authentifizierung und Autorisierung von Benutzern. Diese Aufgaben werden vom R-Profil des zugrunde liegenden Portalverbundprotokolls (PVP) abgedeckt. ELAK-Trans spezifiziert somit lediglich die Art und Weise, in der aktenrelevante Daten zwischen den Systemen ausgetauscht werden können. Wie in Abbildung 1 dargestellt, übernehmen die Komponenten Stammportal und Anwendungsportal die Authentifizierung und Autorisierung bei Requests. Diese Architektur ist für den Anwendungsfall von verteilten Systemen verpflichtend. Soll ELAK-Trans innerhalb der eigenen Systemgrenzen eingesetzt werden, so können die Komponenten des PVP entfallen oder durch andere Komponenten ersetzt werden.

Die Verwendung des S-Profiles des PVP ist zwischen Endanwender und FIS eine Option. Für die Server-zu-Server-Kommunikation im Hintergrund zwischen FIS und ELAK-System ist für ELAK-Trans 3 weiterhin das R-Profil die einzige Option. (Das S-Profil des PVP ist für Server-zu-Server-Kommunikation Stand Februar 2016 noch nicht ausreichend definiert.)

1.2. ELAK-Trans Komponenten

In Abbildung 1 werden für die Kommunikation via ELAK-Trans jeweils ein ELAK-Trans FIS-2-ELAK Client mit entsprechendem ELAK-Trans FIS-2-ELAK Server sowie ein ELAK-Trans ELAK-2-FIS Client mit entsprechendem ELAK-Trans ELAK-2-FIS Server abgebildet. Jede dieser Komponenten kann, muss aber nicht, als eigenständige Softwarekomponente ausgeprägt werden. So ist es auch möglich, den ELAK-Trans FIS-2-ELAK Client direkt in einem FIS zu implementieren.

⁴ PVP-Spezifikation: <http://reference.e-government.gv.at/AG-IZ-PVP2-Version-2-1-2-2-1.2754.0.html>

Da als primäre Schlüssel der ELAK-Trans Pakete immer vom FIS vergebene IDs verwendet werden, muss auf der Seite des ELAK (für ELAK-Trans FIS-2-ELAK Server und ELAK-Trans ELAK-2-FIS Client) eine Objekt Mapping Tabelle geführt werden, in der das Mapping zwischen ELAK Objekt IDs und den FIS IDs festgehalten wird.

1.2.1. ELAK-Trans FIS-2-ELAK

ELAK-Trans FIS-2-ELAK spezifiziert die Kommunikation ausgehend vom FIS in Richtung ELAK System. Requests können dabei vom Anwender im FIS ausgelöst oder auch durch Scheduled Tasks an das ELAK System gesendet werden. Diese Kommunikationsrichtung ist der Standardfall.

Der Funktionsumfang dieser Kommunikation wird in „6.3 ELAK-Trans FIS-2-ELAK Client“ sowie „6.4 ELAK-Trans FIS-2-ELAK Server“ beschrieben.

1.2.2. ELAK-Trans ELAK-2-FIS

ELAK-Trans ELAK-2-FIS spezifiziert die Kommunikation ausgehend vom ELAK System in Richtung FIS. Diese Kommunikationsrichtung wird zur Ermöglichung einer bidirektionalen Kommunikation zwischen ELAK und FIS benötigt. Requests werden von Anwendern im ELAK oder automatisiert vom ELAK ausgehend an das FIS gesendet.

Der Funktionsumfang dieser Kommunikation wird in „6.1 ELAK-Trans ELAK-2-FIS Client“ sowie „6.2 ELAK-Trans ELAK-2-FIS Server“ beschrieben.

1.3. Schema Vereinfachung

Die Datenpakete, die durch ELAK-Trans ausgetauscht werden, basieren auf dem [EDIDOC Schema 1.0.0](#) und dem [Persondata Schema 2.0.0](#). Beide Schemata sind sehr umfangreich und wurden in dem Gesichtspunkt einer maximal vielfältigen Nutzung definiert. Durch diese Anforderung sind sehr vielseitig nutzbare Schemata entstanden, deren Befüllung nicht immer eindeutig zu interpretieren ist. Im Zuge der ELAK-Trans Spezifikation wurde vereinbart, welche Knoten befüllt und genutzt werden können/müssen. Um auch das Generieren der Klassen zu vereinfachen, wurden die Schema Dateien für das EDIDOC und das Persondata Schema angepasst. Die neuen Schema Dateien enthalten eine reduzierte und damit vereinfachte Version der Originalschemata und verwenden dieselben Namespaces. XML-Daten(-strukturen), die auf den ELAK-Trans-Schema-Dateien beruhen, sind nach wie vor valide gegenüber dem Originalschema.

1.4. Allgemeine Architekturmerkmale

1.4.1. Webservice

Die ELAK-Trans Funktionalität wird als Webservice ausgeprägt. Als Protokoll wird SOAP basierend auf HTTPS verwendet. Der Einsatz von MTOM⁵ wird empfohlen, ist aber nicht verpflichtend. Die WSDLs und der verwendete XML Schema Stack werden zusammen mit diesem Dokument veröffentlicht.

1.4.2. Online Betrieb

ELAK-Trans wurde für zustandslose SOAP Webservices spezifiziert. Jeder ELAK-Trans Aufruf ist als atomar zu betrachten und es gibt keine Möglichkeit mehrere ELAK-Trans Aufrufe in eine Transaktion zu bündeln. Durch diese Architekturentscheidung können die

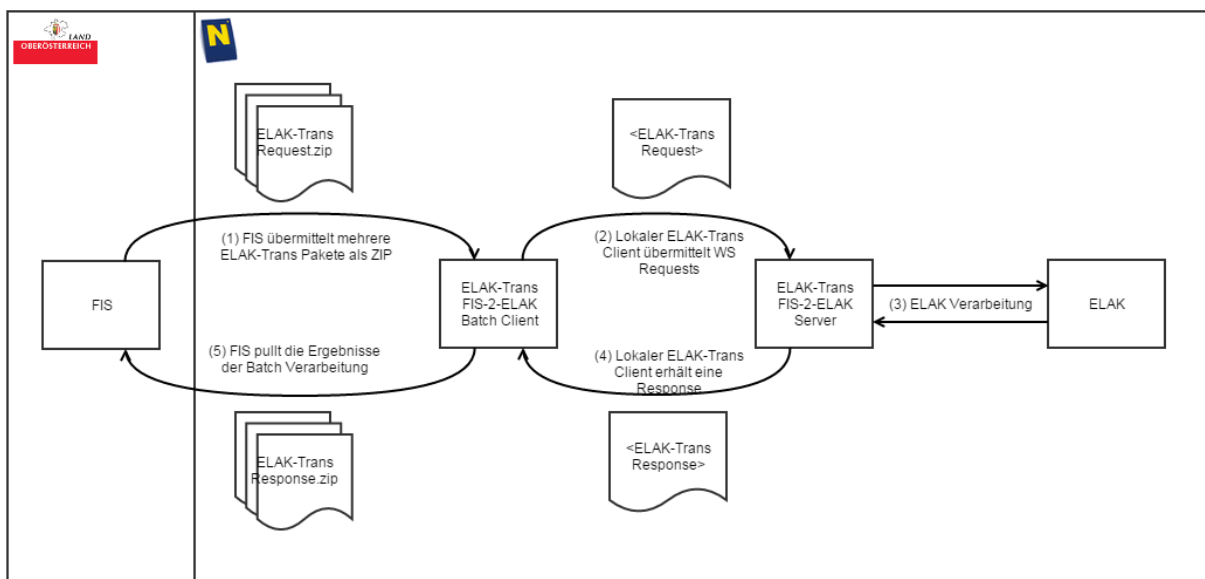
⁵ https://de.wikipedia.org/wiki/SOAP_Message_Transmission_Optimization_Mechanism bzw. <https://www.w3.org/TR/soap12-mtom/>

ELAK-Trans-Server Requests parallelisieren und ohne Abhängigkeiten zwischen den Requests abarbeiten.

1.4.3. Batch Betrieb

Da der ELAK-Trans Server für den Betrieb als SOAP Webservice ausgelegt ist, wird nicht empfohlen, die Webservices im Batch Betrieb zu nutzen. Dafür ist die Verarbeitung der SOAP Requests in verteilten Netzen zu langsam. Um einen Batch Betrieb trotzdem zu ermöglichen, wird empfohlen, die ELAK-Trans Pakete als ZIP Dateien per FTP zu übertragen. Ein lokaler ELAK-Trans Batch Client könnte die Verarbeitung der ZIP Pakete automatisiert parallelisieren und alle Ausgaben des ELAK-Trans Servers wieder als ZIP Pakete für die Rückübermittlung an das Ausgangssystem bereitstellen. Das ELAK-Trans Schema beinhaltet zu diesem Zweck einen Knoten, der die Abhängigkeiten der Pakete untereinander abbildet.

Abbildung 2 - Batch-Betrieb



Weitergehende Informationen dazu sind im Abschnitt „Massenverarbeitung“ des Best Practice – Dokuments zu finden.

2. Beziehungen zwischen Fachinformationssystem und ELAK

2.1. Schlüssel

Das FIS ist das führende System für die Vergabe der Schlüssel. Das FIS speichert keine technischen Schlüssel des ELAKs. (zB Objektadressen)

Die Zuordnung der FIS-ID's zu ELAK-ID's kann entweder

- in einer Datenbanktabelle des ELAK-Trans-Servers ("Objekt Mapping Tabelle") oder
- im ELAK erfolgen (zB in Form eines Fremdschlüssels pro Anwendung der beim Geschäftsobjekt hinterlegt ist)

Damit die Beziehung zwischen FIS und ELAK-System sichergestellt werden kann, sind für die jeweiligen Schlüsselwerte folgende Forderungen zu erfüllen:

- FIS-ID's müssen dauerhaft eindeutig und unveränderbar sein
- die ELAK-ID's sollten ebenfalls unveränderbar sein (alternativ sind im Falle einer Änderung der ELAK-ID's die Daten in einer etwaigen Objekt Mapping Tabelle des ELAK-Trans-Servers nachzuziehen)

Beachten Sie bitte auch den Abschnitt „3.3 Schlüsselhandling/Schlüsseleindeutigkeit“

2.2. Welche Daten werden wo abgelegt?

Grundregel:

- Fachdaten im FIS
- Aktenstrukturen im ELAK

In manchen Szenarien kann es Sinn machen, dies etwas aufzuweichen. Beispielsweise kann es aus Performance-Überlegungen sinnvoll sein, einzelne Metadaten der Aktenstruktur redundant im FIS abzulegen, um diese performant anzeigen zu können (z.B. Geschäftszahlen). Ebenso kann es Sinn machen, im ELAK bei der Aktenstruktur einen Hinweis auf den Querverweis zum FIS abzulegen.

Zu beachten ist jedoch, dass jede redundante Datenhaltung zu Synchronitäts-Problemen führen kann und allen Beteiligten klar sein sollte, welche Daten wann wo zu ändern sind bzw. welche Daten in welche Richtung automatisiert abgeglichen werden.

2.3. In welchem Umfang steuert das FIS die Aktenstruktur?

Dadurch, dass das FIS zum Erzeugen von Aktenstrukturen (mittels Kommunikation 5.2 FIS - > ELAK Kommunikation (FIS-2-ELAK), WS-Methode 5.2.1 SendData) die 4 Ebenen des EDIDOC-Pakets befüllt, nimmt das FIS wesentlichen Einfluss auf die Gestaltung dieser. Es steuert damit im Regelfall, welche Geschäftsstücke zu welchen Akten bzw. Geschäftsfällen zugeordnet werden.

Davon abweichende Logiken, welche die Zuordnung der Geschäftsstücke abseits der Befüllung des EDIDOC-Pakets auf Basis von Metadaten im EDIDOC-Paket umstrukturieren, sind grundsätzlich möglich, schränken den möglichen Integrationsumfang jedoch ein (siehe Best Practice - Dokument).

2.4. Adressierung von FIS bzw. ELAK-Systemen

Die (Webservice-)Adressen (URLs) der angesteuerten Systeme werden im Fachinformationssystem bzw. dem ELAK-Trans-Server geführt. Es wird davon ausgegangen, dass im Hintergrund PVP-Mechanismen verwendet werden.

Dies bedeutet, dass beispielsweise im FIS GISA die URL's der ELAK-Trans-Server der einzelnen Bundesländer und Magistrate hinterlegt sind. (Damit GISA auf die einzelnen dezentralen ELAK's zugreifen kann.)

Umgekehrt müssten im Fall einer bidirektionalen Kommunikation im jeweiligen ELAK-Trans-Server die URL's der ELAK-Trans ELAK-2-FIS Server hinterlegt sein. (Damit die jeweiligen dezentralen ELAK(-Trans-Server) Daten an das zentrale FIS übermitteln können.)

3. Allgemeine technische Festlegungen

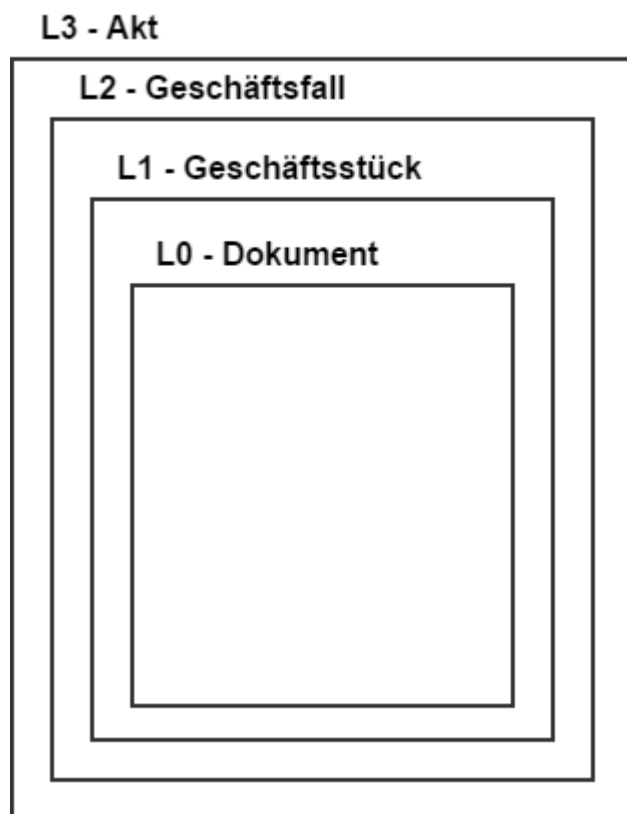
3.1. EDIDOC

ELAK-Trans verwendet für den Austausch von Aktenstrukturen die EDIDOC-Datenstrukturen, welche unter Anwendung des Blaupausen-Prinzips an die Anforderungen des Online-Datenaustausches zwischen FIS und ELAK-Systemen optimiert wurden. Die Anwendung des Blaupausen-Prinzips stellt a) die Validität zum EDIDOC-XML-Schema sicher und b) vereinfacht den Datenaustausch durch Reduktion und Präzisierung.

Die 4 Ebenen sind im Kontext der meisten ELAK-Systeme wie folgt zu verstehen:

- Ebene 3 (L3) = Akt
- Ebene 2 (L2) = Geschäftsfall
- Ebene 1 (L1) = Geschäftsstück
- Ebene 0 (L0) = Dokument

Abbildung 3 – EDIDOC-Paket



Das Wurzelement des EDIDOC-Schema ist der EdiaktType. Dadurch, dass EDIAKT der Vorgänger von EDIDOC war, im EDIDOC-Schema aber weiterhin der EdiaktType verwendet wird, lassen sich Namensverwirrungen nur schwer vermeiden.

3.2. Synchroner Aufrufe

ELAK-Trans verwendet zwischen FIS und ELAK(-Trans-Server) synchrone Webservice-Aufrufe.

Daraus folgt:

- Das Einlangen der Response beim Initiator der Aktion bedeutet, dass der Aufruf erfolgreich war.
- Im Fehlerfall wird ein SOAP-Fault geworfen. Der Absender des SOAP-Faults stellt sicher, dass die Atomarität des Aufrufs gewahrt wird. Mehr Informationen dazu im Abschnitt „3.5 Fehlerbehandlung“.
- Der Initiator der Aktion (zB das FIS) muss sich um Nicht-Verfügbarkeit des ELAK-(Trans-Servers) sowie um das Wiederanlaufen kümmern. (Beispiel: FIS gibt an den Benutzer eine Fehlermeldung zurück, dieser probiert es etwas später nochmal.)

Die dahinterliegende Kommunikation zwischen ELAK-Trans-Server und dem ELAK kann beliebig gestaltet sein. Beispielsweise können mit einem herstellereigenen Protokoll Daten zwischen ELAK und ELAK-Trans-Server ausgetauscht werden. Dies kann beinhalten, dass der ELAK zuerst synchron Daten an den ELAK-Trans-Server übermittelt, dieser asynchron Daten aus dem ELAK nachliest und abschließend den von ELAK-Trans standardisierten synchronen sendData-Aufruf zum FIS durchführt.

Wird im FIS Asynchronität im Sinne von "der Benutzer kann weiterarbeiten, obwohl die dahinterliegende Verarbeitung noch läuft" gewünscht, kann diese im FIS implementiert werden. (Dies gilt nicht nur für ELAK-Trans-Aufrufe, sondern für sämtliche länger laufenden Aktionen.)

Es ist abzuwägen, in welchem Verhältnis die für den Benutzer gewonnene Zeit im Vergleich zur daraus entstehenden technischen Komplexität steht.

1. Der Benutzer muss mgl. darüber informiert werden, dass seine Aktion erfolgreich oder fehlerhaft abgeschlossen wurde.
2. Fachentitäten müssen in der Zwischenzeit gesperrt werden, um einen konsistenten Zustand sicherzustellen.

Für Anforderungen der Massenverarbeitung werden im Best Practice - Dokument alternative Lösungsansätze aufgezeigt.

3.3. Schlüsselhandling/Schlüsseindeutigkeit

Wie in „2 Beziehungen zwischen Fachinformationssystem und ELAK“ beschrieben, ist das FIS das führende System für die Vergabe der Schlüssel. Diese werden als Referenced-Identifizier im EDIDOC-Paket bzw. im LayerXControlType übermittelt.

3.3.1. Wie sind die Referenced-Identifizier der Objekte der vier EDIDOC-Ebenen zu interpretieren?

Die Referenced-Identifizier sind innerhalb eines FIS nur im Kontext des FIS sowie im Kontext der vier Ebenen des EDIDOC-Pakets eindeutig. Sprich der Referenced-Identifizier der Ebene 2 ist nur zusammen mit dem Referenced-Identifizier der Ebene 3 eindeutig usw.

Daraus ergibt sich:

- Der ELAK-Trans-Server muss den Namen der Anwendung und die Schlüssel aller vier Ebenen konkatenieren, um zu einem UID zu kommen. (zB Application_L3RefID_L2RefID_L1RefID, ...)
- Das FIS darf beispielsweise folgendes schicken:
 - Das FIS darf beispielsweise folgendes schicken:
 - Request 1 mit Kombination
 - L3 Referenced-Identifizier: 1
 - L2 Referenced-Identifizier: 1
 - L1 Referenced-Identifizier: 1
 - Request 2 mit Kombination

- L3 Referenced-Identifizier: 2
- L2 Referenced-Identifizier: 2
- L1 Referenced-Identifizier: 1
- Durch diese beiden Request würden zwei gesonderte L1-Objekte entstehen. (Das L1-Objekt 1-1-1 sowie das L1-Objekt 2-2-1.)

Die Persistierung der Verknüpfung zwischen den technischen Identifiern des ELAK (zB Objektadressen) und den Referenced-Identifiern des FIS obliegt der Implementierung des ELAK-Trans-Servers. (zB a) Verwendung einer Objekt Mapping Tabelle in der DB des ELAK-Trans-Servers versus Ablage der Referenced-Identifizier im ELAK oder b) ein konkateniertes Feld in der Objekt Mapping Tabelle versus ein Feld pro Ebene in der Objekt Mapping Tabelle)

Folgende Konventionen gelten:

- Maximale Schlüssellänge pro individuellem Referenced-Identifizier: 36 Zeichen, dies entspricht der Länge einer GUID mit Bindestrichen (somit konkateniert 36 *4 Zeichen für L3-L0 plus 100 Zeichen Anwendung).

Als Best Practice wird empfohlen, GUID's für die Referenced-Identifizier zu verwenden.

3.3.2. Ausnahmen

- NUTZT ein FIS die Bidirektionalität UND
- werden aus dem ELAK L1-Objekte (Geschäftsstücke = "Aufgaben") an das FIS übergeben UND
- kann das FIS für diese L1-Objekte nicht sofort eine L3 (=Aktenzuordnung) und/oder L2 (=Geschäftsfallzuordnung) durchführen,

DANN muss das FIS eindeutige L1-Referenced-Identifizier vergeben.

(Da zu diesem Zeitpunkt noch keine L3 Referenced-Identifizier und L2 Referenced-Identifizier bekannt sind. Das FIS erstellt eine Aufgabe und gibt als L3 sowie L2 - Referenced Identifizier 00000000-0000-0000-0000-000000000000 zurück. Der ELAK-Trans FIS-2-ELAK Server speichert in der Objekt Mapping Tabelle L3-00000000-0000-0000-0000-000000000000, L2-00000000-0000-0000-0000-000000000000, L1-FIS-ID.)

3.3.3. Was muss alles einen Referenced-Identifizier haben?

- Die L3-L0-Objekte des EDIDOC-Pakets.

3.3.4. Wo werden die Referenced-Identifizier beim sendData übergeben?

- Im LayerControl als LayerControl/LayerX/ReferencedIdentifizier/Identification.
- Bewusst redundant in der EDIDOC/EDI AKT-Struktur als Ediakt/Payload/Layer3/ReferencedIdentifizier/Identification usw.
- Kommt es zu Abweichungen zwischen LayerControl und EDIDOC/EDI AKT-Struktur wirft der Server den Fehler INVALID_CONSISTANCY.

3.3.5. Wo werden FIS-ID's von Personen, Adressen und Standorten übergeben?

- Die FIS-ID wird jeweils als Identification vom Typ "ID" mitgegeben.

```
<ns3:Person xsi:type="ns3:PhysicalPersonType" x
- <ns3:Identification>
  <ns3:Value>FIS-ID-Person-1</ns3:Value>
  <ns3:Type>ID</ns3:Type>
</ns3:Identification>
```

- Bei FIS-2-ELAK-Kommunikation

- Bei Personen-Objekten innerhalb der Adressaten (Participants) der L3-L1-Objekte des EDIDOC-Pakets wird EMPFOHLEN, die FIS-ID der Person zu übergeben.
(Wird zB von jenen ELAK-Trans-Implementierungen genutzt, welche statt auf Freitext-Adressaten auf Adressaten-Stammdatensätze abbilden und diese aus den Geschäftsobjekten referenzieren. Detaillierte Beschreibung dazu unter „4.2 PersonData - reduziertes Schema plus Erweiterungen“.)
- Bei Adress-Objekten innerhalb der Adressaten (Participants) der L3-L1-Objekte des EDIDOC-Pakets KANN die FIS-ID der Adresse übergeben werden. (zB Wohnadresse, Zustelladresse)
- Bei Adress-Objekten innerhalb der Standorte (Locations) der L3-L1-Objekte des EDIDOC-Pakets KANN die FIS-ID der Adresse übergeben werden. (Adresse des Ortsbezuges; zB Standort)
- Bei ELAK-2-FIS-Kommunikation
 - Keine. (Weil bei Freitext-Adressaten im ELAK keine ID's mitgespeichert werden können.)

3.4. PVP Application Chaining

ELAK-Trans selbst enthält keine Mechanismen, um Informationen zum Benutzer in dessen Kontext die Aktion ausgeführt wird, mitzugeben. Stattdessen ist der im Portal-Verbund-Protokoll definierte Mechanismus des Application Chaining anzuwenden. (Siehe Dokument "Portalverbundprotokoll Version 2 - eGovernment Attribute Profile". In der zum Zeitpunkt der Redaktion dieser Spezifikation gültigen PVP 2.1.2-Spezifikation ist in diesem Dokument im Abschnitt 5 das "Application Chaining" beschrieben.)

Mit Hilfe der Chained Token - Informationen kann beispielsweise beim Erstellen der Aktenstrukturen der im FIS aktive Benutzer als Ersteller der Aktenstrukturen vermerkt werden.

Webservice-Frameworks bieten Mechanismen an, um Zugriff auf diese im SOAP-Header angesiedelten Informationen zu erlangen. (Im Java-Kontext beispielsweise <https://docs.oracle.com/javaee/5/api/javax/xml/ws/handler/soap/SOAPHandler.html>)

3.5. Fehlerbehandlung

ELAK-Trans verwendet den von W3C standardisierten SOAP Fault - Mechanismus (https://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383507), um im Fehlerfall sowohl technische, als auch für den Benutzer sprechende Fehlermeldungen zurückzugeben. (Schlägt beispielsweise ein Aufruf vom FIS zum ELAK-(Trans-Server) fehl, weil der Akt durch einen Benutzer gesperrt ist, sollte diese Fehlermeldung an den FIS-Benutzer durchgereicht werden.)

Das Atomaritäts-Prinzip der ELAK-Trans-Methoden bedeutet, dass für SOAP-Aufrufe das Prinzip gilt, dass diese ganz oder gar nicht ausgeführt werden. Sprich im Fall eines Fehlers und daraus resultierenden SOAP-Faults muss ein Rollback aller im Rahmen dieses SOAP-Aufrufs bereits durchgeführter (ELAK-Trans-Server-internen) Teiloperation durchgeführt werden. Werden vom ELAK-System Transaktionen nicht unterstützt, so hat der ELAK-Trans-Server zumindest dafür zu sorgen, dass im ELAK keine inkonsistenten Datensätze bestehen bleiben (z.B. durch Löschen oder Stornieren von unvollständigen Daten).

Der ELAK-Trans-Server darf - um Transparenz und Nachvollziehbarkeit zu gewährleisten - keine fehlerhaften Requests selbstständig korrigieren. Ist der Request fehlerhaft, wird im SOAP-Fault die Fehlerbeschreibung zurückgegeben.

Dies impliziert weiters, dass eine Bufferung von Requests durch den ELAK-Trans Server während einer Nicht-Verfügbarkeit des ELAKs nicht zulässig ist.

Eine umfangreiche Betrachtung zu Fehlerszenarien und den Umgang damit findet sich im Best Practice - Dokument.

4. Schema-Beschreibung

In den technischen XML Schema Definitionen (XSD-Dateien) befinden sich umfassende Annotationen, welche die einzelnen Elemente des Schemas erklären. Für Detail-Fragen sollten daher die XSD-Dateien konsultiert werden.

Das Spezifikations-Dokument versucht, einen Überblick über die Datenstrukturen zu geben und Abweichungen/Präzisierungen/Ergänzungen gegenüber den zugrundeliegenden EDIDOC- und PersonData-Schemata genauer zu beschreiben.

XML-Beispiele finden sich im Best Practice – Dokument in den Abschnitten „3.5. Detailbeschreibung incl. SOAP-Messages und Objekt-Mapping-Tabellen-Zustände zu ausgewählten UseCases“ und „10. XML-Beispiele“.

XSD-Diagramme wurde mit <http://regis.cosnier.free.fr/> erstellt.

4.1. EDIDOC - reduziertes Schema

EDIDOC verwendet den Namespace

xmlns:edi = "http://reference.e-government.gv.at/namespace/edidoc/20130808#"

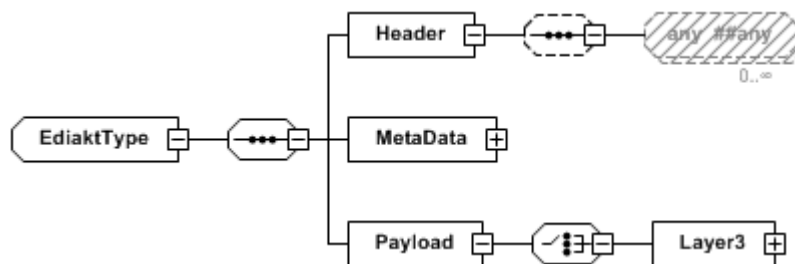
Folgende weitere Namespaces werden verwendet:

- xmlns:p = "http://reference.e-government.gv.at/namespace/persondata/20020228#"
- xmlns:xs = "http://www.w3.org/2001/XMLSchema"

Weiterführende Informationen zum EDIDOC-Schema abseits der Anwendung in ELAK-Trans: <http://reference.e-government.gv.at/EDIAKT-EDIDOC.599.0.html>

Konvention bei der Verwendung von EDIDOC im Kontext von ELAK-Trans ist, dass Ebenen von oben (L3) weg durchgängig befüllt sein müssen, somit z.B. das Schicken ausschließlich eines L1 nicht erlaubt ist. Werden keine Geschäftsfälle verwendet, muss das L2-Objekt als leeres Objekt trotzdem vorhanden sein.

4.1.1. EdiaktType



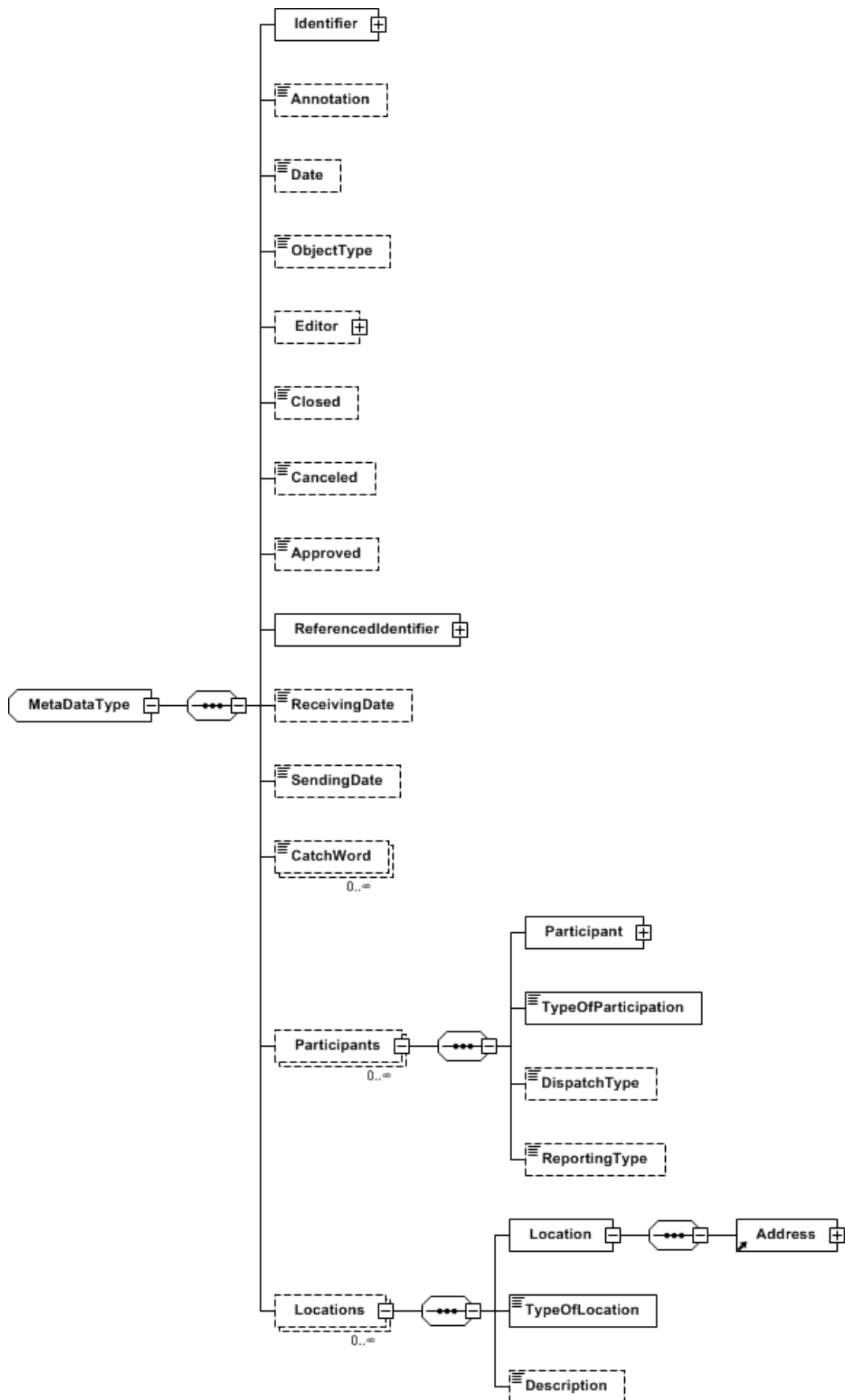
edi:EdiaktType	Description	Type	Occur
Header	Um EDIDOC-konform zu sein, muss dieses Element übermittelt werden. Für ELAK-Trans soll es daher zwar vorhanden, aber leer sein.	edi:Header-Type	1

edi:EdiaktType	Description	Type	Occur
MetaData	Um EDIDOC-konform zu sein, muss ein Metadata.Identifier.Identification-Knoten angelegt werden. In ELAK-Trans soll dieser daher auch an dieser Stelle vorhanden, aber leer sein.	edi:MetaData-Type	1
Payload Layer3	- Aktenstruktur	edi:Layer3Type	1

4.1.2. HeaderType

Dieser Typ hat zwar gegenüber dem entsprechenden Originaltyp aus EDIDOC keine Elemente mehr, muss aber der Kompatibilität halber im Schema bleiben.

4.1.3. MetaDataType



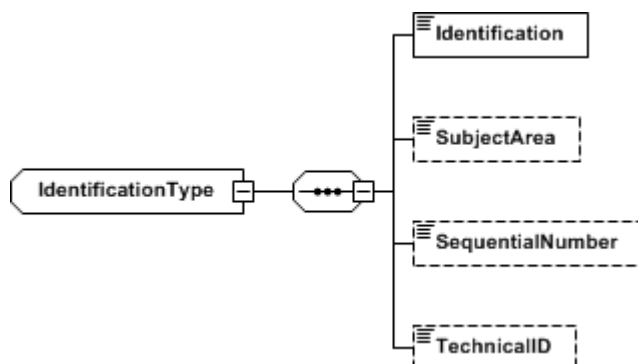
edi:Meta-Data Type	Description	Type	Occur	L 3	L 2	L 1	L 0
Identifizier	Der Identifizier wird für die Identifikation des Layers im ELAK verwendet. Für die ELAK-Trans-Kommunikation gilt allerdings der ReferencedIdentifizier als eindeutiger Schlüssel. Ist ein Identifizier noch nicht bekannt, so muss der Knoten trotzdem erstellt und mit einer leeren Identification befüllt werden, um dem EDIDOC-Schema zu entsprechen.	edi:Identifizierung-Type	1	x	x	x	x
Annotation	Notizen/Anmerkungen zum entsprechenden Layer. Der Knoten wird optional verwendet.	xs:string	0-1	x	x	x	x
Date	Erstellungsdatum. Verwendung muss bei der Implementierung zwischen FIS und ELAK-Trans System geklärt werden. Der Knoten lässt sich nur auf dem Weg ELAK => FIS sinnvoll einsetzen.	xs:date-Time	0-1	x	x	x	x

edi:Meta-Data Type	Description	Type	Occur	L 3	L 2	L 1	L 0
ObjectType	Wird nur auf der Ebene des Layer0 verwendet, beispielsweise zur Kennzeichnung von Hauptdokument und Anhang, oder zur Kennzeichnung eines Bescheids. Die Kennzeichnung der Hauptdokumente erfolgt zusätzlich im Layer1 mit dem MainDocument-Knoten, der auf die ID des entsprechenden Hauptdokuments verweist. Werte werden bei der Implementierung zwischen FIS und ELAK-Trans Schnittstellen vereinbart.	xs:token	0-1				x
Editor	SachbearbeiterIn wird in den Fällen vom ELAK befüllt, könnte aber auch vom FIS befüllt werden, sofern dort eine Liste der Sachbearbeiter geführt wird. Ansonsten wird in der Richtung FIS => ELAK Kommunikation der PVP-Benutzer als SachbearbeiterIn angenommen.	p:Person-Data Type	0-1	x	x	x	
Closed	Abgeschlossen. Bei ReadData ist die Semantik insofern klar, als dieser Knoten immer dann befüllt ist, wenn es in der LayerResponseControl so gewünscht ist. Bei SendData ist die Semantik ähnlich; wenn im LayerSendControl entsprechend angegeben, so ist eine Aktualisierung des Status erwünscht.	xs:boolean	0-1	x			

edi:Meta-Data Type	Description	Type	Occur	L 3	L 2	L 1	L 0
Canceled	Storniert. Bei ReadData ist die Semantik insofern klar, als dieser Knoten immer dann befüllt ist, wenn es in der LayerResponseControl so gewünscht ist. Bei SendData ist die Semantik ähnlich; wenn im LayerSendControl entsprechend angegeben, so ist eine Aktualisierung des Status erwünscht.	xs:boolean	0-1	x	x	x	x
Approved	Genehmigt. Bedeutung bei readData (Primär gedacht für FIS-2-ELAK-Kommunikation) sowie sendData (Primär gedacht für ELAK-2-FIS-Kommunikation): Element vorhanden und TRUE --> genehmigt Element vorhanden und FALSE --> abgelehnt Element nicht vorhanden --> keine Änderung / Relevanz	xs:boolean	0-1			x	
Referenced-Identifizier	Die FIS-ID muss bei jeder Übermittlung erzeugt werden. Gegebenenfalls - analog zum Identifizier - mit einem leeren Identification-Knoten.	edi: Identification-Type	1	x	x	x	x
Receiving-Date	Das Eingangsdatum soll verwendet werden. Der Knoten sollte aber nur beim Layer1 verwendet werden.	xs:date-Time	0-1			x	
CatchWord	Wird nicht von allen ELAK-Systemen unterstützt. Bei Fabasoft-Systemen entsprechen die CatchWords den Begriffen.	xs:string	0-n	x	x	x	x

edi:Meta-Data Type	Description	Type	Occur	L 3	L 2	L 1	L 0
Participants	Adressaten	edi:ParticipantsType	0-n	x	x	x	
Locations	Ortsbezüge	edi:LocationsType	0-n	x	x	x	

4.1.4. IdentificationType



edi:Identification-Type	Description	Type	Occur
Identification	Bei ReferencedIdentifier ID des Layers im FIS, bei Identifier gesamte Akten-/Geschäftszahl des Layers im ELAK.	xs: token	1
SubjectArea	Kurzbezeichnung des Sachgebiets. Gibt an, zu welchem Themenbereich der Akt gehört. Nur beim Lesen aus dem ELAK und nur für den Layer3 relevant. In allen anderen Fällen nicht verwendet. Nur beim Identifier, nicht beim ReferencedIdentifier.	xs: token	0-1
SequentialNumber	Sekundärer Identifier für die Kommunikation, falls eine alternative Findhilfe erforderlich ist. Beispielsweise für Migrationsszenarien. Macht nur im Kontext des ReferencedIdentifier Sinn, und wird im Kontext des Identifier nicht verwendet werden.	xs: token	0-1

edi:Identification-Type	Description	Type	Occur
TechnicalID	URL zum Abspringen in den entsprechenden Layer. Die Befüllung sieht folgendermaßen aus; der ELAK sendet eine absolute URL, das FIS sendet eine relative URL ab dem Stammpfad der Applikation. Beispiel aus GISA: Die Applikation wird in Tirol mit https://portal.tirol.gv.at/at.gv.wien.gisa adressiert. Der relative Pfad der Suchseite innerhalb GISA lautet z.B. searchPage.aspx?searchGewerbe=12345	xs:token	0-1

4.1.5. ParticipantsType

edi:Participants-Type	Description	Type	Occur
Participant	Adressat	p:PersonDataType	1
TypeOf-Participation	Einbringer, Empfänger, ...; sprich Art des Adressaten	xs:string	1
DispatchType	Abfertigungs-/Versandart	xs:string	0-1
ReportingType	"Abschriftlich" oder leer. Weitere Werte sind bei der Implementierung festzulegen. Die Auswahl ist verpflichtend im FIS zu implementieren, wann auch der DispatchType genutzt wird.	xs:string	0-1

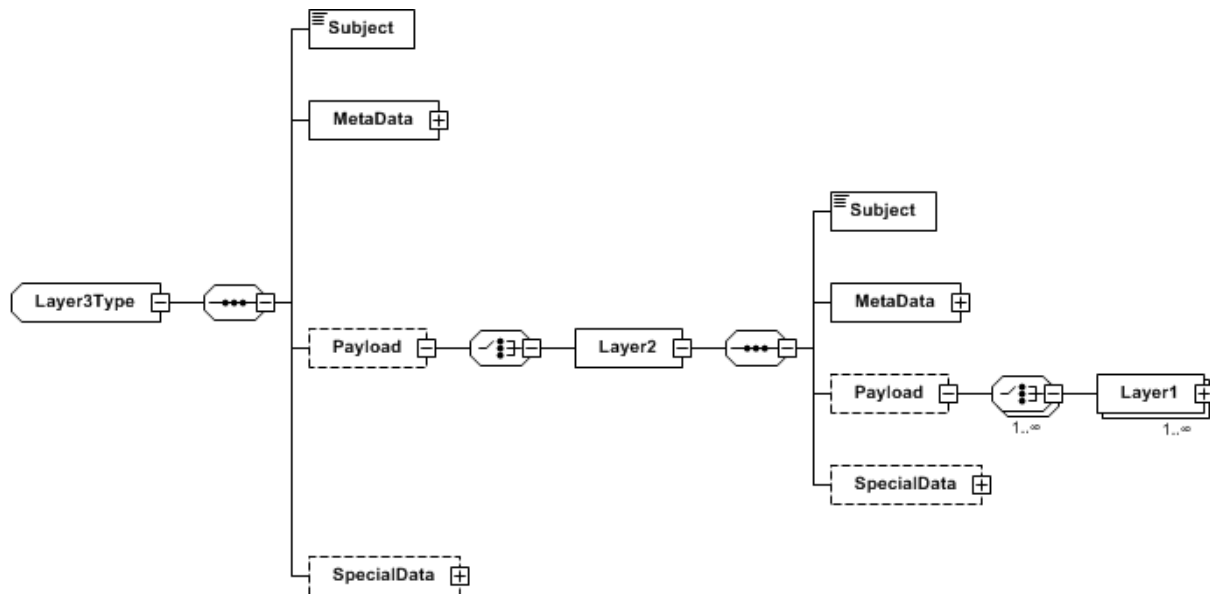
4.1.6. LocationsType

edi:LocationsType	Description	Type	Occur
Location		edi:LocationType	1
TypeOfLocation	Ein Typ ist bei jeder Location erforderlich. Diese müssen bei der Implementierung abgestimmt werden.	xs:string	1
Description		xs:string	0-1

4.1.7. LocationType

edi:LocationType	Description	Type	Occur
Address	Hier sind nur Instanzen des Typs TypedPostalAddressType zulässig.	p:Address	1

4.1.8. Layer3Type, Layer2Type

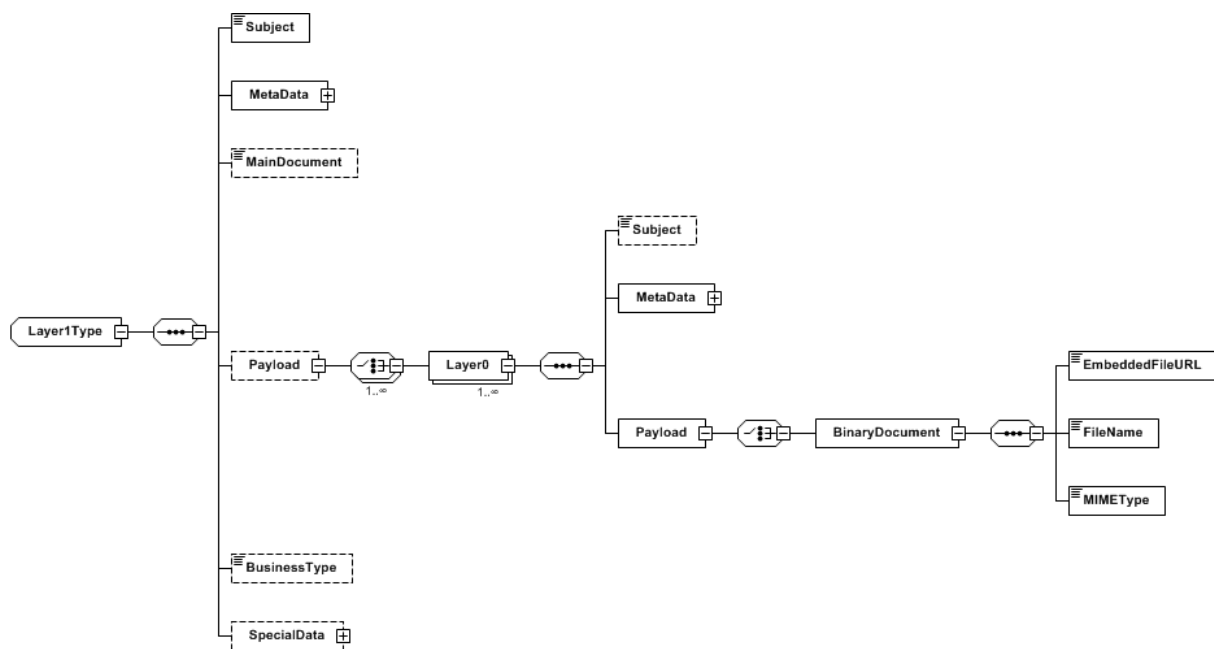


edi:Layer3Type	Description	Type	Occur
Subject	Der Betreff des Aktes ist für ELAK-Trans verpflichtend. Wird Layer2 nicht verwendet ist das Subject des Layer2Type mit einem Leerstring zu befüllen (<Subject></Subject>).	xs:string	1
MetaData		edi:MetaDataType	1
Payload/Layer2	Die ELAK-Trans-Vereinbarung erlaubt im SendDataInputObject nur einen Geschäftsfall zu übermitteln. Im Rahmen der Response (zB ReadData) können (wenn dies angefordert wurde) alle Geschäftsfälle eines Aktes zurückgegeben werden. Werden keine Geschäftsfälle verwendet, dann bleiben die Metadaten des Layer2 einfach leer.	edi:Layer2Type	0-1/1-n

edi:Layer3Type	Description	Type	Occur
SpecialData	Das Element bleibt erhalten, sollte aber nach Möglichkeit nicht verwendet werden.	edi:SpecialDataType	

edi:Layer2Type	Description	Type	Occur
Subject	Der Betreff des Geschäftsfalls ist für ELAK-TRANS verpflichtend.	xs:string	1
MetaData		edi:MetaDataType	1
Payload/Layer1	Geschäftsstück(e) bzw. Subzahl(en). Die ELAK-Trans-Vereinbarung erlaubt, im SendDataInputObject nur ein Geschäftsstück zu übermitteln. Im Rahmen der Response (zB ReadData) können (wenn dies angefordert wurde) alle Geschäftsstücke eines Aktes/Geschäftsfalls zurückgegeben werden.	edi:Layer1Type	0-1/1-n
SpecialData	Das Element bleibt erhalten, sollte aber nach Möglichkeit nicht verwendet werden.	edi:SpecialDataType	

4.1.9. Layer1Type, Layer0Type, PayloadType



edi: Layer1Type	Description	Type	Occur
Subject	Der Betreff des Geschäftsstücks ist für ELAK-Trans verpflichtend.	xs:string	1
MetaData		edi:MetaDataType	1
Main-Document	<p>Um das Hauptdokument zu kennzeichnen, wird das Attribut ID im Layer0Type referenziert. Sobald ein Layer0 enthalten ist, muss dieser Knoten befüllt sein. (Es gibt Fälle, in denen das MainDocument nicht befüllt werden kann. Dies ist zB der Fall wenn eine Anlage (Dokument) zu einem bestehenden Geschäftsstück hinzugefügt wird.)</p> <p>Auch wenn nur ein Dokument übergeben wird. IDREF wird bei manchen Web Service Frameworks auf Object statt auf String aufgelöst; Als Wert darf dann nicht die ID des Objekts, sondern ein Objekt mit der ID gesetzt werden. Man muss also ein Layer0Type-Objekt setzen und bekommt beim Lesen auch ein Layer0Type-Objekt. Im XML steht trotzdem nur <code><ns2:MainDocument>ID</ns2:MainDocument></code></p>	xs:IDREF	0-1

edi: Layer1Type	Description	Type	Occur
Payload/ Layer0	Schriftstück(e)	edi:Layer0Type	0-1/1-n
BusinessType	Typ des Geschäftsstücks	xs:token	0-1
SpecialData	Das Element bleibt erhalten, sollte aber nach Möglichkeit nicht verwendet werden.	edi:SpecialData- Type	0-1

edi:Layer0Type	Description	Type	Occur
Subject		xs:string	1
MetaData	Um das Hauptdokument zu kennzeichnen, wird das Attribut ID im Layer1Type referenziert. Sobald ein Layer0 enthalten ist, muss dieser Knoten befüllt sein. Auch wenn nur ein Dokument übergeben wird.	edi:MetaDataType	1
Payload	Dokument. Bei SendData im Regelfall befüllt. Bei vom "ELAK-Trans FIS-2-ELAK-Server" zurückgelieferten Datenpaketen abhängig von den in Layer0ResponseControl getroffenen Einstellungen.	edi:PayloadType	0-1
id (Attribut)	Für Referenzierung aus Layer1Type / MainDocument.	xs:ID	1

edi:PayloadType	Description	Type	Occur
BinaryDocument/ EmbeddedFileURL	Eindeutiger Dateiname, kann dem Element FileName entsprechen, muss es aber nicht. Es ist jedenfalls die Referenz auf ein Dokument im ZIP-File des "Documents"-Knoten innerhalb des SendDataInputObject. Die Verbindung zwischen BinaryDocument/EmbeddedFileURL- und	xs:stoken	1

edi:PayloadType	Description	Type	Occur
	Documents-Knoten ist im Anwendungsfall "6.4.16 Dokument erzeugen" beschrieben.		
BinaryDocument/ FileName	Ausgegebenener Dateiname	xs:token	1
BinaryDocument/ MIMEType	MIME-Type des übergebenen Dokuments	xs:token	1

4.1.10. SpecialDataType

edi:SpecialDataType	Description	Type	Occur
XMLContent			1
PreviewStylesheet			0-1

4.2. PersonData - reduziertes Schema plus Erweiterungen

PersonData verwendet den Namespace

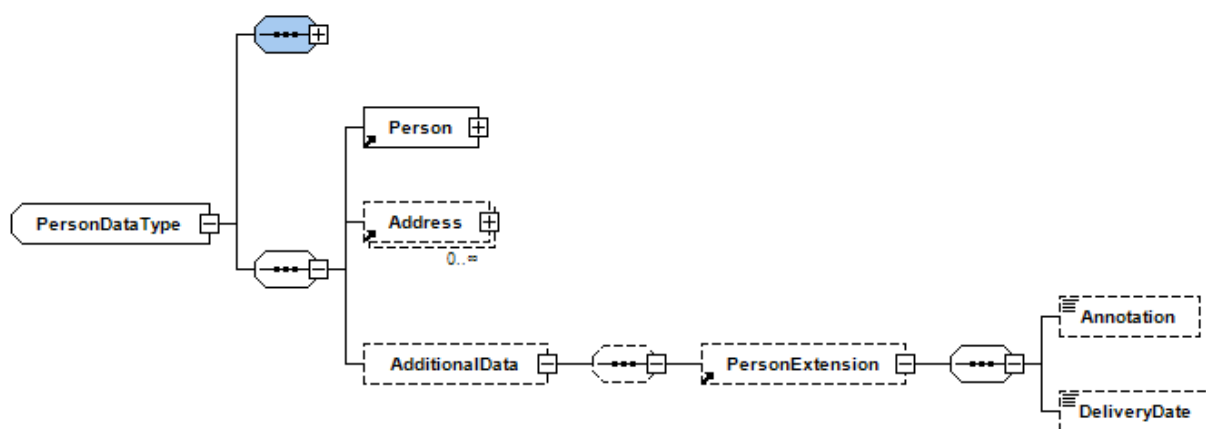
xmlns:p = "http://reference.e-government.gv.at/namespace/persondata/20020228#"

Die ELAK-Trans-Erweiterungen dazu befinden sich im Namespace

xmlns:pe="http://reference.e-government.gv.at/namespace/elaktrans/persondataext/20160401#"

Weiterführende Informationen zum PersonData-Schema abseits der Anwendung in ELAK-Trans: http://reference.e-government.gv.at/Q-KA_XML-Strukturen_fuer_Perso.614.0.html

4.2.1. PersonDataType



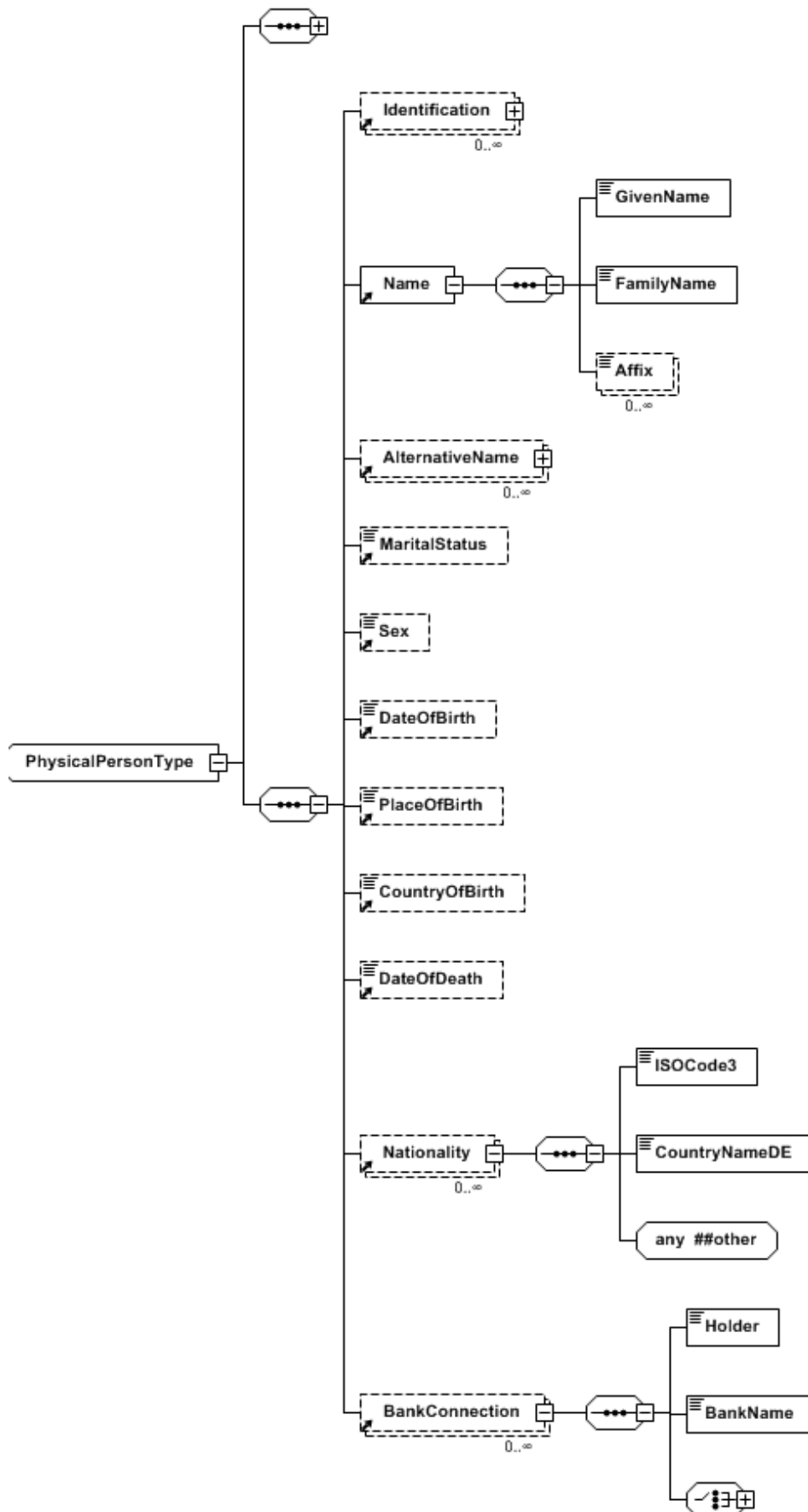
pe:Person- DataType	Description	Type	Occur

pe:Person-DataType	Description	Type	Occur
Person	AbstractPersonType ist nur wegen der Vererbung notwendig. Nicht direkt verwenden, sondern in ELAK-Trans entweder PhysicalPersonType oder CorporateBodyType.	Abstract-PersonType	1
Address	AbstractAddressType ist nur wegen der Vererbung notwendig. Nicht direkt verwenden, sondern in ELAK-Trans entweder TypedPostalAddress, InternetAddress oder TelephoneAddress. Auch PostalAddress ist nicht erlaubt.	Abstract-Address-Type	0-n
AdditionalData	Zusätzliche ELAK-Trans-spezifische Attribute. Diese Konstruktion garantiert folgende Punkte: a) Im Sinne des Blaupausenkonzepts lässt die Spezifikation nur genau festgelegte Knoten im Any-Bereich zu; damit fallen Interpretationsspielräume weg. b) Mit dieser Einschränkung validieren die Pakete nach wie vor gegen den allgemeinen PersonData-Standard mit dem xs:any Element.	pe:Person-Extension	0-1

4.2.2. PersonExtensionType

pe:Person-Extension-Type	Description	Type	Occur
Annotation	Bemerkung; analog zu den Layern der EDIDOC-Pakete gibt es auch hier eine Bemerkung.	xs:string	0-1
SendingDate	Abfertigungs-/Versanddatum als optionale Metainformation vom ELAK zum FIS.	xs:dateTime	0-1
DeliveryDate	Wird ein Schreiben im ELAK per RSa, RSb,... zugestellt, und ein Zustelldatum soll an das FIS kommuniziert werden, so kann dies pro Empfänger an dieser Stelle übermittelt werden.	xs:dateTime	0-1

4.2.3. PhysicalPersonType, PersonNameType, NationalityType



pe:Physical- PersonType	Description	Type	Occur	Restriction
Identification		pe:IdentificationType	0-n	
Name		pe:PersonNameType	1	
AlternativeName	Die Erweiterung im Schema ist etwas komplex, falls gewünscht, einfach dort bzw. einfach in den generierten Proxy-Klassen nachsehen.	pe:PersonNameType (etwas erweitert)	0-n	
MaritalStatus		xs:token	0-1	"single" "married" "divorced" "widowed"
Sex		xs:token	0-1	"male" "female" "unknown"
DateOfBirth		pe:DateOfBirthType	0-1	
PlaceOfBirth	Geburtsort	xs:token	0-1	
CountryOfBirth	Geburtsland	xs:token	0-1	
DateOfDeath		pe:DateOfDeathType	0-1	
Nationality		pe:NationalityType	0-n	
BankConnection		pe:BankConnection	0-n	

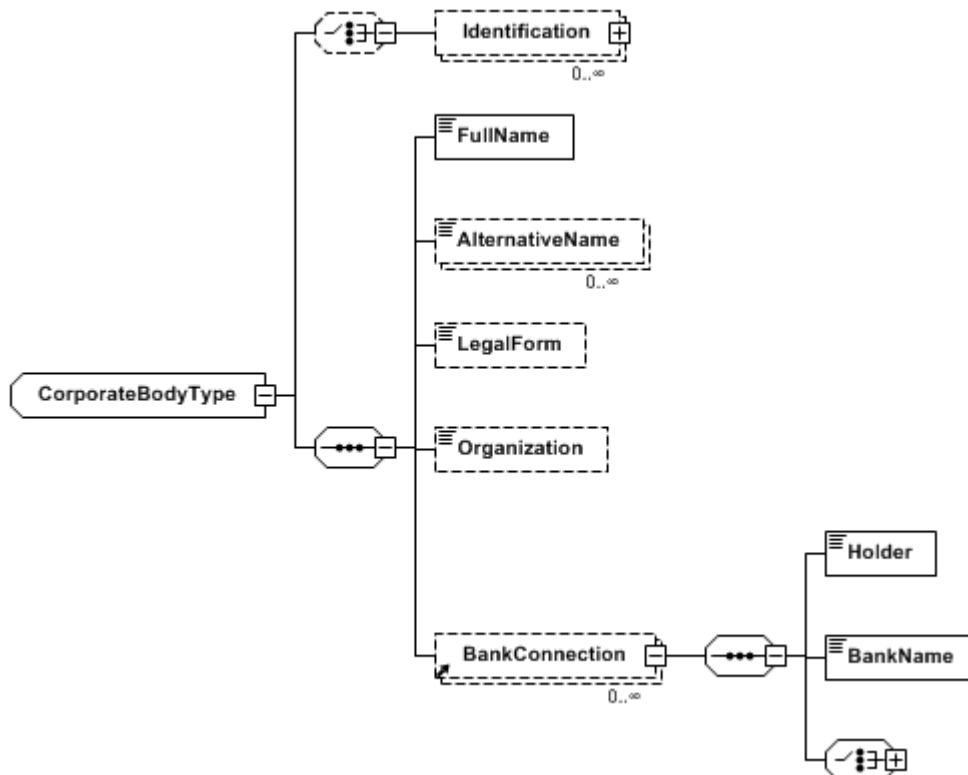
pe:Person- NameType	Description	Type	Occur	Restriction
--------------------------------	--------------------	-------------	--------------	--------------------

pe:Person-NameType		Description	Type	Occur	Restriction
GivenName		Für ELAK-Trans werden hier sämtliche Vornamen in einer formatierten Zeichenkette zusammengefasst.	xs:string	1	
FamilyName		Für ELAK-Trans werden hier sämtliche Familiennamen in einer formatierten Zeichenkette zusammengefasst.	xs:string	1	
Affix		Vor- und nachgestellte Titel, jeweils einzeln mit Typ und Position; die letztere Information ist jedenfalls für Fabasoft-Kunden relevant.	xs:string	0-n	
	type	Wird keine Typisierung des Titels mitgegeben, so wird der Titel 1:1 in die Anwendung (FIS bzw. ELAK je nach Richtung) übernommen. Wenn eine Typisierung mitgegeben wird, hat der Server die Möglichkeit, bestimmte nicht notwendige oder erwünschte Titel herauszufiltern (z.B. man will nur akademische Titel).	xs:string	0-1	"academicGrade" "aristocraticPrefix" "aristocraticTitle" "familyNamePrefix" "familyNameSuffix" "formOfAddress" "generation" "qualification"
	position	Information, ob der Titel vor- oder nachgestellt ist.	xs:string	0-1	"prefix" "suffix"

pe:Nationality-Type		Description	Type	Occur	Restriction
ISOCODE3			xs:string	1	max. 3 Zeichen
CountryNameDE			xs:string	1	

pe:Nationality-Type	Description	Type	Occur	Restriction
##other	Dieser Knoten muss wegen der Schema-Konformität erzeugt werden, darf aber nicht befüllt werden.	xs:any	1	

4.2.4. CorporateBodyType



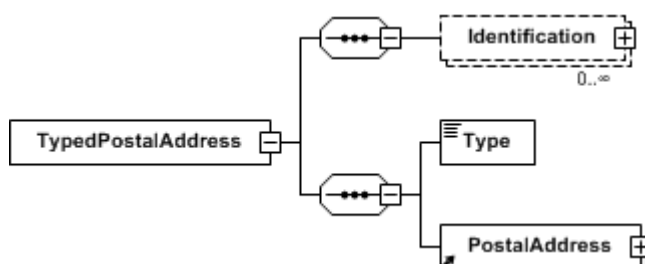
pe:Corporate-BodyType	Description	Type	Occur
Identification		pe:Identification-Type	0-n
FullName	Voller, z.B. laut Firmenbuch rechtlich relevanter Name der juristischen Person	xs:dateTime	0-1
AlternativeName	Alternative Namen der juristischen Person (z.B. Abkürzungen, Kurznamen, Synonyme, ...)	xs:token	0-n

pe:Corporate-BodyType	Description	Type	Occur
LegalForm	Rechtsform (z.B. AG, OHG, ...) als URI; das ist, falls verwendet, im jeweiligen Integrationsprojekt zu vereinbaren.	xs:anyURI	0-1
Organization	Organisationseinheit, etwa Abteilung, Bereich, Fachbereich.	xs:token	0-1
BankConnection		pe:Bank-ConnectionType	0-n

4.2.5. BankConnectionType

pe:BankConnectionType	Description	Type	Occur
Holder	Inhaber	xs:string	1
BankName		xs:string	1
InternationalBankConnection			1
	IBAN	xs:string	1
	BIC	xs:string	1

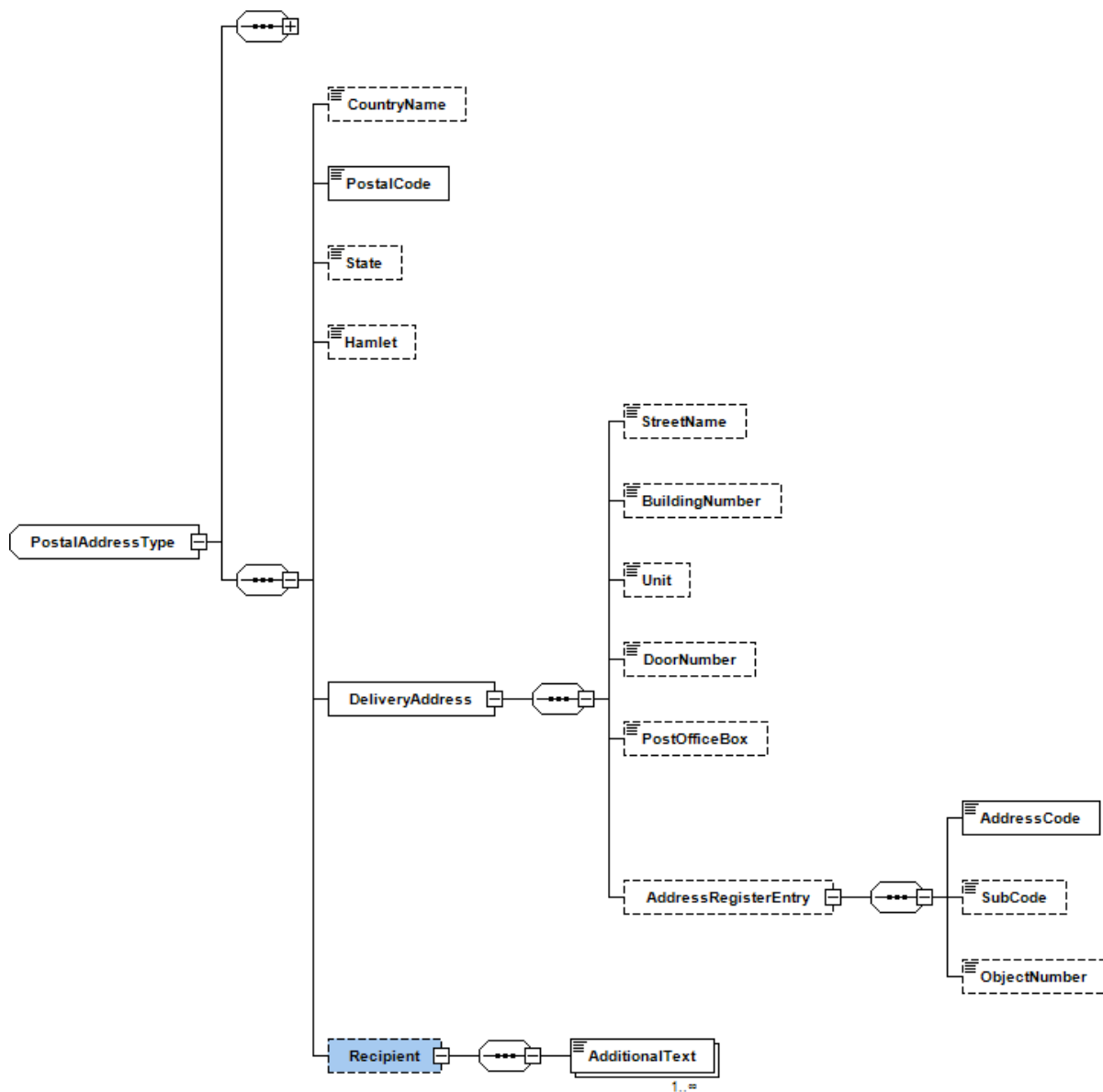
4.2.6. TypedPostalAddressType



pe:Typed-Postal-AddressType	Description	Type	Occur	Restriction
Identification	IDs.	Identification-Type	0-n	

pe:Typed-Postal-AddressType	Description	Type	Occur	Restriction
Type	Für ELAK-Trans ist eine Liste von möglichen Werten spezifiziert worden. Weitere Werte müssen projektspezifisch festgelegt werden.	xs:anyURI	1	"Zustelladresse" "Hauptwohnsitz" "Nebenwohnsitz" "Standort" "Rechnungs- adresse" "weitere"
PostalAddress	Die eigentliche Adresse	pe: Postal- AddressType	1	

4.2.7. PostalAddressType

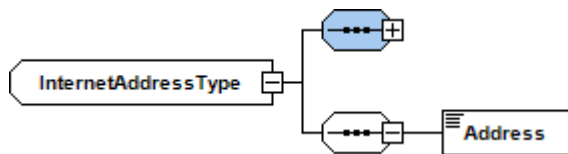


pe: PostalAddressType	Description	Type	Occur	Restriction
CountryName	Ländername nach ISO-3166 oder interner Poststandard.	xs:string	0-1	
PostalCode	Postleitzahl	xs:string	1	
State	Bundesland	xs:string	0-1	

pe: PostalAddressType		Description	Type	Occur	Restriction
Hamlet		Ort	xs: string	1	
DeliveryAddress				1	
	StreetName	Wenn eine Adresse keinen Straßennamen hat, sollte in diesem Feld nochmals der Name der Ortschaft übergeben werden. Null nur bei Verwendung von PostOfficeBox.	xs: string	0-1	
	BuildingNumber	Hausnummer (Ordnungsnummer laut Register)	xs: string	0-1	
	Unit	Stiege	xs: string	0-1	
	DoorNumber	Tür, Top	xs: string	0-1	
	PostOfficeBox	Postfach	xs: string	0-1	
	AddressRegister-Entry	Verweis auf Adressregister-Eintrag	xs: string	0-1	
	AddressCode		xs: string	1	
	SubCode		xs: string	0-1	max. 3 Zeichen
	Object-Number		xs: string	0-1	max. 7 Zeichen
Recipient				0-1	

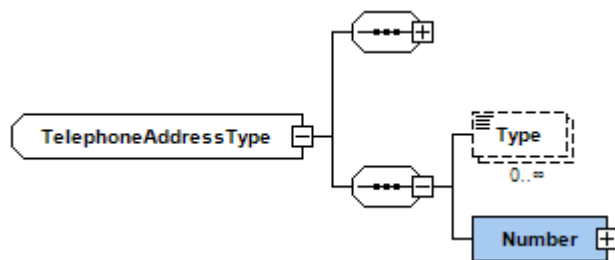
pe: PostalAddressType	Description	Type	Occur	Restriction
AdditionalText	Aufgrund "unbounded" ist hier ein mehrzeiliger Zusatztext für "zu Händen" möglich; soll im Adressblock unter dem Empfängernamen und über der Adresse angedruckt werden.	xs:string	1-n	

4.2.8. InternetAddressType



pe: Internet- AddressType	Description	Type	Occur
Address	In ELAK-Trans hauptsächlich die E-Mail-Adresse, aber auch Web-, FTP-, LDAP-Adressen, etc. denkbar.	xs:string	1

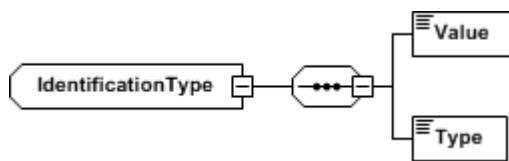
4.2.9. TelephoneAddressType



pe:Telephone- AddressType	Description	Type	Occur	Restriction
Type	Für ELAK-Trans gibt es eine vordefinierte Liste. Bei Bedarf müssen projektspezifisch zusätzliche Werte festgelegt werden.	xs:anyURI	1	"Fax" "Mobil" "Privat" "Geschaeftlich"

pe:Telephone-AddressType	Description	Type	Occur	Restriction
Number	Telefonnummer	pe:Telcom-Number-Type	1	
FormattedNumber	Vollständige Nummer, z.B. +43 1 5131345 4664	xs:string	1	

4.2.10. IdentificationType



pe:IdentificationType	Description	Type	Occur
Value	Wert der ID, z.B. eine bestimmte GUID.	xs:string	1
Type	Information, um welche Art der ID es sich handelt, etwa eine GUID oder BPK. Welcher Typ von IDs für welches Objekt innerhalb der Personen-Struktur verwendet wird, ist weiter unten noch gesondert beschrieben.	xs:anyURI	1

4.2.11. DateOfBirthType

Union aus xs:date xs:gYearMonth xs:gYear

4.2.12. DateOfDeathType

Union aus xs:date xs:gYearMonth xs:gYear

4.2.13. Wo im PersonData legt das FIS die ID's ab?

- PersonData wurde soweit eingeeengt, dass IDs nur noch an jenen Stellen abgelegt werden können, an denen sie für ELAK-Trans fachlich Sinn machen.
- Die FIS-ID wird jeweils mit dem Type "ID" abgelegt.

- Identification-Knoten unterhalb von PhysicalPersonType bzw. CorporateBodyType

```

<ns2:MetaData>
- <ns2:Participants>
  - <ns2:Participant>
    - <ns3:Person xsi:type="ns3:PhysicalPersonType" xmlns:xs
      - <ns3:Identification>
        <ns3:Value>FIS-ID-Person-1</ns3:Value>
        <ns3:Type>ID</ns3:Type>
      </ns3:Identification>
      - <ns3:Identification>
        <ns3:Value>123456</ns3:Value>
        <ns3:Type>Firmenbuchnummer</ns3:Type>
      </ns3:Identification>
      - <ns3:Name>
        <ns3:GivenName>Christoph</ns3:GivenName>
        <ns3:FamilyName>Straßer</ns3:FamilyName>
      </ns3:Name>
    </ns3:Person>
    + <ns3:Address xsi:type="ns3:TypedPostalAddressType" xm
    + <ns3:Address xsi:type="ns3:InternetAddressType" xmlns:
    + <ns3:AdditionalData>
  </ns2:Participant>
</ns2:Participants>
- <ns2:Participants>
  - <ns2:Participant>
    - <ns3:Person xsi:type="ns3:CorporateBodyType" xmlns:xs
      - <ns3:Identification>
        <ns3:Value>FIS-ID-Person-2</ns3:Value>
        <ns3:Type>ID</ns3:Type>
      </ns3:Identification>
        <ns3:FullName>Land Oberösterreich</ns3:FullName>
      </ns3:Person>
    </ns2:Participant>
  </ns2:Participants>
</ns2:MetaData>

```

- Identification-Knoten unterhalb von TypedPostalAddressType

```

<ns3:Address xsi:type="ns3:TypedPostalAddressType" xm
- <ns3:Identification>
  <ns3:Value>FIS-ID-Adresse-1</ns3:Value>
  <ns3:Type>ID</ns3:Type>
</ns3:Identification>
- <ns3:Identification>
  <ns3:Value>478/22</ns3:Value>
  <ns3:Type>GrundstueckNr</ns3:Type>
</ns3:Identification>
  <ns3:Type>Hauptwohnsitz</ns3:Type>
- <ns3:PostalAddress>
  <ns3:PostalCode>4020</ns3:PostalCode>
  <ns3:Hamlet>Linz</ns3:Hamlet>
</ns3:PostalAddress>
</ns3:Address>
<ns3:Address xsi:type="ns3:InternetAddressType" xmlns:x
  <ns3:Address>max.muster@ooe.gv.at</ns3:Address>
</ns3:Address>

```

ACHTUNG: Die XMLs in den beiden obigen Beispielen sind vereinfacht und auf die Identifier-Thematik limitiert. Es handelt sich dabei um keine kompletten, Schema-validen PersonData-Datenstrukturen!

4.2.14. Welche Adresse der Person wird für die Zustellung verwendet?

- Es wurde nur eine TypedPostalAddress übergeben. => Diese eine TypedPostalAddress.
- Es wurden mehrere TypedPostalAddresses übergeben, eine davon wurde als "AdresseDesSchreibens" gekennzeichnet. => Die so gekennzeichnete TypedPostalAddress.

```
<ns3:Address xsi:type="ns3:TypedPostalAddressType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-ns
- <ns3:Identification>
  <ns3:Value>FIS-ID-Adresse-1</ns3:Value>
  <ns3:Type>ID</ns3:Type>
</ns3:Identification>
- <ns3:Identification>
  <ns3:Value>478/22</ns3:Value>
  <ns3:Type>GrundstueckNr</ns3:Type>
</ns3:Identification>
- <ns3:Identification>
  <ns3:Value>true</ns3:Value>
  <ns3:Type>AdresseDesSchreibens</ns3:Type>
</ns3:Identification>
<ns3:Type>Hauptwohnsitz</ns3:Type>
- <ns3:PostalAddress>
  <ns3:PostalCode>4020</ns3:PostalCode>
  <ns3:Hamlet>Linz</ns3:Hamlet>
</ns3:PostalAddress>
</ns3:Address>
```

- Es wurden mehrere TypedPostalAddresses übergeben, keine davon ist als "AdresseDesSchreibens" gekennzeichnet. => Die erste TypedPostalAddress.

4.3. ELAK-Trans-spezifische Datenstrukturen

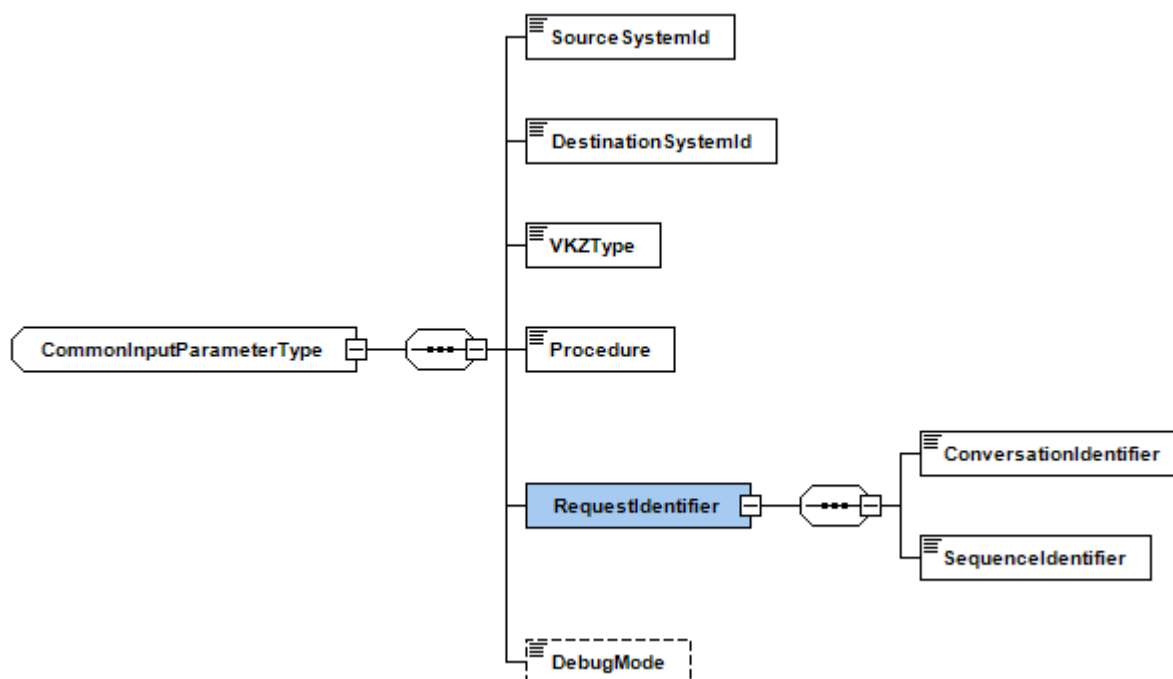
Alle ELAK-Trans-spezifischen Datenstrukturen verwenden den Namespace
 xmlns = "http://reference.e-government.gv.at/namespace/elaktrans/3#".

Folgende weitere Namespaces werden verwendet:

- xmlns:xs = "http://www.w3.org/2001/XMLSchema"
- xmlns:edi = "http://reference.e-government.gv.at/namespace/edidoc/20130808#"

4.3.1. CommonInputParameterType

Allgemeiner Schnittstellenparameter für ELAK-Trans-Requests. Damit werden die ursprünglichen ELAK-Trans-2.0-Typen "commonApplicationInputParameter" und "commonELAK-InputParameter" fusioniert.

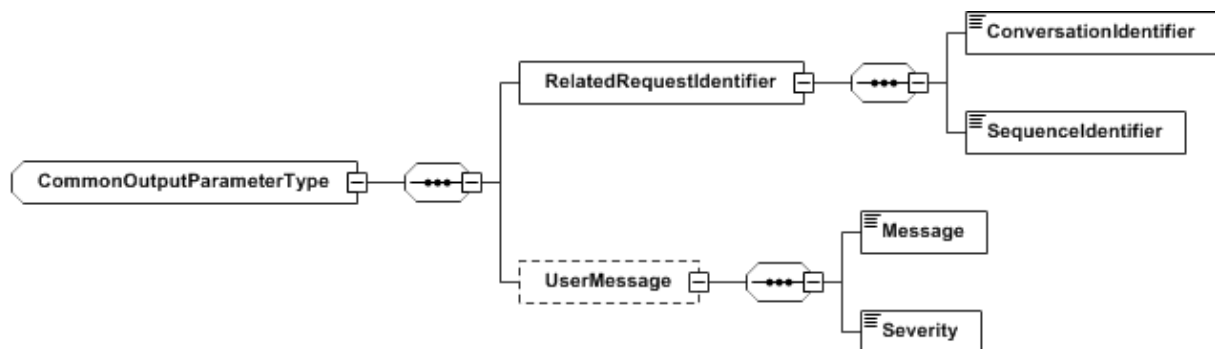


CommonInput-ParameterType	Description	Type	Occur
SourceSystemId	Eindeutige Kennung des Quellsystems (ELAK oder FIS, z.B. "at.gv.noe.elak" bzw. "at.gv.ooe.sotra")	xs:string	1
Destination-SystemId	Eindeutige Kennung des Zielsystems (ELAK oder FIS, z.B. "at.gv.noe.elak" bzw. "at.gv.ooe.sotra")	xs:string	1
VKZType	Verwaltungskennzeichen, welches die Behörde des Empfängers definiert (z.B. "L4AL-IT") (siehe http://reference.e-government.gv.at/KONVENTIONEN-WEITERE-KONZEPT.506.0.html Abschnitt Verwaltungskennzeichen)	xs:string	1
Procedure	Leistungskürzel des Verfahrens laut Verfahrenstabelle (siehe http://reference.e-government.gv.at/KONVENTIONEN-WEITERE-KONZEPT.506.0.html Abschnitt Elektronischer Leistungskatalog der Österreichischen Verwaltung) das Leistungskürzel kann z.B. zur Feststellung des Sachgebiets in der ELAK-Trans-Middleware verwendet werden.	xs:string	1

CommonInput-ParameterType	Description	Type	Occur
RequestIdentifier	Identifizier für a) systemübergreifende Fehlersuche und Logging und b) Ablaufsteuerung bei asynchroner multi-threaded Massenverarbeitung und Zuordnung von Requests und Responses.	Request-IdentifizierType	1
DebugMode	Damit kann man angeben, in welchem Detailgrad der Server Fehlermeldungen, Warnungen und Informationen zurückliefern soll. Wenn der Wert "true" ist, also wenn man etwa gerade einen Client im Testsystem entwickelt, dann will man sinnvollerweise alle am Server verfügbaren Informationen, um das Verhalten desselben besser nachvollziehen zu können. Ist das Element nicht spezifiziert, so ist dies gleichbedeutend, als wäre der Wert "false".	xsd:boolean	0-1

4.3.2. CommonOutputParameterType

Damit werden die ursprünglichen ELAK-Trans-2.0-Typen "commonApplicationOutputParameter" und "commonElakOutputParameter" fusioniert.



CommonOutput-ParameterType	Description	Type	Occur
RequestIdentifier	Identifizier für a) systemübergreifende Fehlersuche und Logging und b) Ablaufsteuerung bei asynchroner multi-threaded Massenverarbeitung und Zuordnung von Requests und Responses.	Request-Identifizier-Type	1

CommonOutput-ParameterType	Description	Type	Occur
UserMessage		User-Message-Type	0-1

4.3.3. RequestIdentifierType

Der RequestIdentifier soll bei der system- und organisationsübergreifenden Fehlersuche helfen. Außerdem ermöglicht er eine Ablaufsteuerung bei asynchroner Multi-Thread-Massenverarbeitung, dabei insbesondere die Zuordnung von Responses zu den entsprechenden Requests.

RequestIdentifierType	Description	Type	Occur
ConversationIdentifier	Die Conversation-ID muss innerhalb der SourceSystem-ID eindeutig sein.	xs:token	1
SequenceIdentifier	Die Sequence-ID muss innerhalb der Conversation-ID eindeutig und vor allem aufsteigend vergeben werden, um eine definierte Reihenfolge der "Requests" bei der Abarbeitung sicherstellen zu können.	xs:token	1

4.3.4. UserMessageType

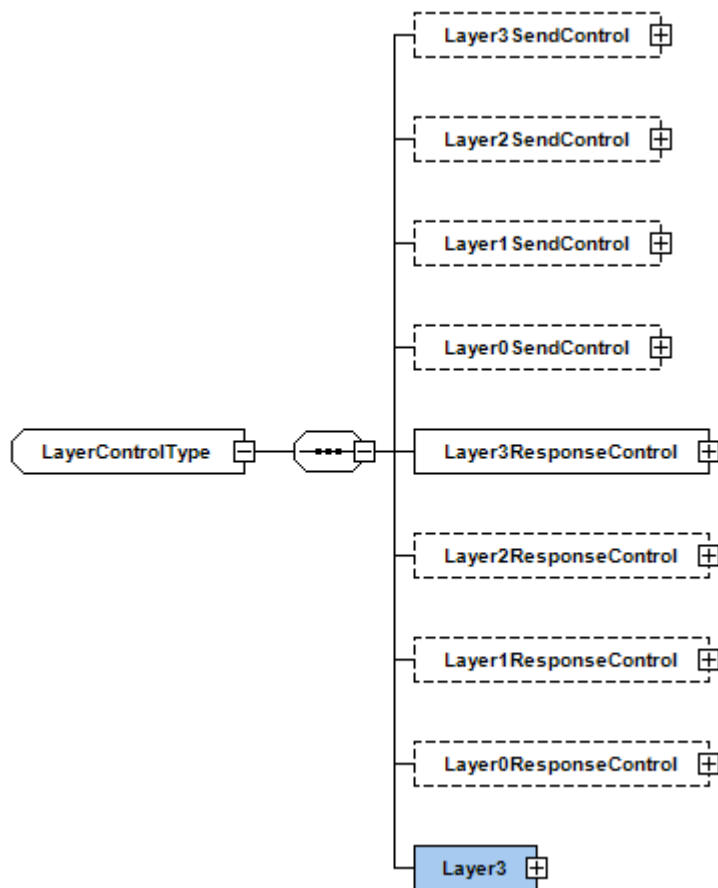
Bietet die Möglichkeit, dem Benutzer in einer für ihn lesbaren Art und Weise Rückmeldung darüber zu geben, was auf der Server-Seite passiert ist. Sämtliche Nachrichten - falls mehrere erwünscht sind - werden in einer Zeichenkette formatiert.

User-Message-Type	Description	Type	Occur	Restriction
Message	Sämtliche Nachrichten werden hier in einer Zeichenkette formatiert. Die aufrufende Anwendung kann dann - falls gewünscht - diese retournierte Zeichenkette einfach im User-Interface anzeigen.	xs:string	1	

User-Message-Type	Description	Type	Occur	Restriction
Severity	<p>Für normale Benutzerinformationen, also etwa für Rückmeldungen, dass etwas gespeichert wurde, wählt man naheliegenderweise den Wert INFORMATION. Sobald sich in den bis zu n zu einer Zeichenkette formatierten Rückmeldungen eine Warnung befindet, soll der Wert WARNING verwendet werden. Sprich, der höchste Schweregrad entscheidet. Normale und fatale Fehler werden über Exceptions bzw. Faults abgehandelt.</p> <p>Beispiel für "INFORMATION": "Es wurde ein Akt mit dem Kennzeichen ... erzeugt."</p> <p>Beispiel für "WARNING": "Der Geschäftsfall ... war bereits abgeschlossen und musste wieder aktiviert werden."</p>	Severity-Type	1	"INFORMATION" "WARNING"

4.3.5. LayerControlType

Mit den Informationen in dieser Struktur kann man einerseits für die SendData-Operation die semantische Bedeutung der Daten innerhalb des EDIAKT-/EDIDOC-Elements im Request konkretisieren. Man kann etwa festlegen, dass im Layer 2 nur die vom Client aktualisierten Adressaten mitgeliefert werden, alle anderen Daten aber unverändert blieben und daher gar nicht mitgeliefert werden. Andererseits kann man für sämtliche ELAK-TRANS-Operationen die Responses genau festlegen; d.h. welche Objekte sollen in der EDIAKT-/EDIDOC-Struktur der Response überhaupt enthalten sein (Selektion), und welche Daten bzw. Details der über die Selektion festgelegten Objekte sollen zurückgeliefert werden (Projektion). Beim UnbindData werden die (FIS-)IDs der Objekte angegeben, für welche die Verknüpfung aufgehoben werden soll.

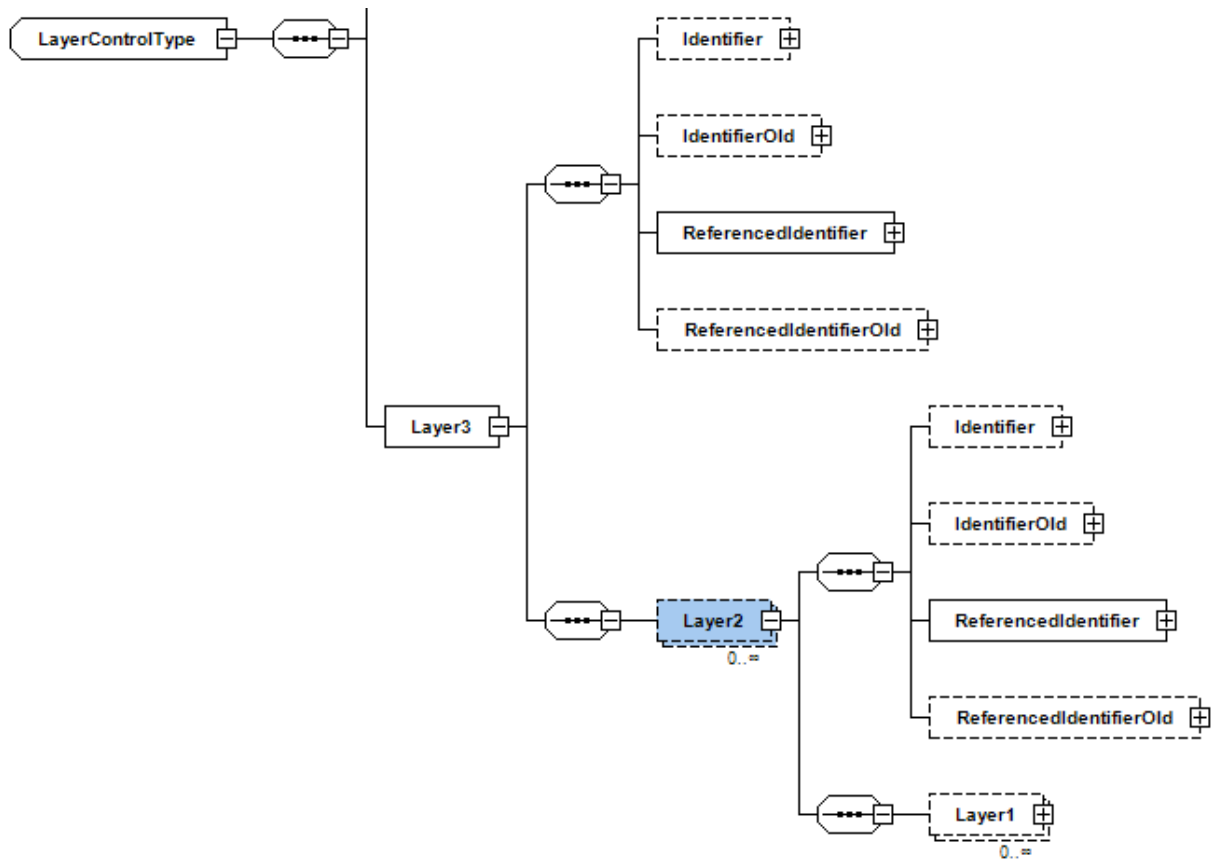


LayerControl-Type	Description	Type	Occur
Layer3Send-Control	Dieser Knoten ist nur bei SendData relevant und steuert in diesem Fall, wie die im Layer 3 (Ebene "Akt") der EDIAKT-/EDIDOC-Struktur im Request mitgelieferten Daten zu interpretieren sind. Zunächst kann die Festlegung sehr granular getroffen werden (BasicData, Participants, State, etc.), und für jeden dieser Bereiche kann explizit festgelegt werden, ob im Request veränderte Daten mitgeliefert werden. Faktisch lässt sich mit einer sinnvollen Parametrisierung Laufzeit am Server einsparen, da sich damit implizit schreibende und damit (vor allem im ELAK) "teure" Zugriffe vermeiden lassen.	LayerSend-ControlType	0-1
Layer2Send-Control	Dieser Knoten ist nur bei SendData relevant und steuert in diesem Fall, wie die im Layer 2 (Ebene "Geschäftsfall") der EDIAKT-/EDIDOC-Struktur im Request mitgelieferten Daten zu interpretieren sind. Die semantische Bedeutung ist gleich wie jene von "Layer3SendControl".	LayerSend-ControlType	0-1

LayerControl-Type	Description	Type	Occur
Layer1Send-Control	Dieser Knoten ist nur bei SendData relevant und steuert in diesem Fall, wie die im Layer 1 (Ebene "Geschäftsstück") der EDIAKT-/EDIDOC-Struktur im Request mitgelieferten Daten zu interpretieren sind. Die semantische Bedeutung ist gleich wie jene von "Layer3SendControl".	LayerSend-ControlType	0-1
Layer0Send-Control	Dieser Knoten ist nur bei SendData relevant und steuert in diesem Fall, wie die im Layer 0 (Ebene "Dokument") der EDIAKT-/EDIDOC-Struktur im Request mitgelieferten Daten zu interpretieren sind. Die semantische Bedeutung ist gleich wie jene von "Layer3SendControl". Wesentlich - im Gegensatz zu den anderen Layern - ist hier die Festlegung für den Bereich "BinaryContent", denn dort ist wohl die größte Performance-Optimierung möglich. Das kann etwa dann interessant sein, wenn nur die Metadaten eines Dokuments aktualisiert werden sollen.	LayerSend-ControlType	0-1
Layer3-Response-Control	Für jede ELAK-Trans-Operation ist eine mehr oder weniger mit Daten befüllte Response vorgesehen. Welche Informationen in der Response mitgeliefert werden sollen, kann man an dieser Stelle für den Layer 3 (Ebene "Akt") festlegen, genauer gesagt, für sämtliche in die Response zu inkludierende Objekte des Layers 3. Zunächst kann über das Layer3-Element festgelegt werden, welche Objekte des Layers 3 überhaupt in der Response enthalten sein sollen (Selektion). Beim Layer 3 gibt es nur diese eine explizite Möglichkeit der Selektion; das kann zwar nicht direkt durch Einschränkungen im Schema erzwungen werden, aber die Spezifikation sieht das vor, und in weiterer Folge werden das die Implementierungen sicherstellen. Für die selektierten Objekte des Layers 3 kann festgelegt werden, welche Bereiche bzw. Details (BasicData, Participants, State, etc.) in der Response berücksichtigt werden sollen (Projektion).	Layer-Response-ControlType	1

LayerControl-Type	Description	Type	Occur
Layer2-Response-Control	Dieser Knoten legt für den Layer 2 (Ebene "Geschäftsfall") fest, welche Informationen dieser Ebene in der Response mitgeliefert werden sollen. Zunächst kann einerseits implizit über genau spezifizierte und adressierte Layer2-Elemente innerhalb des Layer3-Elements festgelegt werden, welche Objekte des Layers 2 in der Response enthalten sein sollen (Variante 1 der Selektion). Alternativ kann über die Unterelemente "IncludeUnspecified-CoupledObjects" bzw. "IncludeUncoupled-Objects" eine Selektion getroffen werden. Die Bedeutung der beiden Elemente wird ebendort speziell erläutert. Über die Selektion hinaus kann wie beim Layer 3 festgelegt werden, welche Informationen der selektierten Objekte berücksichtigt werden sollen (Projektion).	Layer-Response-ControlType	0-1
Layer1-Response-Control	Dieser Knoten legt für den Layer 1 (Ebene "Geschäftsstück") fest, welche Informationen dieser Ebene in der Response mitgeliefert werden sollen. Die syntaktische Festlegung erfolgt wie im Layer 2 und die Semantik entspricht jener des Layers 2.	Layer-Response-ControlType	0-1
Layer0-Response-Control	Dieser Knoten legt für den Layer 0 (Ebene "Dokument") fest, welche Informationen dieser Ebene in der Response mitgeliefert werden sollen. Die syntaktische Festlegung erfolgt wie im Layer 2 und die Semantik entspricht ebenso jener des Layer 2. Aber gerade bei diesem Layer ist diese Kontrollstruktur so essentiell, geht es doch darum, an dieser Stelle zu entscheiden ob etwa die Inhalte der Layer-0-Objekte in der Response enthalten sein sollen und/oder nur die URLs. Das macht im Performance-Verhalten einen riesengroßen Unterschied.	Layer-Response-ControlType	0-1

LayerControl-Type	Description	Type	Occur
Layer3	<p>Hier wird die explizite Selektion des einzigen Layer3-Objekts vorgenommen; für den Layer 3 ist diese implizite Variante nämlich für jeden Anwendungszweck gleichzeitig die einzige. Bei SendData ist die Adressierung prinzipiell über die (parallele und ohnehin vorhandene) EDIAKT-/EDIDOC-Struktur ausreichend, sodass im Standardfall für die Konfiguration des Requests hier nichts getan werden muss. Es gibt aber Sonderfälle, wie zum Beispiel das Umprotokollieren, in denen es notwendig ist, alte und neue IDs mitzugeben; dann ist diese Struktur natürlich relevant. Additiv ist auch bei der SendData-Operation die Response zu konfigurieren, sodass zumindest der Layer 3 aus dieser Perspektive heraus angegeben werden muss. Bei ReadData ist - ebenfalls aufgrund der Response-Konfiguration - zumindest die Angabe eines Layer3-Objekts erforderlich. Die Objekte der hierarchisch darunterliegenden Objekte können entweder ebenfalls explizit mithilfe von Layer2-Knoten spezifiziert werden oder alternativ implizit bzw. generisch selektiert werden. Bei UnbindData ist die Struktur auch zwingend anzugeben, denn da gibt es keine alternative EDIAKT-/EDIDOC-Struktur.</p>	Layer3-Control-Type	1



Die konkreten Kontrollstrukturen für die Layer 3, 2, 1 und 0 basieren auf dem Typ LayerXControlType, der gemeinsame Elemente, im Wesentlichen die Identifiers, definiert.

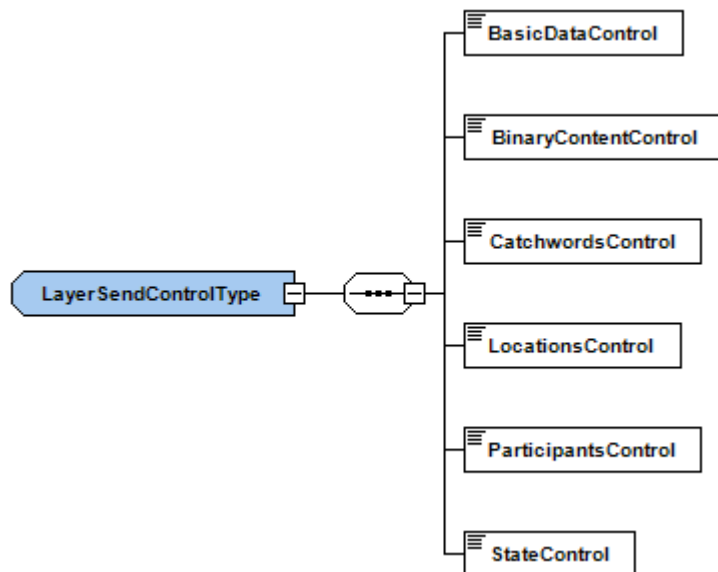
Layer3ControlType (Layer2ControlType) (Layer1ControlType) (Layer0ControlType)	Description	Type	Occur
Identifizier	Falls erforderlich, wird hier ein layer-spezifischer ELAK-Schlüssel; das ist im Normalfall ein fachlicher ELAK-Schlüssel, wie z.B. das Geschäftszeichen, Kennzeichen, oder wie immer das begrifflich heißt.	edi:IdentificationType	0-1
IdentifizierOld	Falls erforderlich, wird hier ein vorheriger ELAK-Schlüssel eingetragen; das ist im Normalfall ein fachlicher ELAK-Schlüssel, wie z.B. das Geschäftszeichen, Kennzeichen, oder wie immer das begrifflich heißt.	edi:IdentificationType	0-1

Layer3ControlType (Layer2ControlType) (Layer1ControlType) (Layer0ControlType)	Description	Type	Occur
ReferencedIdentifier	Bei SendData, ReadData und UnbindData wird hier die FIS-ID eingetragen.	edi:IdentificationType	1
ReferencedIdentifierOld	Beim Umprotokollieren mittels SendData wird hier die vorherige FIS-IDs eingetragen. In allen anderen Fällen ist dieser Knoten nicht sinnvoll und somit nicht wirksam.	edi:IdentificationType	0-1
Layer2 (Layer1) (Layer0) -		Layer2ControlType (Layer1ControlType) (Layer0ControlType) -	0-n

Ebenen 2 - 0 sind demnach nach dem gleichen Muster wie die Ebene 3 aufgebaut.

4.3.6. LayerSendControlType

Mit dieser Struktur teilt der Client dem Server bei der SendData-Operation feingranular mit, welche Daten der Request enthält. Dafür wird die Menge aller Daten eines Layers logisch in folgende Teilmengen zerlegt: die Basisdaten, die Begriffe, die Ortsbezüge, die Adressaten, die Status, die binären Inhalte und die URL(s). Mithilfe dieser Angaben kann und muss der Server entscheiden, welche verknüpften Objekte aktualisiert werden. Gerade im hochgradig verteilten System - und bei ELAK-TRANS haben wir ein solches - ist diese Möglichkeit der expliziten Festlegung relevant, da mit diesem Mechanismus einerseits die Requests so klein wie möglich gehalten werden können und andererseits die schreibenden Operationen inklusive aller begleitenden Maßnahmen (Locks, etc.) am Server auf ein Minimum reduziert werden können. Will ein Client zum Beispiel nur die Daten der Layer 1 und 0 aktualisieren, so hat er auf diese Art und Weise die Möglichkeit, die Objekte der Layer 3 und 2 vor Updates zu bewahren. Oder - anderes Beispiel - möchte der Client etwa nur den Status eines Geschäftsfalls ändern, dann muss er nicht zusätzlich die Basisdaten, die Adressaten, die Ortsbezüge und die Begriffe mitliefern.

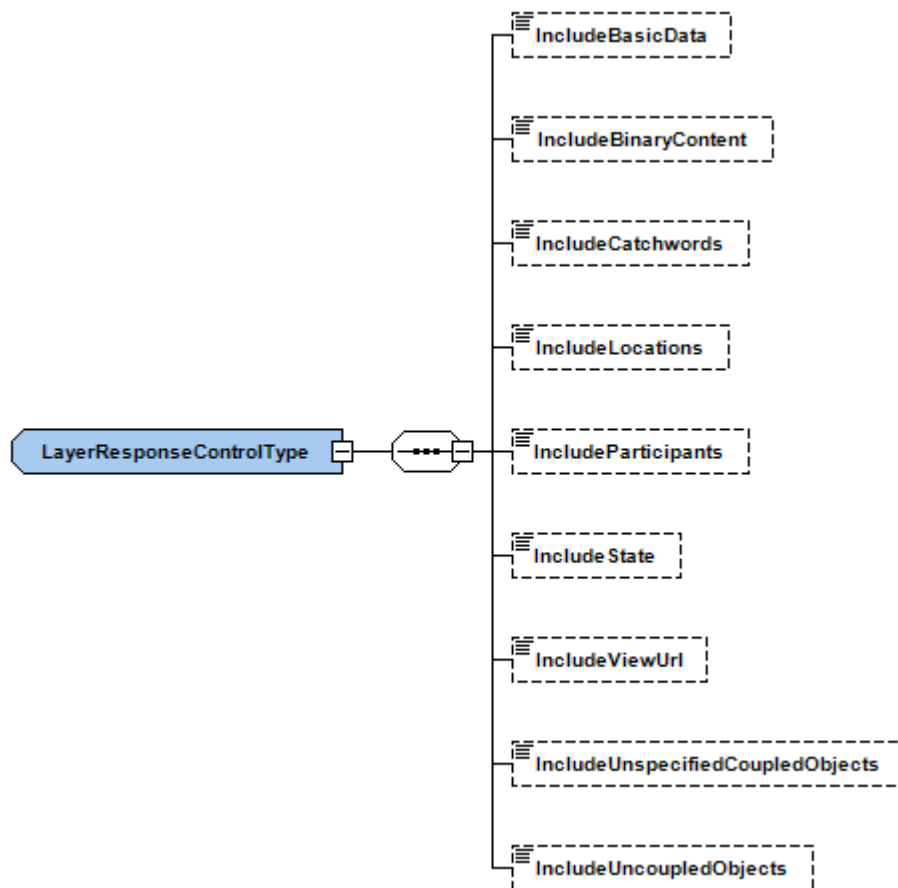


LayerSend- ControlType	Description	Type	Occur
ContainsModified BasicData	Gibt an, ob Objekte des gegenständlichen Layers im Request veränderte Basisdaten enthalten. Betrifft alle Daten, die nicht durch eines der anderen Attribute gesteuert werden, aus dem Layer (z.B. Subject) und aus den MetaData (z.B. Annotation).	xsd:boolean	1
ContainsModified BinaryContent	Gibt an, ob Objekte des gegenständlichen Layers im Request veränderte binäre Inhalte enthalten. Diese Information ist nur für den Layer 0 relevant, bei allen anderen kann man diese Information in der Service-Implementierung ignorieren. Deshalb ist die Kardinalität des Elements - im Gegensatz zu allen anderen Contains...-Elementen hier mit 0..1 und einem Standardwert "false" ausgeprägt. Der Sinn des Elements an sich ist im Wesentlichen der, dass man so auch den Namen eines Dokuments im ELAK ändern kann, ohne gleich erraten zu müssen, ob auch der Inhalt verändert wurde. Diese Konfigurationsmöglichkeit betrifft das Documents-Element von SendDataInput, SendDataOutput und ReadDataOutput.	xsd:boolean	0-1, default: false
ContainsModified Catchwords	Gibt an, ob Objekte des gegenständlichen Layers im Request veränderte Begriffe enthalten. Betrifft die MetaData-Catchwords.	xsd:boolean	1

LayerSend- ControlType	Description	Type	Occur
ContainsModified Locations	Gibt an, ob Objekte des gegenständlichen Layers im Request veränderte Ortsbezüge enthalten. Betrifft die MetaData-Locations.	xsd:boolean	1
ContainsModified Participants	Gibt an, ob Objekte des gegenständlichen Layers im Request veränderte Adressaten enthalten. Betrifft die MetaData-Participants.	xsd:boolean	1
ContainsModified State	Gibt an, ob Objekte des gegenständlichen Layers im Request veränderte Status enthalten. Betrifft MetaData-Closed, MetaData-Canceled und MetaData-Approved.	xsd:boolean	1

4.3.7. LayerResponseControlType

Mit dieser Struktur hat der Client die Möglichkeit, bei sämtlichen Operationen feingranular festzulegen, welche Daten des gegenständlichen Layers der Server in die Response aufnehmen soll. Dafür wird die Menge aller Daten eines Layers logisch in folgende Teilmengen zerlegt: die Basisdaten, die Begriffe, die Ortsbezüge, die Adressaten, die Status, die binären Inhalte und die URL(s). Der Server kann theoretisch mehr Daten als vom Client verlangt in die Response aufnehmen, aber er muss jedenfalls die verlangten Daten inkludieren. Gerade im hochgradig verteilten System - und bei ELAK-TRANS haben wir ein solches - ist diese Möglichkeit der expliziten Festlegung relevant, da mit diesem Mechanismus einerseits die Responses so klein wie möglich gehalten und andererseits die Leseoperationen am Server auf ein Minimum reduziert werden können.



LayerResponse-ControlType	Description	Type	Occur
IncludeBasicData	<p>Der Server soll die Basisdaten des Layers in die Response aufnehmen. Zu den Basisdaten gehören der Betreff, die Anmerkung, das Erstellungsdatum, der Objekttyp, der Bearbeiter, das Eingangsdatum und das Abfertigungs-bzw. Versanddatum; beim Layer 1 kommt der Geschäftsstück-Typ ("BusinessType") dazu. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false".</p> <p>Betrifft alle Daten, die nicht durch eines der anderen Attribute gesteuert werden, aus dem Layer (z.B. Subject) und aus den MetaData (z.B. Annotation).</p>	xs:boolean	0-1

LayerResponse- ControlType	Description	Type	Occur
IncludeBinaryContent	Der Server soll die Inhalte, sofern es im Layer welche gibt, in die Response aufnehmen. Das heißt, für die Layer 1 bis 3 ist diese Information irrelevant, für Layer 0 essentiell. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false". Betrifft das Documents-Element von SendDataInput, SendDataOutput und ReadDataOutput.	xs:boolean	0-1
IncludeCatchwords	Der Server soll die Begriffe des gegenständlichen Layers in die Response aufnehmen. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false". Betrifft die MetaData-Catchwords.	xs:boolean	0-1
IncludeLocations	Der Server soll die Ortsbezüge des gegenständlichen Layers in die Response aufnehmen. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false". Betrifft die MetaData-Locations.	xs:boolean	0-1
IncludeParticipants	Der Server soll die Adressaten des gegenständlichen Layers in die Response aufnehmen. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false". Betrifft die MetaData-Participants	xs:boolean	0-1
IncludeState	Der Server soll die Statusinformationen des gegenständlichen Layers in die Response aufnehmen. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false". Betrifft MetaData-Closed, MetaData-Canceled und MetaData-Approved.	xs:boolean	0-1

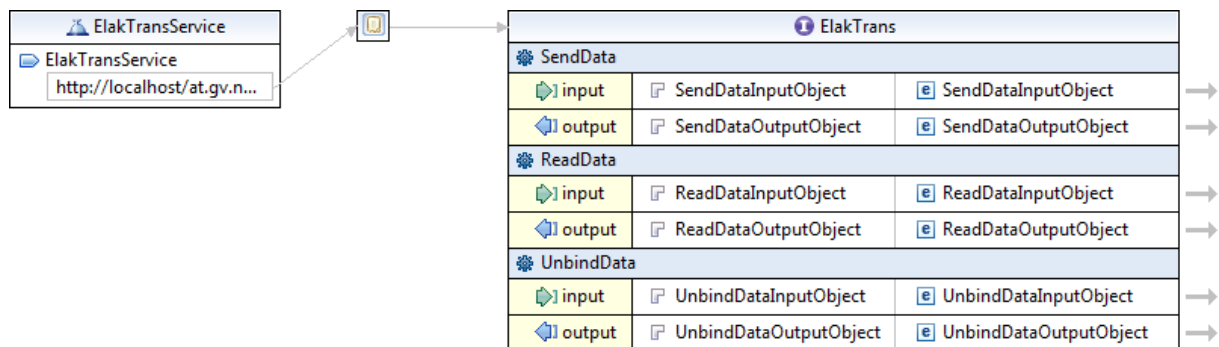
LayerResponse- ControlType	Description	Type	Occur
IncludeViewUrl	<p>Der Server soll die URL, z.B. eine solche zum Bearbeiten des Objekts im User-Interface, des gegenständlichen Layers in die Response aufnehmen. Ist dieses Element nicht angegeben, so hat das die gleiche Bedeutung als wäre der Wert "false".</p> <p>Betrifft die MetaData-Identifizier-TechnicalID bzw. MetaData-ReferencedIdentifizier-TechnicalID.</p>	xs:boolean	0-1
IncludeUnspecified- CoupledObjects	<p>Gibt an, dass auch solche in der ELAK-TRANS-Object-Mappingtabelle gemappten Objekte in der Response berücksichtigt werden sollen, die weder in der EDIDOC- noch in der LayerControl-Struktur explizit erwähnt wurden. Wird das Element nicht angegeben, so ist dies mit dem Wert "false" gleichbedeutend. Das spezifizierte Verhalten gilt für Objekte des gegenständlichen Layers, nicht für darunterliegende.</p> <p>Nur für Layer0, Layer1 und Layer2. Nicht für Layer3.</p>	xs:boolean	0-1
IncludeUncoupled- Objects	<p>Gibt an, ob auch solche Objekte im ELAK berücksichtigt werden sollen, die im ELAK erzeugt wurden und nicht in der ELAK-TRANS-Object-Mappingtabelle gemappt sind. Wird das Element nicht angegeben, so ist dies mit dem Wert "false" gleichbedeutend. Das spezifizierte Verhalten gilt für Objekte des gegenständlichen Layers, nicht für darunterliegende.</p> <p>Nur für Layer0, Layer1 und Layer2. Nicht für Layer3.</p>	xs:boolean	0-1

5. WS Methoden

5.1. Einleitung

In ELAK-Trans 3.0 werden die Webservice Methoden SendData, ReadData und Unbind definiert.

Abbildung 4 – Webservice-Methoden



Es ist jedoch zu beachten, dass es 2 Kommunikationswege und somit 2 Client-Server Implementierungen gibt:

- FIS->ELAK (FIS-2-ELAK) Kommunikation: Dieser Kommunikationsweg ist der Standardfall und deckt alle Aktionen ab, die aus einem FIS heraus angesteuert werden. Auf der Seite des FIS wird ein Webservice Client (FIS-2-ELAK-Client) implementiert, der die entsprechenden Methoden des Webservice (FIS-2-ELAK-Server) auf der ELAK Seite aufruft.
- ELAK->FIS (ELAK-2-FIS) Kommunikation: Dieser Kommunikationsweg wird für die bidirektionale und möglichst nahtlose Integration des ELAK mit dem FIS benötigt. Er deckt vor allem die Übermittlung von im ELAK angelegten Objekten an das FIS ab. Auf der Seite des ELAK wird ein Webservice Client (ELAK-2-FIS-Client) implementiert, der die entsprechenden Methoden im Webservice (ELAK-2-FIS-Server) auf der FIS Seite aufruft.

5.2. FIS -> ELAK Kommunikation (FIS-2-ELAK)

5.2.1. SendData

Die SendData Methode dient dazu, ausgehend vom FIS, folgende UseCases abzudecken:

Aus Sicht ELAK-Trans FIS-2-ELAK Client:

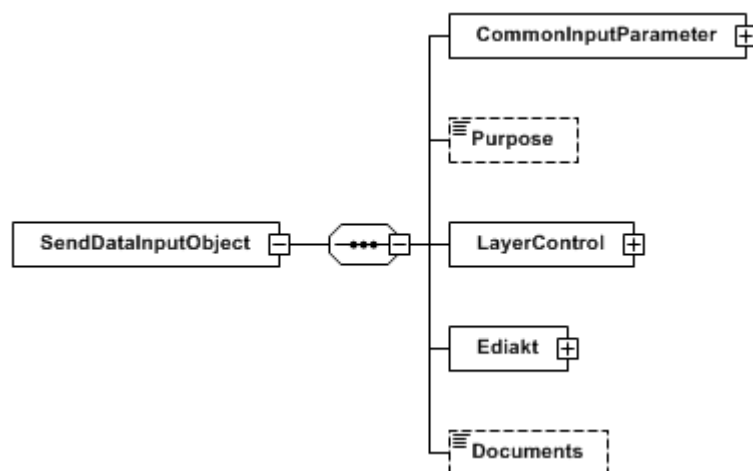
- Aktenstruktur (Geschäftsobjekte) erzeugen
- Aktenstruktur (Geschäftsobjekte) aktualisieren
- Geschäftsstück verschieben

Aus Sicht ELAK-Trans FIS-2-ELAK Server:

- Akt, Geschäftsfall, Geschäftsstück und/oder Dokument erzeugen
- Akt, Geschäftsfall, Geschäftsstück und/oder Dokument aktualisieren
 - inkludiert die Status-Aktualisierung "Akt abschließen"
- Geschäftsstück umprotokollieren

Die Detailbeschreibungen können dem Abschnitt "6 Anwendungsfälle" entnommen werden.

5.2.1.1. *SendDataInputObject*

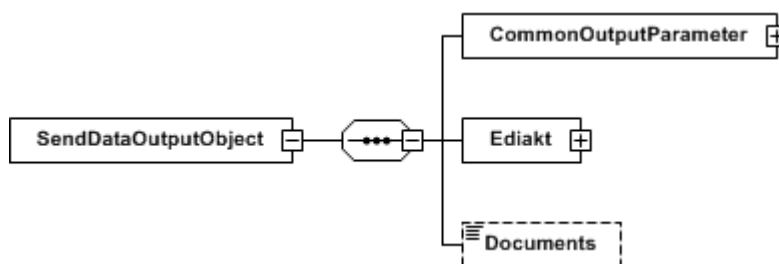


SendData-InputObject	Description	Type	Occur	Restriction
Common-Input-Parameter	Allgemeiner Schnittstellen-Parameter für sämtliche Requests. Damit werden die ursprünglichen ELAK-Trans-2.0-Typen "commonApplicationInputParameter" und "commonELAKInputParameter" fusioniert.	Common-Input-Parameter-Type	1	

SendData-InputObject	Description	Type	Occur	Restriction
Purpose	<p>Dieser Vermerk bezieht sich immer auf den Layer 1 - das Geschäftsstück. Den Wert kann die ELAK-TRANS-Middleware etwa dafür verwenden, um abhängig davon im ELAK verschiedene Objektklassen für das Anlegen des Geschäftsstücks zu verwenden (z.B. Eingangs- oder Ausgangsstück). Auch andere Steuerungen in der Logik lassen sich an dieser Auswahl, eventuell auch in Verbindung mit anderen Elementen, festmachen.</p> <p>INPUT: Fabasoft-Kunden wählen hier im FIS-2-ELAK Server üblicherweise das Eingangsstück als Objektklasse für das neu anzulegende Geschäftsstück.</p> <p>OUTPUT: Fabasoft-Kunden wählen hier im FIS-2-ELAK Server üblicherweise das Ausgangsstück (in der NÖL alternativ die Zuschrift) als Objektklasse für das neu anzulegende Geschäftsstück.</p> <p>INTERNAL: Fabasoft-Kunden wählen hier im FIS-2-ELAK Server üblicherweise das interne Stück als Objektklasse für das neu anzulegende Geschäftsstück.</p>	Purpose-Type	0-1	"INPUT" "OUTPUT" "INTERNAL"

SendData-InputObject	Description	Type	Occur	Restriction
Layer-Control	Mit den Informationen in dieser Struktur kann man einerseits für die SendData-Operation die semantische Bedeutung der Daten innerhalb des EDIAKT-/EDIDOC-Elements im Request konkretisieren. Man kann etwa festlegen, dass im Layer 2 nur die vom Client aktualisierten Adressaten mitgeliefert werden, alle anderen Daten aber unverändert blieben und daher gar nicht mitgeliefert werden. Andererseits kann man für sämtliche ELAK-TRANS-Operationen die Responses genau festlegen; d.h. welche Objekte sollen in der EDIAKT-/EDIDOC-Struktur der Response überhaupt enthalten sein (Selektion), und welche Daten bzw. Details der über die Selektion festgelegten Objekte sollen zurückgeliefert werden (Projektion).	Layer-Control-Type	1	
Ediakt	EDIDOC-Aktenstruktur	edi:Ediakt-Type	1	
Documents	ZIP-File mit allen Dateianhängen ohne Ordnerstruktur. Die Verbindung zwischen Ediakt/Layer0/Payload- und Documents-Knoten ist im Anwendungsfall "6.4.16 Dokument erzeugen" beschrieben.	xsd:base64-Binary	0-1	

5.2.1.2. SendDataOutputObject



SendData-OutputObject	Description	Type	Occur
-----------------------	-------------	------	-------

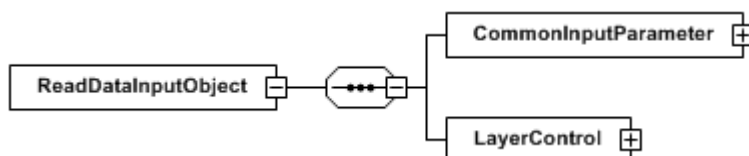
SendData-OutputObject	Description	Type	Occur
Common-Output-Parameter	Allgemeiner Schnittstellen-Parameter für sämtliche Responses. Damit werden die ursprünglichen ELAK-Trans-2.0-Typen "commonApplicationOutputParameter" und "commonElakOutputParameter" fusioniert.	Common-Output-Parameter-Type	1
Ediakt	EDIDOC-Aktenstruktur: Diese wird abhängig von den mit LayerControl ausgewählten Daten befüllt.	edi: EdiaktType	1
Documents	ZIP-File mit allen Dateianhängen ohne Ordnerstruktur. Die Verbindung zwischen Ediakt/Layer0/Payload- und Documents-Knoten ist im Anwendungsfall "6.4.16 Dokument erzeugen" beschrieben.	xsd: base64Binary	0-1

5.2.2. ReadData

Die ReadData Methode dient dazu, ausgehend von FIS, Aktenstrukturen/Geschäftsobjekte im ELAK zu lesen, oder eine URI auf gewisse Geschäftsobjekte im ELAK zu erhalten.

Je nach ELAK-Trans FIS-2-ELAK Server Implementierung und Aufbau des ELAKs kann es beim Lesen der Objekt Metadaten zu unterschiedlichen Ergebnissen kommen, sodass das mittels ReadData gelesene Ediakt Objekt nicht vollständig einem übermittelten SendData Ediakt Objekt entspricht (z.B. entfallen von SpecialData Objekten). Gewisse Metadaten sind allerdings genormt und werden von allen Systemen im selben Knoten geliefert (z.B. Aktenzahl, URI,...).

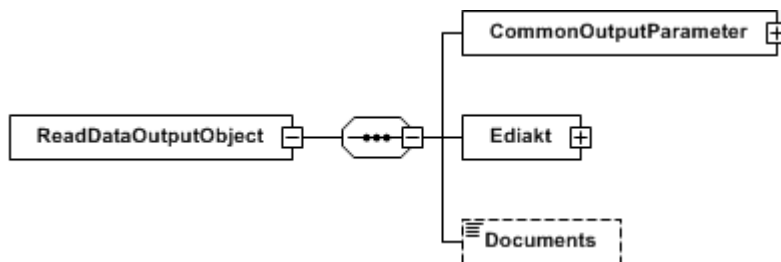
5.2.2.1. ReadDataInputObject



ReadData-InputObject	Description	Type	Occur
CommonInput-Parameter	Damit werden die ursprünglichen ELAK-Trans-2.0 - Typen "commonApplicationInputParameter" und "commonELAKInputParameter" fusioniert.	CommonInput-ParameterType	1

ReadData-InputObject	Description	Type	Occur
LayerControl	Mittels der LayerResponseControlType in der LayerControl wird festgelegt, welche Objekte in der EDIAKT-/EDIDOC-Struktur der Response überhaupt enthalten sind (Selektion), und welche Daten bzw. Details der über die Selektion festgelegten Objekte zurückgeliefert werden sollen(Projektion).	LayerControlType	1

5.2.2.2. ReadDataOutputObject

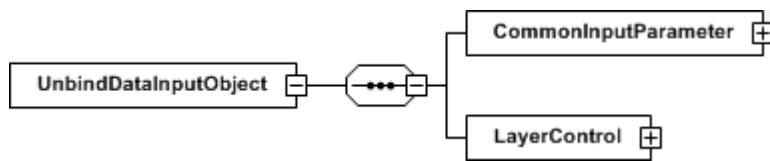


ReadData-OutputObject	Description	Type	Occur
Common-Output-Parameter	Allgemeiner Schnittstellen-Parameter für sämtliche Responses. Damit werden die ursprünglichen ELAK-Trans-2.0-Typen "commonApplicationOutputParameter" und "commonElakOutputParameter" fusioniert.	CommonOutputParameterType	1
Ediakt	EDIDOC-Aktenstruktur: Diese wird abhängig von den mit LayerControl ausgewählten Daten befüllt.	edi:EdiaktType	1
Documents	ZIP-File mit allen Dateianhängen ohne Ordnerstruktur. Die Verbindung zwischen Ediakt/Layer0/Payload- und Documents-Knoten ist im Anwendungsfall "6.4.16 Dokument erzeugen" beschrieben.	xsd:base64Binary	0-1

5.2.3. Unbind

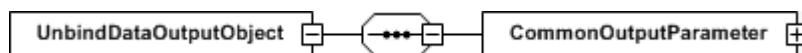
Mit der Unbind Methode informiert das FIS den ELAK darüber, dass die Verbindung zum FIS für das angegebene Objekt gelöst werden soll. Das FIS kann nach Aufruf dieser Methode das Objekt im ELAK nicht mehr korrekt adressieren. Diese Methode dient vor allem der Möglichkeit, Anwender-Fehler zu korrigieren, die durch die Umsetzung der Bidirektionalität entstehen.

5.2.3.1. UnbindDataInputObject



UnbindData-InputObject	Description	Type	Occur
CommonInput-Parameter	Damit werden die ursprünglichen ELAK-Trans-2.0- Typen "commonApplicationInputParameter" und "commonELAKInputParameter" fusioniert.	CommonInput-ParameterType	1
LayerControl	Beim UnbindData werden die (FIS-)IDs der Objekte angegeben, für welche die Verknüpfung aufgehoben werden soll.	LayerControl-Type	1

5.2.3.2. UnbindDataOutputObject



UnbindData-OutputObject	Description	Type	Occur
CommonOutput-Parameter	Allgemeiner Schnittstellen-Parameter für sämtliche Responses. Damit werden die ursprünglichen ELAK-Trans-2.0-Typen "commonApplicationOutputParameter" und "commonElakOutputParameter" fusioniert.	CommonOutput-ParameterType	1

5.3. ELAK -> FIS Kommunikation (ELAK-2-FIS)

5.3.1. SendData

Die SendData Methode dient dazu, ausgehend vom ELAK, folgende UseCases abzudecken:

Aus Sicht ELAK-Trans ELAK-2-FIS Client:

- Daten aktualisieren (L3-L0)
- Geschäftsstück / "Aufgabe" übermitteln (L1)
- Benachrichtigung über Status-Aktualisierung (L3, L1)
- Benachrichtigung über Zustellstatus-Aktualisierung (L1)

Aus Sicht ELAK-Trans ELAK-2-FIS Server:

- FIS-L3-Entität, FIS-L2-Entität, FIS-L1-Entität und/oder FIS-L0-Entität aktualisieren
- Geschäftsstück / "Aufgabe erzeugen" (L1)
- Status-Aktualisierung verarbeiten (L3, L1)
- Zustellstatus-Aktualisierung verarbeiten (L1)

Welche fachlichen FIS-UseCases ausgelöst werden, hängt von der Interpretation der übermittelten Objekte durch die ELAK-Trans ELAK-2-FIS Server-Implementation im FIS ab. Das FIS verarbeitet die Requests nach seinen fachlichen Vorgaben und Regeln.

5.3.1.1. *SendDataInputObject*

Gleiche Datenstruktur wie bei FIS --> ELAK Kommunikation (FIS-2-ELAK)

5.3.1.2. *SendDataOutputObject*

Gleiche Datenstruktur wie bei FIS --> ELAK Kommunikation (FIS-2-ELAK)

5.3.2. ReadData

Die ReadData Methode dient dazu, ausgehend vom ELAK URI's von FIS-Entitäten zu erhalten, welche mit ELAK-Aktenstrukturen verknüpft sind. Der ELAK kann diese zB dafür verwenden, um dem Benutzer in seinem UI die Möglichkeit zu bieten, in einem weiteren Browserfenster oder -tab das FIS mit den zum Kontext passenden FIS-Entitäten zu öffnen.

5.3.2.1. *ReadDataInputObject*

Gleiche Datenstruktur wie bei FIS --> ELAK Kommunikation (FIS-2-ELAK)

5.3.2.2. *ReadDataOutputObject*

Gleiche Datenstruktur wie bei FIS --> ELAK Kommunikation (FIS-2-ELAK)

6. Anwendungsfälle

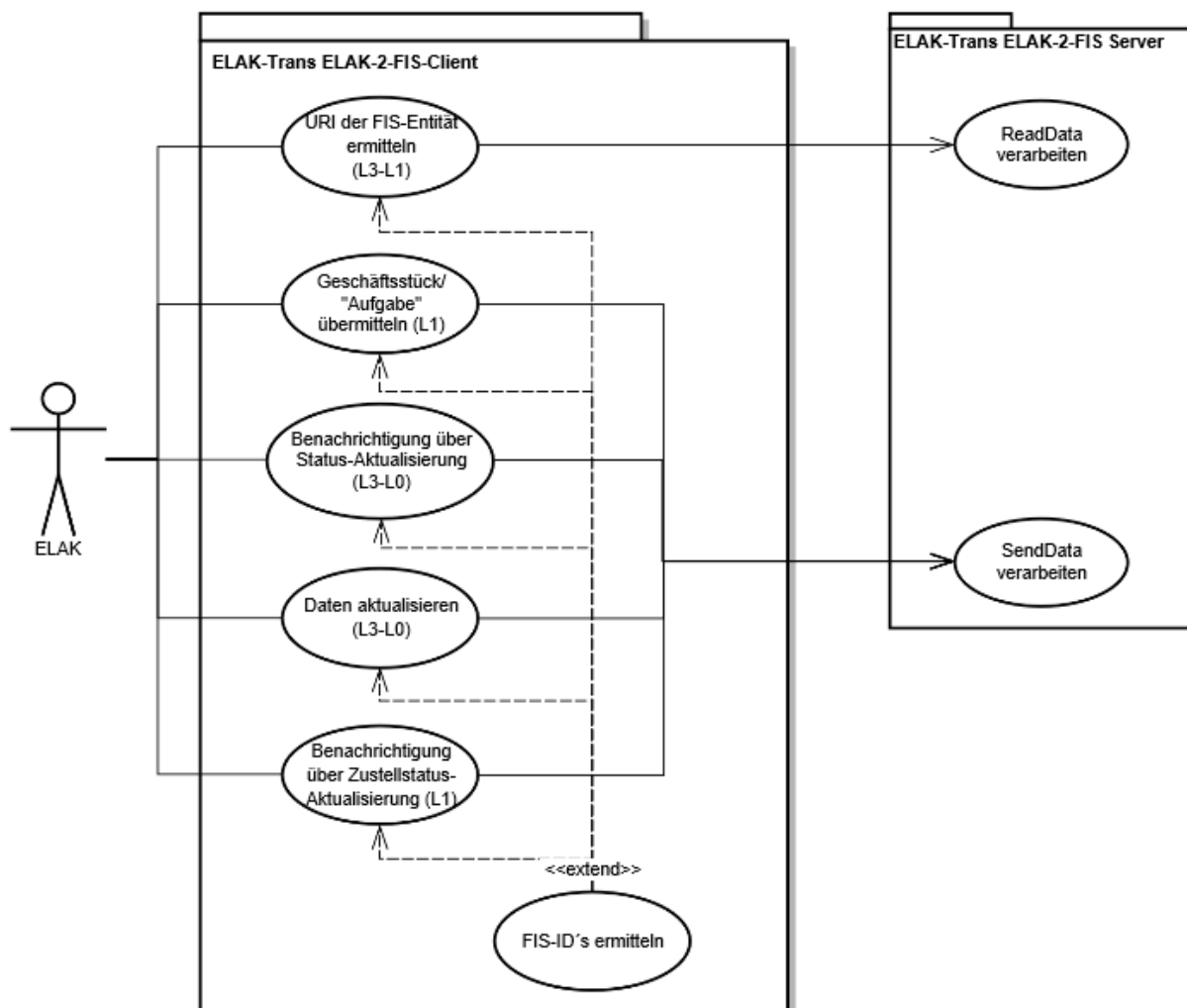
Die Anwendungsfälle beschreiben, welche konkreten Anforderungen an die Integration zwischen ELAK und FIS auf Basis der in der vorhergehenden Kapiteln beschriebenen ELAK-Trans-WS-Methoden und -Datenstrukturen wie abgedeckt werden können.

Die ELAK-Trans FIS-2-ELAK-Server-Anwendungsfälle beschreiben zusätzlich den Ablauf der Verarbeitung im ELAK-Trans FIS-2-ELAK-Server.

6.1. ELAK-Trans ELAK-2-FIS Client

Mit dem "ELAK-Trans ELAK-2-FIS Client" initiiert der ELAK Zugriffe auf das FIS.

Abbildung 5 – Anwendungsfall-Diagramm ELAK-2-FIS-Client



6.1.1. Daten aktualisieren (L3-L0)

Kurzbeschreibung

Dieser Anwendungsfall wird durch den ELAK initiiert, um dem FIS mitzuteilen, dass sich potenziell für das FIS relevante Informationen im ELAK geändert haben. Wurden beispielsweise die Adressaten-Daten (Participants) eines Geschäftsstücks oder Aktes geändert, könnte dies auf diesem Weg dem FIS mitgeteilt werden.

Der ELAK kann über die SendControl-Struktur dem FIS Hinweise darauf geben, welche Teile eines Layers verändert wurden. In der EDIDOC-Datenstruktur müssen neben den FIS-ID's (LayerControl.Layer(3-0).ReferencedIdentifier.Identification) nur jene Teile der Datenstruktur befüllt sein, welche in der SendControl-Struktur als verändert gekennzeichnet sind.

Vorbedingung

Der ELAK-Trans ELAK-2-FIS Client hat die FIS-ID's in der Objekt Mapping Tabelle bereits gefunden.

6.1.2. Geschäftsstück/ "Aufgabe" übermitteln (L1)

Kurzbeschreibung

Dieser Anwendungsfall wird durch den ELAK initiiert, um ein Geschäftsstück zur weiteren Bearbeitung an das FIS zu übergeben. Der ELAK-Trans ELAK-2-FIS Client legt die im Response vom ELAK-Trans ELAK-2-FIS Server zurückgegebenen FIS-ID's gemeinsam mit den ursprünglich vom ELAK übergebenen technischen ELAK-ID's (zB Objektadressen) in der Objekt Mapping Tabelle ab.

Beispiel-Responses vom "ELAK-Trans ELAK-2-FIS Server" an den "ELAK-Trans ELAK-2-FIS Client":

- FIS gibt gültige FIS-ID's für alle Ebenen zurück

L3	L2	L1	Aktion
8A16140E-BD9C-4CAA-87B3-895BC7A913C6			Mapping eintragen
8A16140E-BD9C-4CAA-87B3-895BC7A913C6	F22211F8-EDC5-40F4-950B-E12572215E5F		Mapping eintragen
8A16140E-BD9C-4CAA-87B3-895BC7A913C6	F22211F8-EDC5-40F4-950B-E12572215E5F	B6473B30-B5D4-40DA-AA71-D22B0C6889D0	Mapping eintragen

- FIS gibt 00000000-0000-0000-0000-000000000000 als L3- und L2-FIS-ID's zurück

L3	L2	L1	Aktion
00000000-0000-0000-0000-000000000000			Mapping nicht eintragen
00000000-0000-0000-0000-000000000000	00000000-0000-0000-0000-000000000000		Mapping nicht eintragen

00000000-0000-0000-0000-0000-000000000000	00000000-0000-0000-0000-000000000000	B6473B30-B5D4-40DA-AA71-D22B0C6889D0	Mapping eintragen
---	--------------------------------------	--------------------------------------	-------------------

Wenn für L3 oder L2 als FIS-ID 00000000-0000-0000-0000-000000000000 zurückgegeben wird, werden für L3 und L2 keine Einträge in der Objekt Mapping Tabelle erzeugt. Nur für L1 und eventuelle L0 - Objekte.

ELAK-Trans regelt nicht, was nach dieser Übergabe mit dem Geschäftsstück im ELAK passiert, dies ist der jeweiligen Implementierung überlassen. Denkbar wäre:

- Der Prozess des Geschäftsstücks im ELAK wird beendet und das Geschäftsstück wird aus dem ELAK-Arbeitsvorrat entfernt.
- Der Prozess des Geschäftsstücks im ELAK wird pausiert, bis zu einem späteren Zeitpunkt eine a) (Status-)Aktualisierung oder b) ein Lösen der Verknüpfung durch das FIS erfolgt.
- Der Prozess des Geschäftsstücks im ELAK läuft weiter. Es ist organisatorisch geregelt, welche weiteren Bearbeitungsschritte im FIS und welche im ELAK erfolgen.

6.1.3. Benachrichtigung über Status-Aktualisierung (L3-L0)

Kurzbeschreibung

Dieser Anwendungsfall wird durch den ELAK initiiert, um das FIS über eine Status-Änderung eines Aktes oder Geschäftsstücks zu informieren.

Diese Statusänderungen können sein:

- Der Akt wurde abgeschlossen. (L3; Metadaten-Closed = True)
- Das Geschäftsstück wurde genehmigt. (L1; Metadata-Approved = True)
- Das Geschäftsstück wurde abgelehnt. (L1; Metadata-Approved = False)
- Das Objekt (L3-L0) wurde storniert. (L3-L0; Metadata-Canceled = True)

Vorbedingung

Der ELAK-Trans ELAK-2-FIS Client hat die FIS-ID's in der Objekt Mapping Tabelle bereits gefunden.

6.1.4. Benachrichtigung über Zustellstatus-Aktualisierung (L1)

Kurzbeschreibung

Dieser Anwendungsfall wird durch den ELAK initiiert, um das FIS über die erfolgte Zustellung eines Geschäftsstücks an einen Adressaten zum mitgegebenen Zustelldatum zu informieren.

Dieser Anwendungsfall findet nur dann Anwendung, wenn der Versand über den ELAK erfolgt und der ELAK die Rückmeldung über die erfolgte Zustellung erhält. Stellt das FIS selbst über die DualeZuse zu, ist dieser Anwendungsfall nicht relevant.

Vorbedingung

Der ELAK-Trans ELAK-2-FIS Client hat die FIS-ID's in der Objekt Mapping Tabelle bereits gefunden.

6.1.5. URI der FIS-Entität ermitteln (L3-L1)

Kurzbeschreibung

Dieser Anwendungsfall wird durch den ELAK initiiert, um URI's von FIS-Entitäten zu erhalten, welche mit ELAK-Aktenstrukturen verknüpft sind. Der ELAK kann diese zB dafür verwenden, um dem Benutzer in seinem UI die Möglichkeit zu bieten, in einem weiteren Browserfenster oder -tab das FIS mit den zum Kontext passenden FIS-Entitäten zu öffnen.

6.1.6. FIS-ID's ermitteln

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS-Client bekommt vom ELAK die technischen ELAK-ID's (zB Objektadressen) übergeben und ermittelt mit Hilfe der Objekt Mapping Tabelle die dazugehörigen FIS-ID's.

Auslöser

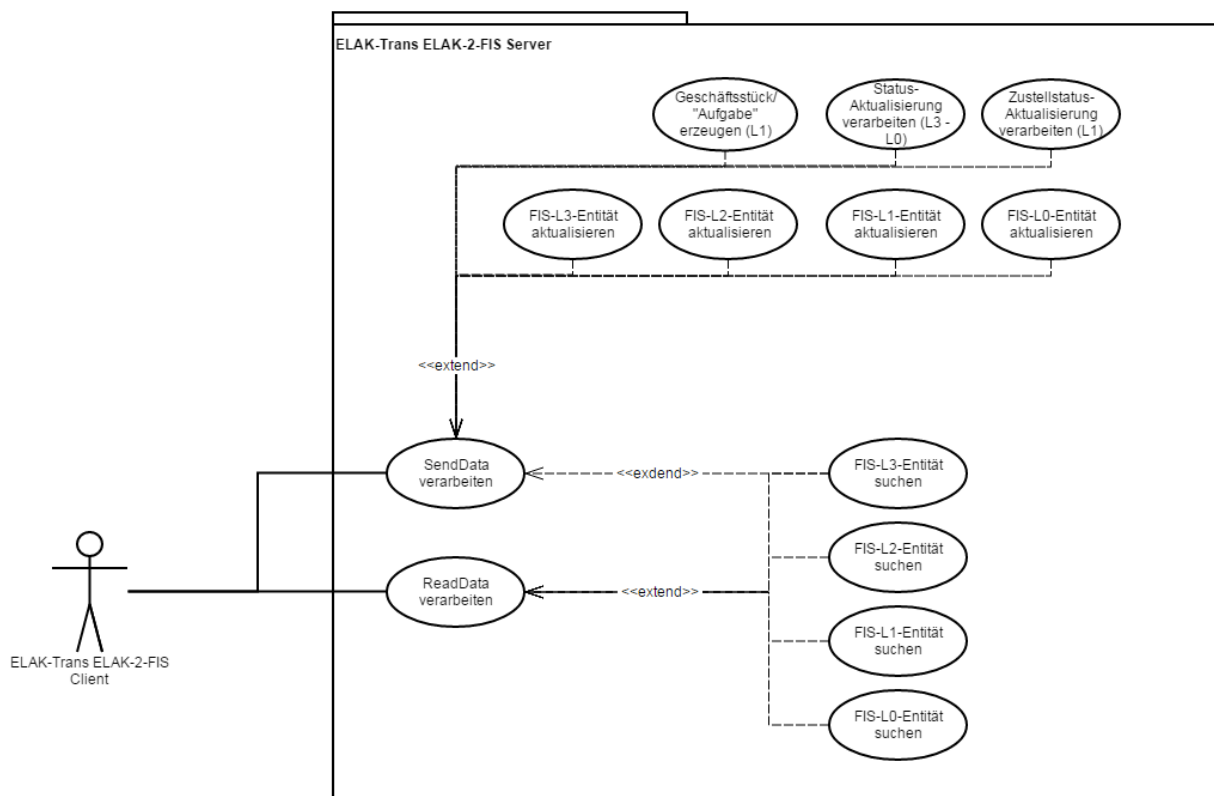
Der Anwendungsfall wird von den Anwendungsfällen "Daten aktualisieren (L3-L0)", "Geschäftsstück/ Aufgabe übermitteln (L1)", "Benachrichtigung über Status-Aktualisierung (L3-L0)", "Benachrichtigung über Zustellstatus-Aktualisierung (L1)" und "URL der FIS-Entität ermitteln (L3-L1)" aufgerufen.

6.2. ELAK-Trans ELAK-2-FIS Server

Der ELAK-Trans ELAK-2-FIS Server übernimmt die Verarbeitung von ELAK-Trans Requests auf der Seite des FIS. Der Server kann dabei zwei Arten von Requests verarbeiten, die als entsprechende Anwendungsfälle abgebildet wurden: SendData und ReadData.

Die durch diese Anwendungsfälle getriggerten Operationen im FIS richten sich an den fachlichen Anforderungen an das FIS aus und sind nicht durch ELAK-Trans normiert.

Abbildung 6 – Anwendungsfall-Diagramm ELAK-2-FIS-Server



6.2.1. FIS-L3-L0-Entität suchen

Kurzbeschreibung

Der ELAK-Trans ELAK 2-FIS Server sucht mit Hilfe der FIS-ID (LayerControl.Layer(3-0).ReferencedIdentifier.Identification) in der FIS-Datenbank nach der FIS-Entität.

Auslöser

Der Anwendungsfall wird von den Anwendungsfällen "SendData verarbeiten" und "ReadData verarbeiten" aufgerufen.

6.2.2. ReadData verarbeiten

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS Client sendet einen ELAK-Trans ReadData Aufruf an den ELAK-Trans ELAK-2-FIS Server. In dem Aufruf wird mittels der LayerControl-Datenstruktur angegeben, welche Daten vom FIS gelesen werden sollen. Der ELAK-Trans ELAK-2-FIS Server liest die entsprechenden Daten aus dem FIS und verarbeitet sie in eine Antwort. Es wird davon ausgegangen, dass ELAK-2-FIS nur das Ermitteln der URI's der FIS-Entitäten implementiert wird.

Auslöser

Der ELAK-Trans ELAK-2-FIS Server empfängt einen ELAK-Trans ReadData Aufruf eines ELAK-Trans ELAK-2-FIS Clients.

6.2.3. SendData verarbeiten

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS Client sendet einen ELAK-Trans SendData Aufruf an den ELAK-Trans ELAK-2-FIS Server. Der ELAK-Trans ELAK-2-FIS Server führt die nötigen Operationen zur Verarbeitung des Pakets durch. Das ELAK-Trans Paket kann mit dem FIS verknüpfte Objekte (Akt, Geschäftsfall, Geschäftsstück, Dokument) aber auch neue Objekte (Geschäftsstück, Dokument) enthalten.

Auslöser

Der ELAK-Trans ELAK-2-FIS Server empfängt einen ELAK-Trans SendData Aufruf eines ELAK-Trans ELAK-2-FIS Clients.

6.2.4. Geschäftsstück/“Aufgabe“ erzeugen (L1)

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS Server bekommt vom ELAK-Trans ELAK-2-FIS Client ein Geschäftsstück (sinngemäße "Aufgabe") zur weiteren Bearbeitung übergeben. Auf Basis der übergebenen FIS-ID's (LayerControl.Layer(3-0).ReferencedIdentifier.Identification) wird beurteilt, ob eine automatisierte Zuordnung zu bestehenden FIS-Entitäten möglich ist. Der ELAK-Trans ELAK-2-FIS Server legt im FIS eine Aufgabe an, die in weiterer Folge von einem Benutzer des FIS weiterbearbeitet wird.

Der ELAK-Trans ELAK-2-FIS Server vergibt für das L1-Objekt sowie für eventuell übergebene L0-Objekte FIS-Entitäten (und somit FIS-ID's) und gibt diese an den ELAK-Trans ELAK-2-FIS Client zurück.

Welche L3- und L2-FIS-ID's werden zurückgegeben?

- Diese sind im eingehenden Request bereits befüllt --> 1:1 zurückgegeben
- Diese sind im eingehenden Request leer
 - Variante 1: Das FIS gibt als L3- und L2-FIS-ID's 00000000-0000-0000-0000-000000000000 zurück. (Somit noch keine Zuordnung zu L3- und L2-FIS-Entitäten)
 - Variante 2: Das FIS erzeugt L3- und L2-FIS-Entitäten (und somit auch -ID's) "on-the-fly" und gibt diese zurück.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

6.2.5. Status-Aktualisierung verarbeiten (L3-L0)

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS Server reagiert auf Status-Änderungen von L3-L0-Objekten und führt im Bedarfsfall die im fachlichen Kontext des FIS zielführenden Operationen aus.

Diese könnten beispielsweise sein:

- Der Akt wurde abgeschlossen. (L3; Metadaten-Closed = True)
 - Prüfen, ob die dazugehörige FIS-Entität ebenfalls abgeschlossen ist bzw. werden kann. Falls nicht zB a) erneutes Öffnen des Aktes oder b) Anlegen eines neuen Aktes oder c) Erstellen einer Aufgabe für die manuelle Bearbeitung dieses Status durch einen Benutzer.
- Das Geschäftsstück wurde genehmigt. (L1; Metadata-Approved = True)
 - Fortführung des fachlichen Prozesses im FIS. Beispielsweise a) Erzeugen einer Aufgabe, zur weiteren Bearbeitung durch einen Benutzer oder b) Anstoßen der Auszahlung an den Förderungswerber.
- Das Geschäftsstück wurde abgelehnt. (L1; Metadata-Approved = False)

- Wiederaufsetzen des fachlichen Prozesses im FIS. zB Erstellen einer Aufgabe, zur erneuten Bearbeitung durch einen Benutzer.
- Das Objekt (L3-L0) wurde storniert. (L3-L0; Metadata-Canceled = True)
 - Prüfen, ob die dazugehörigen FIS-Entitäten ebenfalls storniert werden können.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans ELAK-2-FIS Server hat die FIS-Entität(en) bereits gefunden.

6.2.6. Zustellstatus-Aktualisierung verarbeiten (L1)

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS Server reagiert auf die Änderung des Zustellstatus eines (oder mehrerer) Adressaten eines Geschäftsstücks. Er übernimmt automatisiert das Zustelldatum in die FIS-Datenbank. Falls fachlich notwendig, kann darüber hinaus eine Aufgabe oder Benachrichtigung für den FIS-Benutzer erzeugt werden, in welcher dieser darüber informiert wird.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans ELAK-2-FIS Server hat die FIS-L1-Entität bereits gefunden.

6.2.7. FIS-L3-L0-Entität aktualisieren

Kurzbeschreibung

Der ELAK-Trans ELAK-2-FIS Server reagiert auf die Änderung von L3-L0-Objekten und führt im Bedarfsfall im fachlichen Kontext des FIS zielführende Operationen aus.

Diese könnten beispielsweise sein:

- Automatisiertes Aktualisieren von verknüpften FIS-Entitäten (zB Adressaten-Daten des ELAK-Aktes mit den FIS-Personenstammdaten abgleichen)
- Erstellen einer Aufgabe für einen Benutzer, die ihn anweist, einen Abgleich zwischen ELAK-Objekt und FIS-Entität vorzunehmen
- ignorieren

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

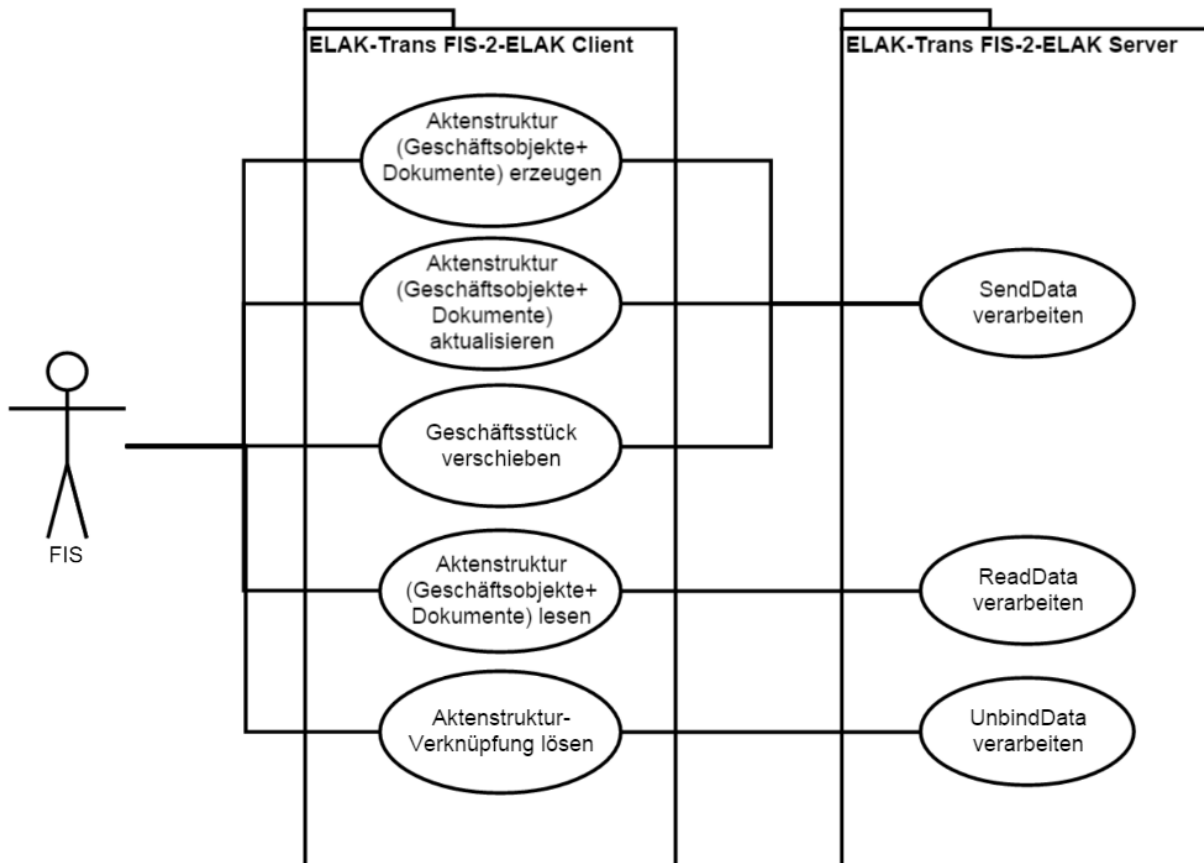
Vorbedingung

Der ELAK-Trans ELAK-2-FIS Server hat die FIS-Entität(en) bereits gefunden.

6.3. ELAK-Trans FIS-2-ELAK Client

Mit dem "ELAK-Trans FIS-2-ELAK Client" initiiert das FIS Zugriffe auf den ELAK.

Abbildung 7 – Anwendungsfall-Diagramm FIS-2-ELAK-Client



6.3.1. Aktenstruktur (Geschäftsobjekte+Dokumente) erzeugen

Kurzbeschreibung

Das FIS ruft diesen Anwendungsfall auf, um neue Informationen im ELAK abzulegen. Mit dem Anwendungsfall ist es möglich, folgende Anforderungen abzudecken:

- Es ist möglich, in einem Request die gesamte Aktenstruktur (Akt, Geschäftsfall, Geschäftsstück und Dokumente) anzulegen und zu verknüpfen. Es muss nicht eine Anfrage pro hierarchischem Objekt gestellt werden.
- Es ist möglich, neue Objekte (Geschäftsfälle, Geschäftsstücke und Dokumente) in bereits bestehende übergeordnete Objekte einzufügen.
- Es ist möglich, durch Angabe von ELAK-Identifiern (Aktenzahl/Geschäftszahl) neue Geschäftsstücke in bereits im ELAK existierenden Akten/Geschäftsfälle abzulegen, die noch nicht mit dem FIS verknüpft sind. Ein Seiteneffekt der Verwendung der ELAK-Identifizier ist, die Verknüpfung eines bisher unverknüpften Ziel-Aktes und Ziel-Geschäftsfalls mit der jeweils entsprechenden FIS ID.

6.3.2. Aktenstruktur (Geschäftsobjekte+Dokumente) aktualisieren

Kurzbeschreibung

Das FIS aktualisiert die Daten eines beliebigen Objektes, das mit dem ELAK verknüpft ist. Das FIS ruft den Anwendungsfall auf, um die Aktualisierung an den ELAK weiter zu geben. Eine Aktualisierung kann auf allen Ebenen passieren (Layer3-Layer0) und sowohl Metadaten, Statusübergänge als auch Änderungen im Dokument (Layer0 Payload) umfassen.

Das FIS kann über die SendControl-Struktur genau steuern, welche Teile eines Layers aktualisiert werden sollen.

6.3.3. Geschäftsstück verschieben

Kurzbeschreibung

Das FIS ruft diesen Anwendungsfall auf, um speziell Geschäftsstück-Objekte (Gruppierung um ein oder mehrere Dokumente) im ELAK umzuprotokollieren.

Der Anwendungsfall wird vom FIS in den folgenden 2 Fällen benötigt:

- Fehlerkorrektur:
Das Geschäftsstück wurde irrtümlich dem falschen übergeordneten Objekt zugeordnet und entsprechend im ELAK falsch protokolliert. Dies soll nun vom FIS ausgehend korrigiert werden.
- Bidirektionalität:
Das Geschäftsstück wurde im ELAK angelegt (unprotokolliert oder in einem Sammelakt) und entsprechend im FIS registriert. Nun wird das Geschäftsstück im FIS dem korrekten übergeordneten Objekt zugewiesen und soll im ELAK auch entsprechend (um-) protokolliert werden.

6.3.4. Aktenstruktur (Geschäftsobjekte+Dokumente) lesen

Kurzbeschreibung

Mit diesem Anwendungsfall greift der ELAK-Trans FIS-2-ELAK Client lesend auf den ELAK zu. Dabei ist es sowohl möglich die Daten komplett aus dem ELAK zu lesen, als auch nur URIs auf die entsprechenden Objekte im ELAK abzufragen. Das FIS ruft diesen Anwendungsfall auf, wenn...

- ... das FIS die Daten und Dokumente direkt aus dem ELAK im FIS anzeigen und weiterverarbeiten will.
- ... das FIS einen Abprungpunkt zu einem ELAK Objekt zur Verfügung stellen will. Es ist dabei zu beachten, dass die übermittelten URIs des ELAKs nicht als statisch anzusehen sind und somit nicht im FIS gespeichert werden dürfen, sondern zur Laufzeit angefragt werden müssen. (Nur kurzfristiges Cachen sinnvoll.)

6.3.5. Aktenstruktur-Verknüpfung lösen

Kurzbeschreibung

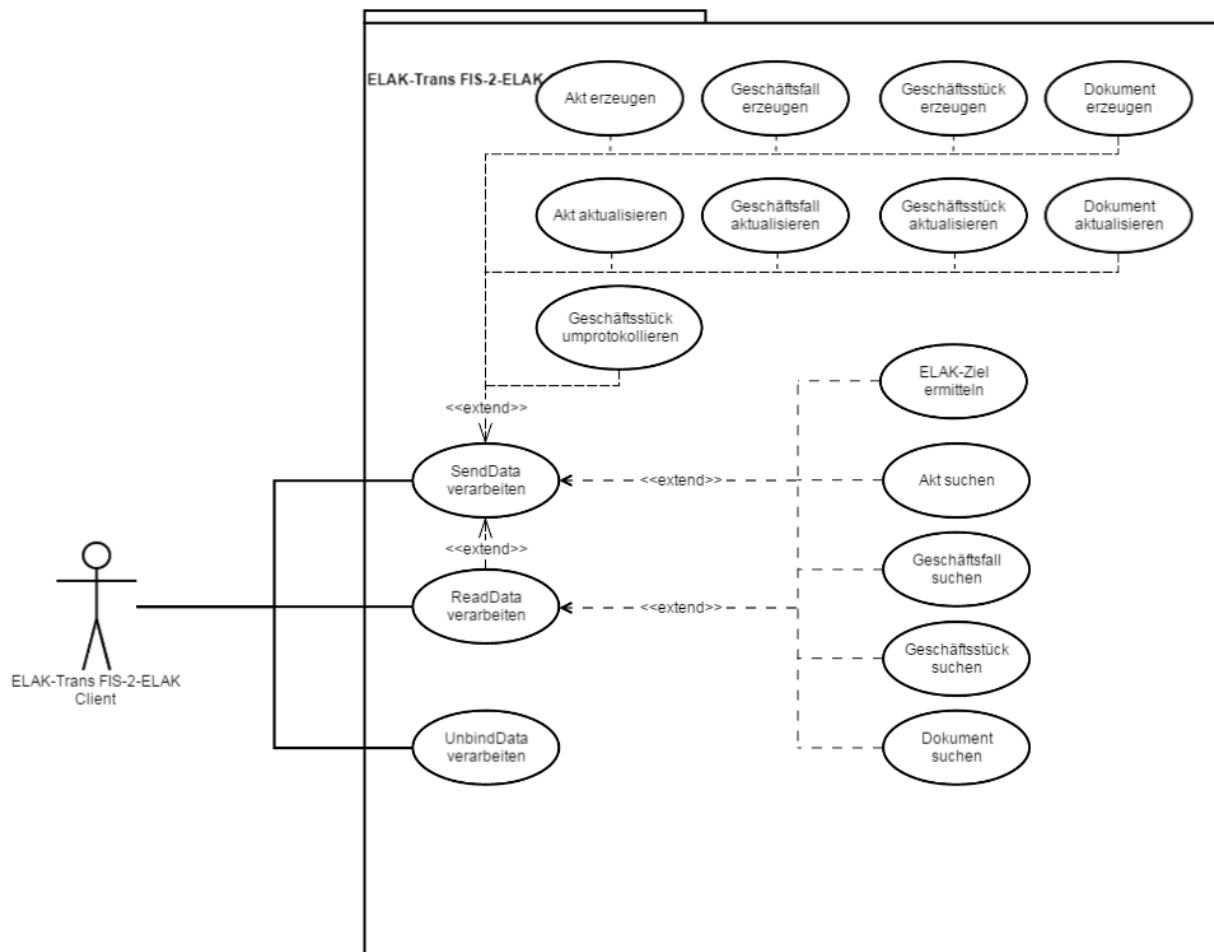
Der Anwendungsfall wird vom FIS aufgerufen wenn...

- ... das Objekt zuvor mittels "Geschäftsstück/Aufgabe übermitteln" vom ELAK an das FIS übergeben wurde. Im FIS der Bearbeiter aber feststellt, dass das Geschäftsstück für das FIS nicht relevant ist und wieder im ELAK weiterbearbeitet werden muss.
- ... das FIS eine Skartierung durchführt und die ELAK Verknüpfung nicht mehr verwendet.

6.4. ELAK-Trans FIS-2-ELAK Server

Der ELAK-Trans FIS-2-ELAK Server übernimmt die Verarbeitung von ELAK-Trans Requests auf der Seite des ELAK. Der Server kann dabei drei Arten von Requests verarbeiten, die als entsprechende Anwendungsfälle abgebildet wurden: SendData, ReadData und UnbindData.

Abbildung 8 – Anwendungsfall-Diagramm FIS-2-ELAK-Server



6.4.1. Akt suchen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server sucht mit Hilfe der Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification) in der Objekt Mapping Tabelle nach dem Akt.

Auslöser

Der Anwendungsfall wird von den Anwendungsfällen "SendData verarbeiten" und "ReadData verarbeiten" aufgerufen.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Akt Objekt mit:
Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
Wird ein Akt gefunden?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server gibt das Akt Objekt zurück.
 2. NEIN: Der ELAK-Trans FIS-2-ELAK Server gibt kein Akt Objekt an den aufrufenden Anwendungsfall zurück.

6.4.2. Geschäftsfall suchen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server sucht mit Hilfe der Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification) und der GF-FIS-ID (LayerControl.[...].Layer2.ReferencedIdentifier.Identification) in der Objekt Mapping Tabelle nach dem Geschäftsfall.

Auslöser

Der Anwendungsfall wird von den Anwendungsfällen "SendData verarbeiten" und "ReadData verarbeiten" aufgerufen.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Geschäftsfall mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification.Wurde ein Geschäftsfall gefunden?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server gibt den gefundene Geschäftsfall an den aufrufenden Anwendungsfall zurück.
 2. NEIN: Der ELAK-Trans FIS-2-ELAK Server gibt kein Geschäftsfall Objekt an den aufrufenden Anwendungsfall zurück.

6.4.3. Geschäftsstück suchen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server sucht mit Hilfe der Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification), der GF-FIS-ID (LayerControl.[...].Layer2.ReferencedIdentifier.Identification) und der GS-FIS-ID (LayerControl.[...].Layer1.ReferencedIdentifier.Identification) in der Objekt Mapping Tabelle nach dem Geschäftsstück.

Auslöser

Der Anwendungsfall wird von den Anwendungsfällen "SendData verarbeiten" und "ReadData verarbeiten" aufgerufen.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Geschäftsstück mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
2. Wurde ein Geschäftsstück gefunden?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server gibt das gefundene Geschäftsstück an den aufrufenden Anwendungsfall zurück.
 2. NEIN: Der ELAK-Trans FIS-2-ELAK Server gibt kein Geschäftsstück an den aufrufenden Anwendungsfall zurück.

6.4.4. Dokument suchen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server sucht mit Hilfe

- der Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification),
- der GF-FIS-ID (LayerControl.[...].Layer2.ReferencedIdentifier.Identification),
- der GS-FIS-ID (LayerControl.[...].Layer1.ReferencedIdentifier.Identification) und
- der Dok-FIS-ID (LayerControl.[...].Layer0[i].ReferencedIdentifier.Identification)

in der Objekt Mapping Tabelle nach dem Dokument.

Auslöser

Der Anwendungsfall wird von den Anwendungsfällen "SendData verarbeiten" und "ReadData verarbeiten" aufgerufen.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Dokument mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
 - Dok-FIS-ID = LayerControl.[...].Layer0[i].ReferencedIdentifier.Identification
2. Wurde ein Dokument gefunden?
 - JA: Der ELAK-Trans FIS-2-ELAK Server gibt das gefundene Dokument an den aufrufenden Anwendungsfall zurück.
 - NEIN: Der ELAK-Trans FIS-2-ELAK Server gibt kein Dokument an den aufrufenden Anwendungsfall zurück.

6.4.5. ELAK-Ziel ermitteln

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server analysiert die übergebenen Daten des Requests und überprüft einerseits, ob er für die Nachricht zuständig ist und entscheidet, welcher Mandant im ELAK das Ziel für die Verarbeitung des ELAK-Trans Paketes ist.

Zur Entscheidung dienen vor allem die Datenstruktur CommonInputParameter sowie der PVP-Header für Informationen zum aufrufenden Anwender.

Die Umsetzung dieses Anwendungsfalles kann für jeden ELAK sehr unterschiedlich sein und hängt stark von der Art der Aktenführung im jeweiligen ELAK ab.

6.4.6. ReadData verarbeiten

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Client sendet einen ELAK-Trans ReadData Aufruf an den ELAK-Trans FIS-2-ELAK Server. In dem Aufruf wird angegeben, welche Daten vom ELAK gelesen werden sollen. Der ELAK-Trans FIS-2-ELAK Server liest die entsprechenden Daten aus dem ELAK und verarbeitet sie in eine Antwort. Entsprechend der Datenstruktur LayerControl.LayerXResponseControl, ist es sowohl möglich, die Daten direkt zu lesen, als auch die URIs der angeforderten Objekte zurückzugeben. Eine Abfrage kann maximal einen Akt (Layer3) aber 0-n Geschäftsfälle, 0-n Geschäftsstücke und 0-n Dokumente betreffen. Die Objekte unterhalb des Aktes müssen nicht zwangsläufig bereits mit dem FIS verknüpft sein.

Auslöser

Der ELAK-Trans FIS-2-ELAK Server empfängt einen ELAK-Trans ReadData Aufruf eines ELAK-Trans FIS-2-ELAK Clients.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK ruft den Anwendungsfall ELAK-Ziel ermitteln auf.
2. Der ELAK-Trans FIS-2-ELAK Server ruft für den übermittelten LayerControl.Layer3 den Anwendungsfall "Akt suchen" auf.
3. Der ELAK-Trans FIS-2-ELAK Server erstellt einen Layer3 abhängig von den Include-Parametern (ausgenommen IncludeUnspecifiedCoupledObjects und IncludeUncoupledObjects) der LayerControl.Layer3ResponseControl.
4. Wurde ein Layer2 verwendet?
 1. JA:
 1. Wurde LayerControl.Layer2ResponseControl.IncludeUnspecifiedCoupledObjects=true oder LayerControl.Layer2ResponseControl.IncludeUncoupledObjects=true übermittelt?
 - a. JA: Wurde Layer2ResponseControl.IncludeUnspecifiedCoupledObjects=true übermittelt?
 - i. JA: Der ELAK-Trans FIS-2-ELAK Server sucht nach allen Geschäftsfällen in der Objekt Mapping Tabelle, die mit dem gefundenen Akt verknüpft sind.
 - ii. NEIN (LayerControl.Layer2ResponseControl.IncludeUncoupledObjects=true): Der ELAK-Trans FIS-2-ELAK Server sucht nach allen Geschäftsfällen im ELAK, die mit dem gefundenen Akt verknüpft sind.
 - b. NEIN (einzelne Layer2 Objekte): Der ELAK-Trans FIS-2-ELAK Server ruft für den übermittelten LayerControl.Layer3 und LayerControl.Layer3.Layer2[i] den Anwendungsfall "Geschäftsfall suchen" auf.
 2. Der ELAK-Trans FIS-2-ELAK Server erstellt für alle gefundenen Geschäftsfälle entsprechende Layer2 abhängig von den Include-Parametern (ausgenommen IncludeUnspecifiedCoupledObjects und IncludeUncoupledObjects) der LayerControl.Layer2ResponseControl.
 3. Der ELAK-Trans FIS-2-ELAK Server fügt die Layer2 in den bereits erstellten Layer3 ein.
 2. NEIN (Kein Layer2): Der ELAK-Trans FIS-2-ELAK Server fügt einen leeren Layer2 in den bereits erstellten Layer3 ein.
5. Wurde LayerControl.Layer1ResponseControl.IncludeUnspecifiedCoupledObjects=true oder LayerControl.Layer1ResponseControl.IncludeUncoupledObjects=true übermittelt?
 1. JA: Wurde Layer1ResponseControl.IncludeUnspecifiedCoupledObjects=true übermittelt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server sucht nach allen Geschäftsstücken in der Object Mapping Tabelle, die mit den gefundenen Geschäftsfällen bzw. dem Akt (wenn keine Geschäftsfälle verwendet werden) verknüpft sind.
 2. NEIN (LayerControl.Layer1ResponseControl.IncludeUncoupledObjects=true): Der ELAK-Trans FIS-2-ELAK Server sucht nach allen Geschäftsstücken im ELAK, die

mit den gefundenen Geschäftsfällen bzw. Akt (wenn keine Geschäftsfälle verwendet werden) verknüpft sind.

2. NEIN (einzelne Layer1 Objekte): Der ELAK-Trans FIS-2-ELAK Server ruft für den übermittelten LayerControl.Layer3, LayerControl.Layer3.Layer2[i], LayerControl.Layer3.Layer2[i].Layer1[j] den Anwendungsfall "Geschäftsstück suchen" auf.
6. Der ELAK-Trans FIS-2-ELAK Server erstellt für alle gefundenen Geschäftsstücke entsprechende Layer1 abhängig von den Include-Parametern (ausgenommen IncludeUnspecifiedCoupledObjects und IncludeUncoupledObjects) der LayerControl.Layer1ResponseControl.
7. Der ELAK-Trans FIS-2-ELAK Server fügt die Layer1 in die entsprechenden, bereits erstellten Layer2 ein.
8. Wurde LayerControl.Layer0ResponseControl.IncludeUnspecifiedCoupledObjects=true oder LayerControl.Layer0ResponseControl.IncludeUncoupledObjects=true übermittelt?
 1. JA: Wurde Layer0ResponseControl.IncludeUnspecifiedCoupledObjects=true übermittelt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server sucht nach allen Dokumenten in der Objekt Mapping Tabelle, die mit den gefundenen Geschäftsstücken verknüpft sind.
 2. NEIN (LayerControl.Layer0ResponseControl.IncludeUncoupledObjects=true): Der ELAK-Trans FIS-2-ELAK Server sucht nach allen Dokumenten im ELAK, die mit gefundenen Geschäftsstücken verknüpft sind.
 2. NEIN (einzelne Layer0 Objekte): Der ELAK-Trans FIS-2-ELAK Server ruft für den übermittelten LayerControl.Layer3, LayerControl.Layer3.Layer2[i], LayerControl.Layer3.Layer2[i].Layer1[j], LayerControl.Layer3.Layer2[i].Layer1[j].Layer0[k] den Anwendungsfall "Dokument suchen" auf.
9. Der ELAK-Trans FIS-2-ELAK Server erstellt für alle gefundenen Dokumente entsprechende Layer0 abhängig von den Include-Parametern (ausgenommen IncludeUnspecifiedCoupledObjects und IncludeUncoupledObjects) der LayerControl.Layer0ResponseControl.
10. Der ELAK-Trans FIS-2-ELAK Server fügt die Layer0 in die entsprechenden, bereits erstellten Layer1 ein.
11. Der ELAK-Trans FIS-2-ELAK Server erstellt eine Antwort mit dem erstellten EDIDOC Paket.
12. Der ELAK-Trans FIS-2-ELAK Server schickt die Antwort an den ELAK-Trans FIS-2-ELAK Client.

6.4.7. SendData verarbeiten

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Client sendet einen ELAK-Trans SendData Aufruf an den ELAK-Trans FIS-2-ELAK Server. Das ELAK-Trans Paket kann neue oder bereits im ELAK existierende Objekte (Akt, Geschäftsfall, Geschäftsstück, Dokument) enthalten. Der ELAK-Trans FIS-2-ELAK Server führt die nötigen Operationen zur Verarbeitung des Pakets durch. Die Verarbeitung beinhaltet Aktionen für das Erzeugen und Aktualisieren von Akt, Geschäftsfall, Geschäftsstück und Dokument Objekten sowie das (Um-)Protokollieren von Geschäftsstücken. Der SendData Request besteht aus 1 Akt (Layer3), 0-1 Geschäftsfall (Layer2), 0-1 Geschäftsstück (Layer1), 0-n Dokumente (Layer0). Weitere Informationen zu

diesem Anwendungsfall befinden sich im Best Practice Dokument im Abschnitt ELAK-Trans-Server - Implementierungsvorlage.

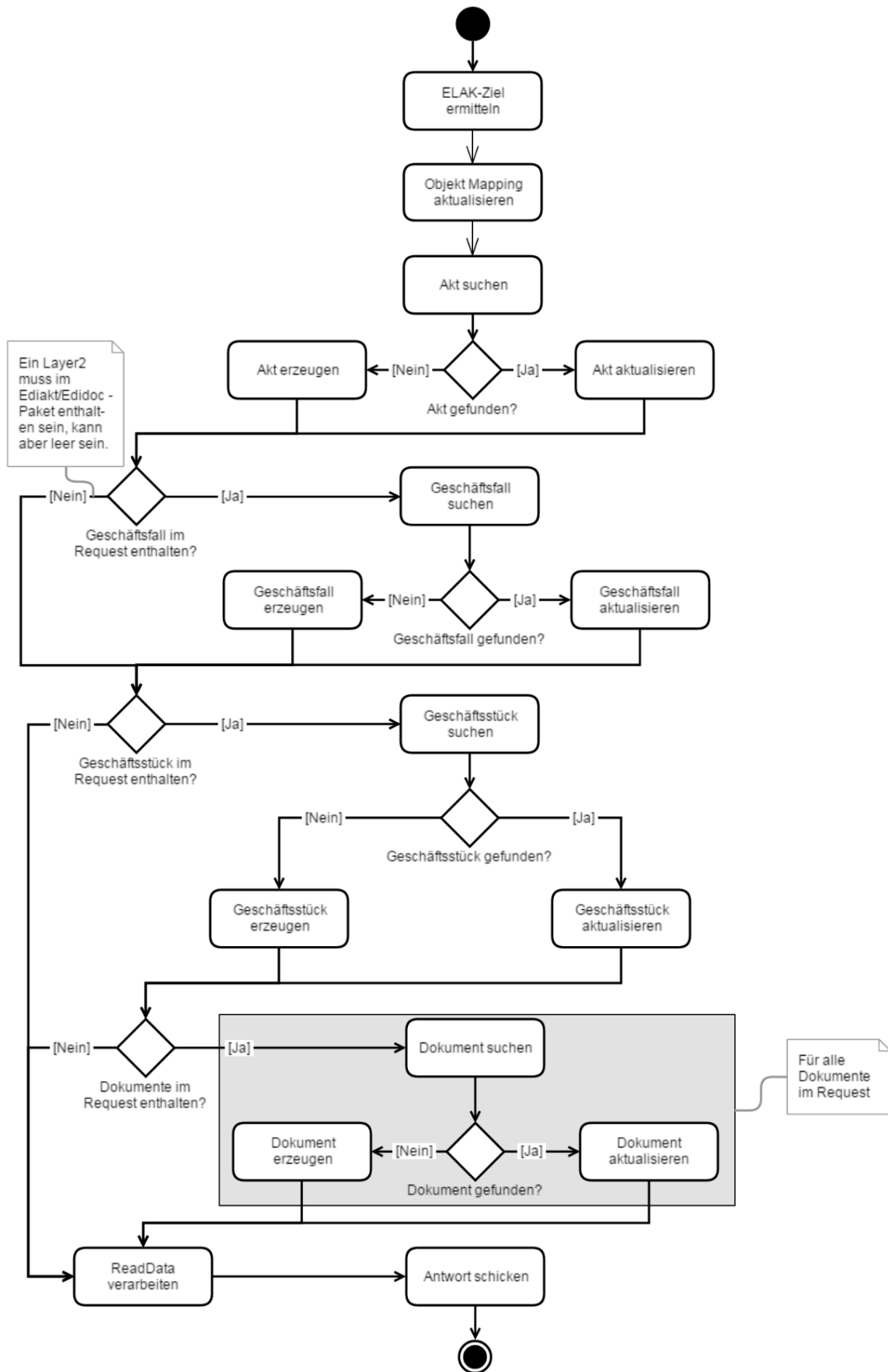
Auslöser

Der ELAK-Trans FIS-2-ELAK Server empfängt einen ELAK-Trans SendData Aufruf eines ELAK-Trans FIS-2-ELAK Clients.

Ablauf

Wird in einer der Aktivitäten ein Fehler geworfen, so müssen alle bereits vorgenommenen Änderungen zurückgenommen werden und ein Fehler an den ELAK-Trans FIS-2-ELAK Client gesendet werden!

Abbildung 9 – FIS-2-ELAK-Server SendData verarbeiten



6.4.8. UnbindData verarbeiten

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Client sendet einen UnbindData Request an den ELAK-Trans FIS-2-ELAK Server. Der ELAK-Trans FIS-2-ELAK Server sucht alle Objekte, in der LayerControl Datenstruktur die in der Objekt Mapping Tabelle verknüpft sind. Für alle gefundenen Objekte wird die Verknüpfung aus der Objekt Mapping Tabelle entfernt. Das UnbindData bezieht sich dabei immer auf den innersten definierten Layer und wird auf diesen und alle darunter liegenden Layer angewendet. Werden im LayerControl beispielsweise die FIS-IDs für Layer3 (Akt-FIS-ID), Layer2 (GF-FIS-ID) und Layer1 (GS-FIS-ID) übergeben, so soll die Verknüpfung für das spezifizierte Layer1 Objekt (Geschäftsstück) und alle darunter liegenden Layer0 Objekte (Dokumente) aufgehoben werden. Der Anwendungsfall muss implementierungsabhängig angepasst werden, um nach dem Unbind eines Objektes ggf. weitere Prozessschritte im ELAK zu starten.

Auslöser

Der ELAK-Trans FIS-2-ELAK Server empfängt einen ELAK-Trans UnbindData Aufruf eines ELAK-Trans FIS-2-ELAK Clients.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server ermittelt die Akt-FIS-ID aus LayerControl.Layer3.ReferencedIdentifier.Identification.
Wurde eine Akt-FIS-ID ermittelt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server ermittelt die GF-FIS-IDs aus LayerControl.[...].Layer2[i].ReferencedIdentifier.Identification. Werden Geschäftsfälle nicht verwendet, so kann dieser Wert auch null sein und gilt als erfolgreich ermittelte GF-FIS-ID.
Wurde eine oder mehrere GF-FIS-IDs ermittelt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server ermittelt die GS-FIS-ID aus LayerControl.[...].Layer1[j].ReferencedIdentifier.Identification.
Wurde eine oder mehrere GS-FIS-IDs ermittelt?
 - a. JA: Der ELAK-Trans FIS-2-ELAK Server ermittelt die Dok-FIS-IDs aus LayerControl.[...].Layer0[k].ReferencedIdentifier.Identification.
Wurden eine oder mehrere Dok-FIS-IDs ermittelt?
 - i. JA: Der ELAK-Trans FIS-2-ELAK Server löscht alle Einträge (Akt, Geschäftsfälle, Geschäftsstücke, Dokumente) aus der Objekt Mapping Tabelle mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2[i].ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1[j].ReferencedIdentifier.Identification
 - DOK-FIS-ID = LayerControl.[...].Layer0[k].ReferencedIdentifier.Identification
 - ii. NEIN (Keine Dok-FIS-IDs ermittelt): Der ELAK-Trans FIS-2-ELAK Server löscht alle Einträge (Akt, Geschäftsfälle,

Geschäftsstücke, Dokumente) aus der Objekt Mapping Tabelle mit:

- Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2[i].ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1[j].ReferencedIdentifier.Identification
 - DOK-FIS-ID = ANY
- b. NEIN (Keine GS-FIS-ID ermittelt): Der ELAK-Trans FIS-2-ELAK Server löscht alle Einträge (Akt, Geschäftsfälle, Geschäftsstücke, Dokumente) aus der Objekt Mapping Tabelle mit:
- Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2[i].ReferencedIdentifier.Identification
 - GS-FIS-ID = ANY
 - DOK-FIS-ID = ANY
2. NEIN (Keine GF-FIS-ID ermittelt): Der ELAK-Trans FIS-2-ELAK Server löscht alle Einträge (Akt, Geschäftsfälle, Geschäftsstücke, Dokumente) aus der Objekt Mapping Tabelle mit:
- Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = ANY
 - GS-FIS-ID = ANY
 - DOK-FIS-ID = ANY
2. NEIN (Keine Akt-FIS-ID ermittelt):
1. Der ELAK-Trans FIS-2-ELAK Server sendet einen INVALID_CONSISTANCY Fehler mit der Zusatzinformation über die fehlende Akt-FIS-ID an den ELAK-Trans FIS-2-ELAK Client.
 2. Der ELAK-Trans FIS-2-ELAK Server bricht den Anwendungsfall ab.
2. Der ELAK-Trans FIS-2-ELAK Server stößt optional abhängig von seiner Konfiguration einen Prozess auf das übergebene Objekt an.
 3. Der ELAK-Trans FIS-2-ELAK Server erstellt eine ELAK-Trans Antwort.
 4. Der ELAK-Trans FIS-2-ELAK Server sendet die ELAK-Trans Antwort an den ELAK-Trans FIS-2-ELAK Client.

6.4.9. Objekt Mapping aktualisieren

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Objekt Mapping Tabelle für die übermittelten Layer. Dabei können folgende Situationen abgedeckt werden:

1. Einen Eintrag in der Objekt Mapping Tabelle für ein im ELAK bestehendes Objekt hinzufügen.
2. Einen Eintrag in der Objekt Mapping Tabelle für ein im ELAK bestehendes Objekt aktualisieren.

Bei einer Umprotokollierung eines Geschäftsstücks von einem Akt in einen anderen Akt, wird mit diesem Anwendungsfall bereits das Mapping für den neuen Akt hinterlegt, bevor das Geschäftsstück mit dem neuen Akt verknüpft ist!

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Akt Objekt mit:
Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
Wird ein Akt gefunden?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server setzt bei Hauptpunkt 2. fort.
 2. NEIN (Der ELAK-Trans FIS-2-ELAK Server versucht ein Mapping herzustellen):
 1. Wurde eine LayerControl.Layer3.Identifier.Identification übermittelt?
 - a. JA: Der ELAK-Trans FIS-2-ELAK Server sucht im ELAK nach einem Akt mit der Aktenzahl = LayerControl.Layer3.Identifier.Identification.
Wird ein Akt gefunden?
 - i. JA:
 1. Der ELAK-Trans FIS-2-ELAK Server überprüft, ob die Akt-FIS-ID des gefundenen Aktes bereits in der Objekt Mapping Tabelle verknüpft ist.
 - a. JA: Der ELAK-Trans FIS-2-ELAK Server wirft einen INVALID_CONSISTANCY Fehler und bricht den Anwendungsfall ab.
 2. Der ELAK-Trans FIS-2-ELAK Server prüft, ob der gefundene Akt im zuständigen Mandanten für den Request liegt.
 - a. NEIN: Der ELAK-Trans FIS-2-ELAK Server wirft einen RECORD_INSUFFICIENT_PRIVILEGES Fehler und bricht den Anwendungsfall ab.
 3. Der ELAK-Trans FIS-2-ELAK Server fügt einen Eintrag für den Akt in der Objekt Mapping Tabelle mit Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification ein.
 - ii. NEIN (Akt nicht gefunden):
 1. Wird die Aktenzahl Suche strikt implementiert?
 - a. JA: Der ELAK-Trans FIS-2-ELAK Server wirft einen RECORD_NOT_FOUND Fehler und bricht den Anwendungsfall ab.
2. OPTIONAL: Wurde eine Ediakt.Payload.Layer3.MetaData.ReferencedIdentifier.SequentialNumber übermittelt?
 - a. JA: Der ELAK-Trans FIS-2-ELAK Server führt eine für das Absender FIS spezifische Suche durch.
Wird ein Akt gefunden?

-
- i. JA: Der ELAK-Trans FIS-2-ELAK Server fügt einen Eintrag für den Akt in der Objekt Mapping Tabelle mit Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification ein.
 - 2. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Geschäftsfall mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification.
Wurde ein Geschäftsfall gefunden?
 - 1. JA: Der ELAK-Trans FIS-2-ELAK Server setzt bei Hauptpunkt 3. fort.
 - 2. NEIN (Der ELAK-Trans FIS-2-ELAK Server versucht ein Mapping herzustellen):
 - 1. Der ELAK-Trans FIS-2-ELAK Server sucht mit Hilfe der sekundären Findmittel:
 - a. Der ELAK-Trans FIS-2-ELAK Server sucht nach dem Akt in der Objekt Mapping Tabelle mit Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification.
Wurde ein Akt gefunden?
 - b. JA: Der ELAK-Trans FIS-2-ELAK Server sucht im ELAK nach einem Geschäftsfall, der in dem gefundenen Akt ist und die Geschäftszahl = LayerControl.[...].Layer2.Identifier.Identification hat.
Wurde ein Geschäftsfall gefunden?
 - i. JA: Ist der Geschäftsfall bereits in der Objekt Mapping Tabelle vorhanden?
 - 1. JA: Der ELAK-Trans FIS-2-ELAK Server liefert einen Fehler an den aufrufenden Anwendungsfall.
 - 2. NEIN: Der ELAK-Trans FIS-2-ELAK Server fügt einen Eintrag für den Geschäftsfall in der Objekt Mapping Tabelle ein mit:
 - a. Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - b. GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification.
 - 3. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Geschäftsstück mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
Wurde ein Geschäftsstück gefunden?
 - 1. JA: Der ELAK-Trans FIS-2-ELAK Server beendet den Anwendungsfall.
 - 2. NEIN (Der ELAK-Trans FIS-2-ELAK Server bereitet eine Umprotokollierung vor):
Wurden die Knoten LayerControl.Layer3.ReferencedIdentifierOld bzw. LayerControl.Layer2.ReferencedIdentifierOld übermittelt?
 - 1. JA: Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach einem Geschäftsstück mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifierOld.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifierOld.Identification

- GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
Wurde ein Geschäftsstück gefunden?
 - i. JA:
 - 1. Der ELAK-Trans FIS-2-ELAK Server aktualisiert den Eintrag für das Geschäftsstück mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
 - 2. Der ELAK-Trans FIS-2-ELAK Server sucht in der Objekt Mapping Tabelle nach Dokumenten mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifierOld.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifierOld.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
 - Dok-FIS-ID = LayerControl.[...].Layer0.ReferencedIdentifier.IdentificationWurden Dokumente gefunden?
 - i. JA: Der ELAK-Trans FIS-2-ELAK Server aktualisiert alle Einträge für die Dokumente mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl.[...].Layer2.ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl.[...].Layer1.ReferencedIdentifier.Identification
 - Dok-FIS-ID = LayerControl.[...].Layer0.ReferencedIdentifier.Identification
 - ii. NEIN: Der ELAK-Trans FIS-2-ELAK Server wirft einen RECORD_NOT_FOUND Fehler und bricht den Anwendungsfall ab.

6.4.10. Akt erzeugen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server erstellt anhand des ELAK-Trans Paketes einen Akt im ELAK und registriert den Akt in der Objekt Mapping Tabelle mit dem übergebenen Wert für die Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification).

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server legt einen neuen Akt ausgehend vom übermittelten ELAK-Trans Paket unter Berücksichtigung des Layer3SendControls an.
2. Wurde ein Wert für LayerControl.Layer3.Identifier.Identification übergeben, so kann der ELAK-Trans Server diesen Wert implementierungsabhängig wie folgt interpretieren:
 1. Gewünschte Aktenzahl des FIS: Der ELAK-Trans FIS-2-ELAK Server versucht - wenn dies vom jeweiligen ELAK unterstützt wird - diese Aktenzahl als Aktenzahl des Aktes zu verwenden.
 2. Historische Aktenzahl: Der ELAK-Trans FIS-2-ELAK Server fügt die Aktenzahl als historische in ein anderes Feld des Aktes ein.
 3. Ignorieren: Der ELAK-Trans FIS-2-ELAK Server ignoriert diesen Wert.
3. Der ELAK-Trans FIS-2-ELAK Server fügt den neuen Akt mit Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification in die Objekt Mapping Tabelle ein.

6.4.11. Akt aktualisieren

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server aktualisiert einen bereits existierenden Akt im ELAK mit den Daten aus dem ELAK-Trans Paket. Dies kann im ELAK zu Statusänderungen, zum Start eines Workflows oder zur Stornierung des Aktes führen.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans FIS-2-ELAK Server hat das Akt Objekt im ELAK bereits gefunden.

Ablauf

1. Wurde der Wert für Layer3.MetaData.Canceled auf true gesetzt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server storniert den Akt.
(Abhängig vom jeweiligen ELAK können implementierungsspezifische Prüfungen vorgeschaltet werden. Manche ELAK Systeme stornieren ebenfalls mit dem Stornieren des Aktes alle darunterliegenden Ebene. Dies könnte auch zur Stornierung von nicht mit dem FIS Verknüpften Objekten führen und somit zu unerwünschten Seiteneffekten.)
 2. NEIN:
 1. Wurde ein Knoten LayerControl.Layer3SendControl übermittelt?

- a. JA:
 - i. Wurde der Knoten
LayerControl.Layer3SendControl.ContainsModifiedBasicData
gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Basisdaten
des Aktes.
 - ii. Wurde der Knoten
LayerControl.Layer3SendControl.ContainsModifiedCatchwords
gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die "Begriffe"
des Aktes.
 - iii. Wurde der Knoten
LayerControl.Layer3SendControl.ContainsModifiedLocations
gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die
Ortsbezüge des Aktes.
 - iv. Wurde der Knoten
LayerControl.Layer3SendControl.ContainsModifiedParticipants
gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die
Teilnehmer/Adressaten des Aktes
 - v. Wurde der Knoten
LayerControl.Layer3SendControl.ContainsModifiedState
gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert den Status des
Aktes.
- b. NEIN (Kein LayerControl.Layer3SendControl):
 - i. Der ELAK-Trans FIS-2-ELAK Server führt keine Aktualisierung
durch.

6.4.12. Geschäftsfall erzeugen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server erstellt anhand des ELAK-Trans Paketes einen Geschäftsfall im ELAK und registriert den Geschäftsfall in der Objekt Mapping Tabelle mit dem übergebenen Wert für die Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification) und GF-FIS-ID (LayerControl[...].Layer2.ReferencedIdentifier.Identification).

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans FIS-2-ELAK Server hat bereits einen übergeordneten Akt angelegt/gefunden.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server legt einen neuen Geschäftsfall ausgehend vom übermittelten ELAK-Trans Paket unter Berücksichtigung des Layer2SendControls an.
2. Der ELAK-Trans FIS-2-ELAK Server fügt den Geschäftsfall zum übergeordneten Akt hinzu.

3. Der ELAK-Trans FIS-2-ELAK Server startet ggf. Workflow Schritte im ELAK (abhängig von der Konfiguration der Schnittstelle und des übermittelten ELAK-Trans Pakets)
4. Der ELAK-Trans FIS-2-ELAK Server fügt den neuen Geschäftsfall in die Objekt Mapping Tabelle ein, mit:
 1. Akt-FIS-ID =LayerControl.Layer3.ReferencedIdentifier.Identification
 2. GF-FIS-ID = LayerControl[...].Layer2.ReferencedIdentifier.Identification

6.4.13. Geschäftsfall aktualisieren

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server aktualisiert einen bereits existierenden Geschäftsfall im ELAK mit den Daten aus dem ELAK-Trans Paket. Dies kann im ELAK zu Statusänderungen, Workflow-Start oder zur Stornierung des Geschäftsfalls führen.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans FIS-2-ELAK Server hat das Geschäftsfall Objekt im ELAK bereits gefunden.

Ablauf

1. Wurde der Wert für Layer2.MetaData.Canceled auf true gesetzt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server storniert den Geschäftsfall.
(Abhängig vom jeweiligen ELAK können implementierungsspezifische Prüfungen vorgeschaltet werden. Manche ELAK Systeme stornieren ebenfalls mit dem Stornieren des Geschäftsfalls alle darunterliegenden Ebene. Dies könnte auch zur Stornierung von nicht mit dem FIS Verknüpften Objekten führen und somit zu unerwünschten Seiteneffekten.)
 2. NEIN:
 1. Wurde ein Knoten LayerControl.Layer2SendControl übermittelt?
 - a. JA:
 - i. Wurde der Knoten LayerControl.Layer2SendControl.ContainsModifiedBasicData gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Basisdaten des Geschäftsfalls.
 - ii. Wurde der Knoten LayerControl.Layer2SendControl.ContainsModifiedCatchwords gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die "Begriffe" des Geschäftsfalls.
 - iii. Wurde der Knoten LayerControl.Layer2SendControl.ContainsModifiedLocations gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Ortsbezüge des Geschäftsfalls.
 - iv. Wurde der Knoten LayerControl.Layer2SendControl.ContainsModifiedParticipants gesetzt:

Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Teilnehmer/Adressaten des Geschäftsfalls

- v. Wurde der Knoten `LayerControl.Layer2SendControl.ContainsModifiedState` gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert den Status des Geschäftsfalls.

2. NEIN (Kein `LayerControl.Layer2SendControl`):

- a. Der ELAK-Trans FIS-2-ELAK Server führt keine Aktualisierung durch.

6.4.14. Geschäftsstück erzeugen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server erstellt anhand des ELAK-Trans Paketes ein Geschäftsstück im ELAK und registriert das Geschäftsstück in der Objekt Mapping Tabelle mit dem übergebenen Wert für die Akt-FIS-ID (`LayerControl.Layer3.ReferencedIdentifier.Identification`), GF-FIS-ID (`LayerControl[...].Layer2.ReferencedIdentifier.Identification`) und GS-FIS-ID (`LayerControl[...].Layer1.ReferencedIdentifier.Identification`).

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Es wurde ein Geschäftsfall im Ediakt Paket übermittelt: Der ELAK-Trans FIS-2-ELAK Server hat bereits eine übergeordneten Geschäftsfall im ELAK angelegt/gefunden.

Es wurde kein Geschäftsfall im Ediakt Paket übermittelt: Der ELAK-Trans FIS-2-ELAK Server hat bereits eine übergeordneten Akt im ELAK angelegt/gefunden.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server legt ein neues Geschäftsstück ausgehend vom übermittelten ELAK-Trans Paket unter Berücksichtigung des `Layer1SendControls` an.
2. Wurde ein Geschäftsfall im Ediakt Paket übermittelt?
 1. JA:
 1. Der ELAK-Trans FIS-2-ELAK Server fügt das Geschäftsstück zum übergeordneten Geschäftsfall hinzu.
 2. Der ELAK-Trans FIS-2-ELAK Server fügt das neue Geschäftsstück in die Objekt Mapping Tabelle ein, mit:
 - Akt-FIS-ID = `LayerControl.Layer3.ReferencedIdentifier.Identification`
 - GF-FIS-ID = `LayerControl[...].Layer2.ReferencedIdentifier.Identification`
 - GS-FIS-ID = `LayerControl[...].Layer1.ReferencedIdentifier.Identification`
 2. NEIN:
 1. Der ELAK-Trans FIS-2-ELAK Server fügt das Geschäftsstück zum übergeordneten Akt hinzu.
 2. Der ELAK-Trans FIS-2-ELAK Server fügt das neue Geschäftsstück in die Objekt Mapping Tabelle ein, mit:
 - Akt-FIS-ID = `LayerControl.Layer3.ReferencedIdentifier.Identification`
 - GF-FIS-ID = null

- GS-FIS-ID = LayerControl[...].Layer1.ReferencedIdentifier.Identification
3. Der ELAK-Trans FIS-2-ELAK Server startet ggf. Workflow Schritte im ELAK (abhängig von der Konfiguration der Schnittstelle und des übermittelten ELAK-Trans Pakets)

6.4.15. Geschäftsstück aktualisieren

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server aktualisiert ein bereits existierendes Geschäftsstück im ELAK mit den Daten aus dem ELAK-Trans Paket. Dies kann im ELAK zu Statusänderungen, Workflow-Start, Umprotokollierung oder zur Stornierung des Geschäftsstücks führen.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans FIS-2-ELAK Server hat das Geschäftsstück Objekt im ELAK bereits gefunden.

Ablauf

1. Wurde der Wert für Layer1.MetaData.Canceled auf true gesetzt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server storniert das Geschäftsstück.
 2. NEIN:
 1. Wurden LayerControl.Layer3.ReferencedIdentifierOld.Identification und LayerControl[..].Layer2.ReferencedIdentifierOld.Identification übergeben?
 - a. JA
 - i. Zuordnung des Geschäftsstücks zu übergeordneten Objekten (Akt, Geschäftsfall) aktualisieren. (Entspricht "gehört zu" Eigenschaft aktualisieren oder Umprotokollierung)
 2. Wurde ein Knoten LayerControl.Layer1SendControl übermittelt?
 - a. JA:
 - i. Wurde der Knoten LayerControl.Layer1SendControl.ContainsModifiedBasicData gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Basisdaten des Geschäftsstücks.
 - ii. Wurde der Knoten LayerControl.Layer1SendControl.ContainsModifiedCatchwords gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die "Begriffe" des Geschäftsstücks.
 - iii. Wurde der Knoten LayerControl.Layer1SendControl.ContainsModifiedLocations gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Ortsbezüge des Geschäftsstücks.
 - iv. Wurde der Knoten LayerControl.Layer1SendControl.ContainsModifiedParticipants

gesetzt:

Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Teilnehmer/Adressaten des Geschäftsstücks.

- v. Wurde der Knoten LayerControl.Layer1SendControl.ContainsModifiedState gesetzt:

Der ELAK-Trans FIS-2-ELAK Server aktualisiert den Status des Geschäftsstücks.

- b. NEIN (Kein LayerControl.Layer1SendControl):

- i. Der ELAK-Trans FIS-2-ELAK Server führt keine Aktualisierung durch.

6.4.16. Dokument erzeugen

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server erstellt anhand des ELAK-Trans Paketes ein Dokument im ELAK und registriert das Dokument in der Objekt Mapping Tabelle mit dem übergebenen Wert für die Akt-FIS-ID (LayerControl.Layer3.ReferencedIdentifier.Identification), GF-FIS-ID (LayerControl[...].Layer2.ReferencedIdentifier.Identification), GS-FIS-ID (LayerControl[...].Layer1.ReferencedIdentifier.Identification) und Dok-FIS-ID (LayerControl[...].Layer0[i].ReferencedIdentifier.Identification).

Der binäre Content des Dokuments befindet sich nicht in der Ediakt Struktur, sondern in der Documents Struktur. In der Ediakt Struktur (Layer0/Payload/BinaryDocument/EmbeddedFileURL) wird nur auf das entsprechende Dokument im Documents referenziert.

```

<Payload>
- <Layer0 id="FILE001">
+ <MetaData>
- <Payload>
- <BinaryDocument>
  <EmbeddedFileURL>FILE001</EmbeddedFileURL>
  <FileName>TestText0.txt</FileName>
  <MIMEType>text/plain</MIMEType>
</BinaryDocument>
</Payload>
</Layer0>
- <Layer0 id="FILE002">
+ <MetaData>
- <Payload>
- <BinaryDocument>
  <EmbeddedFileURL>FILE002</EmbeddedFileURL>
  <FileName>TestText1.txt</FileName>
  <MIMEType>text/plain</MIMEType>
</BinaryDocument>
</Payload>
</Layer0>
</Payload>

```

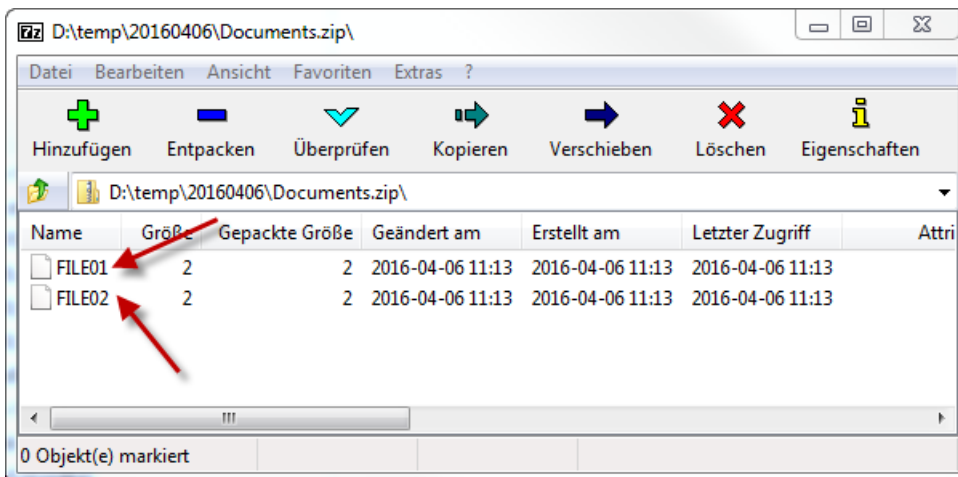
Im Documents-Knoten befindet sich die ZIP-Datei mit Dokumenten als Byte-Array (xsd:base64Binary) eingehängt.

```

<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  - <s:Header>
    <Action xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none" s:mustUnderstand="1" />
  </s:Header>
  - <s:Body xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    - <SendDataInputObject xmlns="http://reference.e-government.gv.at/namespace/elaktrans/3#">
      + <CommonInputParameter>
        <Purpose>OUTPUT</Purpose>
      + <LayerControl>
        - <Ediakt>
          <Header xmlns="http://reference.e-government.gv.at/namespace/edidoc/20130808#" />
          + <MetaData xmlns="http://reference.e-government.gv.at/namespace/edidoc/20130808#" />
          + <Payload xmlns="http://reference.e-government.gv.at/namespace/edidoc/20130808#" />
        </Ediakt>
        <Documents>QUJDKsTW3OT2/N8qMTIz</Documents>
      </SendDataInputObject>
    </s:Body>
  </s:Envelope>

```

In der ZIP-Datei liegen nun jene Dokumente, die aus Layer0/Payload/BinaryDocument/EmbeddedFileURL referenziert werden.



Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans FIS-2-ELAK Server hat bereits ein übergeordnetes Geschäftsstück angelegt/gefunden.

Ablauf

1. Der ELAK-Trans FIS-2-ELAK Server entzippt die Dateien aus dem Binary Content des Documents Knoten
2. Der ELAK-Trans FIS-2-ELAK Server ermittelt die Datei-ID aus Ediakt[...].Layer0[i].Payload.BinaryDocument.EmbeddedFileURL
3. Der ELAK-Trans FIS-2-ELAK Server sucht die Datei, deren Dateiname=Datei-ID ist.
4. Der ELAK-Trans FIS-2-ELAK Server benennt die Datei in Ediakt[...].Layer0[i].Payload.BinaryDocument.FileName um.
5. Der ELAK-Trans FIS-2-ELAK Server legt ein neues Dokument Objekt ausgehend vom übermittelten ELAK-Trans Paket unter Berücksichtigung des Layer0SendControls an.
6. Der ELAK-Trans FIS-2-ELAK Server fügt die Datei dem Dokument Objekt hinzu.

7. Der ELAK-Trans FIS-2-ELAK Server fügt das neue Dokument Objekt zum übergeordneten Geschäftsstück hinzu.
8. Der ELAK-Trans FIS-2-ELAK Server startet ggf. Workflow-Schritte im ELAK (abhängig von der Konfiguration der Schnittstelle und des übermittelten ELAK-Trans Pakets)
9. Der ELAK-Trans FIS-2-ELAK Server fügt das neue Dokument Objekt in die Objekt Mapping Tabelle ein, mit:
 - Akt-FIS-ID = LayerControl.Layer3.ReferencedIdentifier.Identification
 - GF-FIS-ID = LayerControl[...].Layer2.ReferencedIdentifier.Identification
 - GS-FIS-ID = LayerControl[...].Layer1.ReferencedIdentifier.Identification
 - Dok-FIS-ID = LayerControl[...].Layer0[i].ReferencedIdentifier.Identification

6.4.17. Dokument aktualisieren

Kurzbeschreibung

Der ELAK-Trans FIS-2-ELAK Server aktualisiert eine bereits existierendes Dokument im ELAK mit den Daten aus dem ELAK-Trans Paket. Dies kann im ELAK zu Statusänderungen, Workflow-Start oder zur Löschung des Dokuments führen.

Auslöser

Der Anwendungsfall wird vom Anwendungsfall "SendData verarbeiten" aufgerufen.

Vorbedingung

Der ELAK-Trans FIS-2-ELAK Server hat das Dokument Objekt im ELAK bereits gefunden.

Ablauf

1. Wurde der Wert für Layer0.MetaData.Canceled auf true gesetzt?
 1. JA: Der ELAK-Trans FIS-2-ELAK Server löscht/entfernt/storniert das Dokument.
 2. NEIN:
 1. Wurde ein Knoten LayerControl.Layer0SendControl übermittelt?
 - a. JA:
 - i. Wurde der Knoten LayerControl.Layer0SendControl.ContainsModifiedBasicData gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Basisdaten des Dokuments.
 - ii. Wurde der Knoten LayerControl.Layer0SendControl.ContainsModifiedBinaryContent gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Datei im Dokument Objekt analog zur Dateiverarbeitung im Anwendungsfall "Dokument erzeugen"
 - iii. Wurde der Knoten LayerControl.Layer0SendControl.ContainsModifiedCatchwords gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die "Begriffe" des Dokuments.
 - iv. Wurde der Knoten LayerControl.Layer0SendControl.ContainsModifiedLocations gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Ortsbezüge des Dokuments.

- v. Wurde der Knoten LayerControl.Layer0SendControl.ContainsModifiedParticipants gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert die Teilnehmer/Adressaten des Dokuments.
 - vi. Wurde der Knoten LayerControl.Layer0SendControl.ContainsModifiedState gesetzt:
Der ELAK-Trans FIS-2-ELAK Server aktualisiert den Status des Dokuments.
- b. NEIN (Kein LayerControl.Layer0SendControl):
- i. Der ELAK-Trans FIS-2-ELAK Server führt keine Aktualisierung durch.

7. Abgrenzung

7.1. Was ist nicht Scope von ELAK-Trans?

Folgende funktionalen Wünsche wurden an die FIS-ELAK-Integration herangetragen, wurden aber bewusst nicht in die Spezifikation aufgenommen:

- Abspeichern von Fachdaten im ELAK
Weil: Fachdaten im Regelfall im FIS verbleiben und keine Notwendigkeit besteht, diese redundant im ELAK abzuspeichern.
- Nutzung des ELAKs für Bezahlprozesse (Tabreiter "Bescheidkosten" im Fabasoft ELAK-Länderstandard)
Weil: Bezahlprozess wird nicht als Teil der ELAK-Kernfunktionalität gesehen. Bezahlprozesse werden im Regelfall über darauf spezialisierte Systeme wie SAP abgewickelt.
- Durchführen von Registerabfragen (zB ZMR)
Weil: ELAK-Trans ist eine auf Aktenstrukturen und -prozesse spezialisierte Schnittstelle. Das FIS wird als zentrales Bindeglied angesehen, von welchem die Kommunikation zu den spezialisierten Systemen ausgeht. (ELAK-Kommunikation über ELAK-Trans, ZMR-Kommunikation über die Schnittstellen des ZMR, ...)
- Austausch von bPKs bzw. vbPKs⁶
ELAK-Trans sieht keine dedizierten Datenstrukturen und Mechanismen für den Austausch von bPKs bzw. vbPKs vor. Diese können jedoch - soweit rechtlich zulässig und fachlich sinnvoll - beispielsweise als entsprechend typisierte Identification der PhysicalPerson übergeben werden.
- Backup-Recovery-Szenarien
Verteilte, ineinander integrierte Systeme bringen große Herausforderungen an Backup-Recovery-Szenarien mit sich. Ein durchgängiges Wiederherstellen eines konsistenten Datenstands ist mit Hilfe von Backup-Recovery alleine de facto nicht möglich. (Steigt zB ein zentrales FIS auf den Datenstand von vor zwei Tagen zurück, ist ein Recovery für die angebotenen ELAK-Systeme aufgrund der vielen unabhängig von diesem FIS vorgenommenen Datenbearbeitungen keine Option. Es kommt daher potenziell zu Datenleichen im ELAK.) In der Praxis wird - wenn irgendwie möglich - auf ein Recovery verzichtet und eine Vorwärts-Strategie angewendet werden, die darauf abzielt, Datenfehler bzw. Dateninkonsistenzen in einem der beteiligten Systeme durch automatisierte oder manuelle Eingriffe zu minimieren.
Weiters wird als Best Practice empfohlen, im Fall geplanter Wartungsarbeiten (zB Migration des ELAKs auf eine neue Version) den (vorgeschalteten) ELAK-Trans-Server offline zu nehmen, sowie diese Nicht-Verfügbarkeit des ELAK-Trans-Servers vorab mit den Betreibern zentraler FIS abzustimmen, damit diese etwaige Vorkehrungen treffen können. (zB Timeouts für die ELAK-Trans-Zugriffe eines Mandanten heruntersetzen) Ebenso kann angedacht werden, zentrale FIS im Stammportal des jeweiligen Partners offline zu nehmen, während die Wartungsarbeiten im angeschlossenen ELAK durchgeführt werden.
Eine umfassende Betrachtung von Backup-Recovery-Szenarien in verteilten Systemen wäre Aufgabe eines gesonderten Projekts.

⁶ Bereichsspezifische Personenkennzeichen – siehe <https://www.stammzahlenregister.gv.at/site/5970/default.aspx>

7.2. Was ist nicht Scope von ELAK-Trans 3.0 (evtl. spätere Revisionen)

Folgende funktionalen Wünsche werden an die FIS-ELAK-Integration herangetragen, wurden in die Version 3.0 aber nicht aufgenommen.

- Nutzung des Vorlagensystem des ELAKs.
 Weil: Unterschiedliche Herangehensweisen in unterschiedlichen Behörden existieren. Derzeit meist dahingehend gelöst, dass das FIS sein eigenes Vorlagensystem mitbringt und (weitgehend) fertige Dokumente im ELAK ablegt.
 - Mögliches zukünftiges Lösungsszenario könnte (im Rahmen einer eigenen "Dokumentvorlagen-System" - Spezifikation) sein:
 - FIS übergibt Vorlagenname und Fachdaten an das Dokumentvorlagen-System
 - das Dokumentvorlagen-System liefert auf Basis dessen (weitgehend) fertige DOC(X), PDF, .. an das FIS zurück
 - das FIS übergibt diese mittels sendData an den ELAK
- Verschieben von Akten(strukturen) zwischen Behörden
 Weil: Hohe Komplexität und daraus resultierend hoher Zeitbedarf für die Spezifikation und Implementierung. Lösung könnte in die Richtung gehen, dass aus dem ELAK der Quell-Behörde der Akt in eine PDF-Datei exportiert wird und diese an die Ziel-Behörde übergeben wird. EDIAKT II / EDIDOC - Pakete als Alternative können zwar aus den meisten ELAK-Systemen exportiert werden, der Import ist aber nur in den seltensten Fällen möglich.
 Für manche Szenarien könnte auf Basis von ELAK-Trans 3 das Auslesen der Aktenstruktur in der Quell-Behörde durch das FIS mittels "readData" und das anschließende Neuanlegen der Aktenstruktur durch das FIS in der Ziel-Behörde mittels "sendData" einen Lösungsansatz darstellen. (Aus Laufzeit-Überlegungen - insbesondere bei größeren Akten - sollte bei "readData" von einem Auslesen des gesamten Aktes mit einem "readData"-Aufruf Abstand genommen werden. Stattdessen könnte zB jeweils ein Geschäftsstück incl. dazugehöriger Dokumente mit "readData" gelesen werden. Bei "sendData" ist für L3 - L1 ohnehin jeweils nur ein Objekt zulässig, wodurch ein Geschäftsstück incl. dazugehöriger Dokumente jeweils mit einem eigenen "sendData"-Aufruf erzeugt werden muss.)
 - Behandlung der Reorganisation von FIS-ID
 Im Normalfall sollten FIS-IDs so gewählt werden, dass diese dauerhaft eindeutig und unveränderbar sind. (zb GUID) Sollte trotzdem eine Änderung der FIS-IDs notwendig werden, ist diese mittels Austausch des Schlüssel-Mappings im CSV-Format zu lösen. (Dies kann nicht mit sendData abgedeckt werden, da hier keine klare Unterscheidung zwischen Umprotokollierung und Schlüsselupdate möglich ist.)
 - Behandlung der Reorganisationen von FIS-verknüpften Aktenstruktur im ELAK
 Dies sollte keine Auswirkungen auf das FIS haben, da das FIS keine technischen Identifier des ELAKs speichert. Die Schlüsseländerung ist innerhalb der jeweiligen Behörde zwischen ELAK und ELAK-Trans-Server zu koordinieren.
 - Suche
 ELAK-Trans 3.0 beschränkt sich bewusst auf den Austausch von Aktenstrukturen und direkt damit zusammenhängenden Informationen wie Adressaten und Ortsbezüge. ELAK-Trans sieht aber keine generischen Suchen nach beliebigen ELAK-Objekten vor. Es ist somit nicht möglich Benutzerobjekte, Prozessdefinitionen, Vorlagen oder Geschäftsstücktypen auszulesen um nur einige wenige beispielhaft aufzuzählen. Diese Entscheidung wurde im Sinne der Entkopplung zentraler FIS von dezentralen ELAK's unterschiedlicher Hersteller, unterschiedlicher ELAK-Datenmodelle, ELAK-Architekturen und organisationsspezifischer Prozesse bzw. Einsatzszenarien gewählt.

Würde irgendeine Form der herstellerunabhängigen Suche nach Objekten im ELAK angeboten werden und diese Suchtreffer/Objekte in einer herstellerunabhängigen Form zurückgeben wäre die Folgefrage, was mit diesen Suchtreffer/Objekten im FIS weiter passiert und welche weiteren zusätzlichen Anwendungsfälle definiert werden müssten.

7.3. Sonstiges

- ELAK-Trans steht in keiner direkten "Konkurrenz" zur Dualen Zuse.
Erfolgt der Versand von Geschäftsstücken/Dokumenten über den ELAK kann sich dieser für die Zustellung der Dualen Zuse bedienen. Die Rückmeldung über den Zustellstatus erfolgt in diesem Fall von der Dualen Zuse an den ELAK, von dort an ELAK-Trans und schließlich an das FIS.
Erfolgt der Versand von Geschäftsstücken/Dokumenten über das FIS kann sich dieses direkt der Dualen Zuse bedienen. In diesem Fall erfolgt die Rückmeldung über den Zustellstatus von der Dualen Zuse direkt an das FIS.
- "ELAK-Trans FIS-2-ELAK Servers" - Referenzimplementierung
Im Feedback-Prozess wurde vorgeschlagen eine Referenzimplementierung eines "ELAK-Trans FIS-2-ELAK Servers" zur Verfügung zu stellen, gegen welche FIS ihre ELAK-Anbindung testen können und welches in Fällen unterschiedlicher Interpretation des Standards die korrekte Implementierung vorgibt.
Grobe Skizze des Umfangs: Diese Referenzimplementierung sollte ohne dahinterliegendes ELAK-System auskommen. Es sollte "FIS-2-ELAK" - Requests entgegennehmen und die Verarbeitungsschritte durchführen, ohne aber auf ein ELAK-System zuzugreifen. Stattdessen sollen die Daten in einer eigenen Datenbank persistiert werden um über mehrere Requests hinweg ein konsistentes Verhalten zu gewährleisten. Weiters sollte eine solche Referenzimplementierung auch eine Weboberfläche umfassen, von welcher aus testweise ELAK-2-FIS-Requests abgesetzt werden können.
Eine derartige Referenzimplementierung ist aus Ressourcengründen Stand März 2016 nicht geplant, wird als Idee aber im Hinterkopf behalten.
- "ELAK-Trans FIS-2-ELAK Client" - Bibliotheken
Im Feedback-Prozess wurde weiters vorgeschlagen eine "ELAK-Trans FIS-2-ELAK Client" - Bibliothek als OpenSource zur Verfügung zu stellen, welche von FIS-Anbietern integriert werden kann. Eine derartige Bibliothek müsste für unterschiedliche Plattformen angeboten werden (zB Java und .NET). Weiters würde eine derartige Bibliothek Abhängigkeiten zu weiteren Frameworks wie Webservice-Runtimes aufweisen.
Derartige Bibliotheken sind Stand März 2016 nicht geplant, werden als Idee aber im Hinterkopf behalten. Es obliegt jedem Partner im ELAK-Trans-Umfeld, eine solche Bibliothek zu implementieren und als Open Source zur Verfügung zu stellen.

8. Änderungsbeschreibung von ELAKTrans 2.0 zu ELAKTrans 3.0

8.1. Methodenzusammenführung

insertELAK, insertApplication, updateELAK, updateApplication, updateApplicationState, updateELAKState, sendELAK wurde zu sendData.

readELAK wird zu readData. Es können mehrere Geschäftsfälle, Geschäftsstücke eines Aktes auf einmal gelesen werden. Mehr Einstellungsmöglichkeiten.

Entfall von updateApplicationKeys, updateELAKKeys. Umprotokollierung implizit durch sendData.

Entfall von deleteELAK (Im ELAK kann sowieso nicht gelöscht werden). Stattdessen gibt es ein unbindData sowie das Stornieren im Rahmen von sendData.

Entfall von deleteApplication. Bei Umprotokollierung im ELAK wurde bisher deleteApplication und insertApplication aufgerufen. Wird in Zukunft durch sendData abgedeckt.

8.2. Implizite (Um-)Protokollierung

Werden Geschäftsstücke die noch keinem Akt zugeordnet sind an das FIS übertragen, gibt dieses vorerst nur 00000000-0000-0000-0000-000000000000 für die Layer3ID und Layer2ID zurück. D.h. die Layer1ID muss eindeutig sein.

Nach der Zuordnung im FIS zum richtigen Objekt (Akt) werden die Daten mit sendData an den ELAK übergeben.

Der ELAK-Trans-Server erkennt die geänderte Zuordnung und nimmt implizit eine (Um-)Protokollierung vor.

8.3. Objektidentifizierung

Alle Objekte sollen mit einer eindeutigen ID versehen werden.

Das ist in erster Linie für Participants (Adressaten) erforderlich, da einzelne ELAKs dafür Stammobjekte aufbauen und diese identifizieren müssen.

8.4. Geänderte Elemente

8.4.1. Basiselement EDIAKT wird zu EDIDOC

Neue Version des EDIAKT-Schemas. Führt zu einer anderen Art der Dokumenteinbindung, welche im Anwendungsfall "6.4.16 Dokument erzeugen" beschrieben ist. Zusätzlich wurden alle nicht benötigten Attribute entfernt.

Beim HeaderType wurden Receiver und Sender entfernt. EdiAkt-Schema von ELAKTrans kann damit nicht mehr zum Exportieren von Dateien, die später mit dem Ediakt-Viewer geöffnet werden, verwendet werden.

Bei mehreren Dokumenten ist eine Kennzeichnung, welches das Hauptdokument ist, notwendig.

Bei den Metadaten werden Statusflags (closed, canceled, approved) übergeben.

Es werden Participants und Locations statt Receiver angegeben.

Bei Participants können TypeOfParticipation, DispatchType und ReportingType angegeben werden.

Bei Locations (Speicherung als Adresse im ELAK) können TypeOfLocation und Description angegeben werden.

8.4.2. Basiselement PersonData

Statt der deutschen Version wird die englische Version verwendet. Zusätzlich wurden alle nicht benötigten Attribute entfernt.

Neu wird ein Knoten AdditionalData mit Bemerkung und Zustelldatum verwendet.

Für Zusatzinformationen bei der Adresse (z.B. c/o) wird statt der AddressLine der AdditionalText verwendet.

8.4.3. CommonParameter

CommonELAKInputParameter und CommonApplicationInputParameter werden zu CommonInputParameter.

Die CommonInputParameter beinhalten einen RequestIdentifier.

CommonELAKOutputParameter und CommonApplicationOutputParameter werden zu CommonOutputParameter.

Die CommonOutputParameter beinhalten einen RequestIdentifier.

8.4.4. LayerUID

Wurde durch LayerControl abgelöst und vereinfacht. Bisher wurde z.B. die ID von Layer3 mehrfach (bei Layer3ID, bei Layer2ID, bei Layer1ID, bei Layer0ID) übergeben. Jetzt wird jede ID nur mehr einmal übergeben. Zusätzlich gibt es jedoch die alten LayerIDs (z.B. für Umprotokollierung) und Optimierungshinweise pro Layer.

9. Glossar

Begriff	Erläuterung
ELAK	siehe Elektronischer Akt
Elektronischer Akt	Elektronisches Aktenführungssystem
Fach-informations-system	Anwendung, die für die Verarbeitung und Verwaltung von Fachdaten zuständig ist
FIS	siehe Fachinformationssystem
Kopplung	Verknüpfung zwischen Aktenstruktur und Fachentität
EDIDOC	XML-Schema für die allgemeingültige Abbildung und den Austausch von Aktenstrukturen. Das Wurzelement des EDIDOC-Schema ist der EdiaktType
Ediakt-Struktur, Ediakt-Paket	Gemeint ist damit der EdiaktType als Wurzelement des EDIDOC-Schema. Dadurch, dass EDIAKT der Vorgänger von EDIDOC war, im EDIDOC-Schema aber weiterhin der EdiaktType verwendet wird, lassen sich Namensverwirrungen nur schwer vermeiden
Ebene 3	des EDIDOC-Pakets; auch L3 oder Layer 3; meist sinngemäß für "Akt"
Ebene 2	des EDIDOC-Pakets; auch L2 oder Layer 2; meist sinngemäß für "Geschäftsfall"
Ebene 1	des EDIDOC-Pakets; auch L1 oder Layer 1; meist sinngemäß für "Geschäftsstück"
Ebene 0	des EDIDOC-Pakets; auch L0 oder Layer 0; meist sinngemäß für "Dokument"
ETS	ET-Server oder Elak-Trans-Server; im Regelfall vor dem ELAK-System vorgeschaltet

FIS-2-ELAK	FIS zu ELAK
ELAK-2-FIS	ELAK zu FIS
Objekt Mapping Tabelle	Zuordnungstabelle; Mechanismus mit dem der ELAK-Trans-Server die Zuordnung zwischen FIS-ID's und technischen ELAK-Identifizieren persistiert (Alternative: Speichern der FIS-ID's direkt im ELAK)
Kanzlist	auch "Protokollkraft"
Akt-FIS-ID	ID des Aktes im FIS. Im ELAK-Trans Paket wird diese ID redundant in den folgenden Knoten angegeben: <ul style="list-style-type: none"> • LayerControl.Layer3.ReferencedIdentifier.Identification • Ediakt.Layer3.MetaData.ReferencedIdentifier.Identification
GF-FIS-ID	ID des Geschäftsfalls im FIS. Im ELAK-Trans Paket wird diese ID redundant in den folgenden Knoten angegeben: <ul style="list-style-type: none"> • LayerControl.Layer3.Layer2.ReferencedIdentifier.Identification • Ediakt.Layer3.Payload.Layer2.ReferencedIdentifier.Identification
GS-FIS-ID	ID des Geschäftsstücks im FIS. Im ELAK-Trans Paket wird diese ID redundant in den folgenden Knoten angegeben: <ul style="list-style-type: none"> • LayerControl.Layer3.Layer2.Layer1.ReferencedIdentifier.Identification • Ediakt.Layer3.Payload.Layer2.Payload.Layer1.ReferencedIdentifier.Identification
Dok-FIS-ID	ID des Dokuments im FIS. Im ELAK-Trans Paket wird diese ID redundant in den folgenden Knoten angegeben: <ul style="list-style-type: none"> • LayerControl.Layer3.Layer2.Layer1.Layer0.ReferencedIdentifier.Identification • Ediakt.Layer3.Payload.Layer2.Payload.Layer1.Payload.Layer0.ReferencedIdentifier.Identification
Mandant	Abgrenzung innerhalb eines ELAK Systems; auch Sachgebiet und Protokoll genannt
ELAK-Trans-Server	auch (ELAK-) "Middleware", Summe der Komponenten "ELAK-Trans FIS-2-ELAK Server" und "ELAK-Trans ELAK-2-FIS Client"
bPK	bereichsspezifisches Personenkennzeichen

vbPK	verschlüsseltes bereichsspezifisches Personenkennzeichen
ZMR	Zentrales Melderegister

A. Anhang - Abbildungsverzeichnis

Abbildung 1 - Verteilte ELAK-Trans Architektur mit PVP	9
Abbildung 2 - Batch-Betrieb.....	11
Abbildung 3 – EDIDOC-Paket	14
Abbildung 4 – Webservice-Methoden	60
Abbildung 5 – Anwendungsfall-Diagramm ELAK-2-FIS-Client.....	68
Abbildung 6 – Anwendungsfall-Diagramm ELAK-2-FIS-Server.....	72
Abbildung 7 – Anwendungsfall-Diagramm FIS-2-ELAK-Client.....	75
Abbildung 8 – Anwendungsfall-Diagramm FIS-2-ELAK-Server.....	77
Abbildung 9 – FIS-2-ELAK-Server SendData verarbeiten.....	83

B. Anhang - Tabellenverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

C. Anhang - Änderungsprotokoll

Version: 3.0.0 zu 3.0.1
Datum: 8.8.2016
Autor: Christoph Straßer

Änderungen:

- Sending-Date von MetaDataType in PersonExtensionType verschoben. (Weil Fabasoft-ELAK's das Versanddatum pro Empfänger eines Geschäftsstücks getrennt abspeichern und diese potenziell voneinander abweichen könnten.)
- Erweiterte Erläuterung bei MainDocument des Layer1Type.
- Usecase-Diagramm ELAK-Trans ELAK-2-FIS Client mit Hinblick auf bessere Lesbarkeit umformatiert.
- Ein oder zwei Stellen bei denen die Einrückung/Aufzählungszeichen nicht korrekt gewählt war korrigiert.