



*der Kommunikationsstandard für digitale Prozesskette*

## **eXTra Best Practices**

**Version 1.5**

**Ausgabestand 1.5.0**

**Herausgeber:**

AWV – Arbeitsgemeinschaft für wirtschaftliche Verwaltung e. V.  
Düsseldorfer Str. 40  
65760 Eschborn  
Vereinsregister 73 VR 5158, Amtsgericht Frankfurt am Main  
Telefon: 0 61 96/7 77 26-0  
Fax: 0 61 96/7 77 26-51  
Mail: [info@awv-net.de](mailto:info@awv-net.de)  
Web: [www.extra-standard.de](http://www.extra-standard.de), [www.awv-net.de](http://www.awv-net.de).

Das vorliegende Dokument „eXTra Best Practices“ des eXTra Kommunikationsstandards wurde von Mitarbeiterinnen und Mitarbeitern des AWV-Arbeitskreises 2.1 „Vereinheitlichung von Datenübermittlungssystemen“ im Fachausschuss 2 „Verwaltungsvereinfachung und Entbürokratisierung im personalwirtschaftlichen Umfeld“ entwickelt.

Eine Weitergabe des Dokuments an Dritte darf nur unentgeltlich und in unveränderter Form erfolgen.

## Änderungsprotokoll für Version 1.5

<b>Autor[en]</b>	<b>Datum</b>	<b>Beschreibung</b>
[gelöscht]	25.05.2018	Geänderte Benennung: eXTra der Kommunikationsstandard für digitale Prozessketten Basis ist eXTra Version V1.5 mit dem neuen PlugIn BusinessProcess und der neuen Standardnachricht RepeatResponseRequest, jedoch (noch) ohne Berücksichtigung von DocumentSets

## Erstausgabe Version 1.4

<b>Autor[en]</b>	<b>Datum</b>	<b>Beschreibung</b>
[gelöscht]	09.02.2014	Erstausgabe Version 1.4

---

## Inhalt

<b>0. Hinweise zu diesem Dokument.....</b>	<b>5</b>
0.1. Notation.....	5
0.2. Zielgruppen.....	5
<b>1. Einführung.....</b>	<b>6</b>
<b>2. Rahmenbedingungen eines verbundspezifischen eXtra-Standards.....</b>	<b>8</b>
2.1. Die vorgegebene Topologie des Datenübermittlungsverbundes.....	8
2.2. Betriebsmodelle und deren Unterstützung mit dem eXtra-Standard.....	10
2.2.1. Betriebsmodelle mit asynchron verarbeitenden Fachverfahren.....	13
2.2.1.1. Das Betriebsmodell Sendebetrieb.....	13
2.2.1.2. Das Betriebsmodell Holbetrieb.....	14
2.2.1.3. Das Betriebsmodell Sende- Holbetrieb.....	16
2.2.1.4. Das Betriebsmodell Hol- Bestätigungsbetriebs und Hol- Bestätigungs- Sendebetrieb.....	17
2.2.1.5. Das Betriebsmodell Sende-Hol Bestätigungs- und Hol-Bestätigungs-Sendebetrieb.....	19
2.2.2. Betriebsmodell Dialogbetrieb mit synchron verarbeitenden Fachverfahren.....	21
<b>3. Gestaltung eines verbundspezifischen eXtra-Standards.....</b>	<b>22</b>
3.1. Gestaltung der Schemadateien eines verbundspezifischen eXtra-Standards.....	22
3.1.1. Profilierung und Verwendung der Schemadateien.....	22
3.1.2. Prozessübergreifende oder prozessspezifische Schemadateien.....	22
3.1.3. Schemadateien für eXtra-Request und eXtra-Response.....	23
3.1.4. Relation der Schemadateien des eXtra-Basisstandards gegenüber denjenigen eines verbundspezifischen eXtra-Standards.....	24
3.1.5. Identifikation des Profils.....	25
3.2. Gestaltung der Ebenen eines verbundspezifischen eXtra-Standards.....	25
3.2.1. Sendeprozess.....	25
3.2.2. Holprozess.....	27
3.2.3. Bestätigungsprozess.....	28
3.3. Gestaltung der Header eines verbundspezifischen eXtra-Standards.....	28
3.4. Sicherheit, Datenschutz.....	34
<b>4. Der laufende Betrieb – dynamische Aspekte.....</b>	<b>35</b>
4.1. Verfügbarkeit des eXtra Systems auf Empfängerseite, die Nachricht ExtraError.....	35
4.2. Zusammenspiel der eXtra Senderseite mit der eXtra Empfängerseite.....	35
4.2.1. Der Sendeprozess.....	36
4.2.2. Der Holprozess.....	49
4.2.3. Der Bestätigungsprozess.....	57
4.2.4. Nachvollzug und Recherche.....	59
4.2.4.1. Die Auskunftsfunktionen StatusRequest und StatusResponse sowie ListRequest und ListResponse.....	61
4.3. Zusammenspiel der beiden Fachverfahren auf Sender- und Empfängerseite.....	63
<b>5. Das eXtra-Gesamtsystem: Verbindliche Festlegungen, Betriebsparameter und Leistungsumfang.....</b>	<b>65</b>
<b>6. Anhang.....</b>	<b>68</b>
6.1. Referenzen.....	68
6.2. Glossar.....	70

---

## 0. Hinweise zu diesem Dokument

### 0.1. Notation

Verweise auf Stellen innerhalb dieses Dokumentes referenzieren die Absatznummer und schließen sie in runde Klammern ein (z.B. „(4)“); Verweise auf externe Dokumente haben die Form eines Kurznamens aus Großbuchstaben und stehen in eckigen Klammern, z.B. [XSD]. Eine Übersicht der Referenzen befindet sich im Anhang (6.1).

Ist ein Begriff im Glossar (6.2) erläutert, so wird er bei der ersten Verwendung im Dokument mit einem vorangestellten „①“ versehen, z.B. „① Plug-In“.

Hervorhebungen sind *kursiv* gesetzt.

Die eXtra-Standardnachrichten sind **fett** markiert.

XML-Elemente und Attribute sind in `Courier New` gesetzt.

### 0.2. Zielgruppen

Das vorliegende Dokument wendet sich vornehmlich an Projektleiter, Softwarearchitekten und Softwareentwickler, die mit den Grundlagen des eXtra-Standards bereits vertraut sind (insbesondere dem „Kompendium“ [KOMP] sowie den „Design Guidelines“ [DSIG]) und eine Implementierung vorbereiten oder durchführen.

Weitere relevante Dokumente für die Zielgruppe Projektleiter und Softwarearchitekten sind die „Profilierung“ [PROF], die „Versionierung“ [VERS], die „Sicherheit und Verfügbarkeit“ [EXSEC] und gegebenenfalls „eXtra und WebServices“ [EXWS].

Für die Zielgruppe Softwarearchitekten und Softwareentwickler stehen darüber hinaus der „Überblick über die eXtra-Standardnachrichten“ [ÜMSG] sowie die „Schnittstellenbeschreibung für den eXtra-Basisstandard“ [IFACE] und die „Schnittstellenbeschreibung der eXtra-Standardnachrichten“ [EMSG] zur Verfügung.

## 1. Einführung

Zentrales Anliegen des eXTra Kommunikationsstandards mit integrierter Logistik und Datenübermittlung ist es, die Aufwände für seine Anwendung bei der Migration, der Konsolidierung oder Neuentwicklung von Verfahren gering zu halten. Zu diesem Zweck ist der eXTra Kommunikationsstandard zum einen modular aufgebaut und bietet zum andern mit Hilfe des Alleinstellungsmerkmals von eXTra - der ① Profilierung - weitgehende Gestaltungs- und Wahlmöglichkeiten für eine bedarfsorientierte Maßschneidung des zu konstruierenden ① verbundspezifischen eXTra-Standards.

Die Sprachmittel und Funktionen des eXTra Kommunikationsstandards sind auf folgende Module verteilt:

- den eXTra Basisstandard Transport mit den allgemein notwendigen Kernfunktionen für die Kommunikation und den Datenaustausch,
- den optionalen ① PlugIns, die weitere, nicht allgemein notwendige Funktionen implementieren und
- den optionalen Standardnachrichten, die Sprachmittel für typische Prozessabläufe und Prozessketten zur Verfügung stellen.

Jeder dieser Bestandteile ist in Form von XML Schemata implementiert. Die Gestaltung eines verbundspezifischen eXTra-Standards erfolgt im Wege der Profilierung einerseits des eXTra Basisstandards Transport, also dessen Schemadateien, im Sinne einer Maßschneidung. Andererseits bewirkt die Profilierung eine Anreicherung des Basisstandards um die Auswahl und Profilierung der optionalen PlugIns und der optionalen eXTra Standardnachrichten. Die Profilierung ist immer restriktiv, führt also stets zu Einschränkungen in den jeweiligen Schemadateien gegenüber dem unprofilieren Zustand. Erweiterungen in den Schemadateien sind nicht zulässig. Zu den Regeln und Restriktionen bei der Profilierung siehe [PROF].

eXTra-Standardnachrichten sind per Definition fachliche Nachrichten und somit nicht Teil des eXTra-Kommunikationsprotokolls. Sie werden wie Nutzdaten behandelt und wie diese zwischen eXTra-Anwendungsinstanzen ausgetauscht. Ihre Anwendung ist vollständig optional, aber ihre Profilierung unterliegt den in [PROF] definierten Regeln. Sie implementieren Funktionen, die bei der Kommunikation mit ① Fachverfahren in typischen Prozessabläufen und Prozessketten

benötigt werden, wie etwa Statusanfragen und das Abholen bzw. Anfordern von Verarbeitungsergebnissen.

Bei der Analyse bestehender und neuer ① Datenübermittlungsverfahren und deren Anforderungen haben sich insbesondere bei asynchronen Fachverfahren mehrere typische Prozessabläufe herauskristallisiert, die mittels eXTra-Standardnachrichten durchgeführt werden können. eXTra unterstützt dabei insbesondere:

- Prozessketten, die aus mehreren Prozessen auf Sender- wie Empfängerseite bestehen. Dies gilt insbesondere für Prozessketten, in die ein asynchron verarbeitendes Fachverfahren auf Empfängerseite involviert ist und bei denen der Sender sehr an den Verarbeitungsergebnissen seiner Sendungen interessiert ist.
- den automatischen, bedienerlosen Betrieb auf Sender- wie Empfängerseite.
- die Einholung von Auskünften über den Bearbeitungsstatus von Sendungen und über den Werdegang von Vorgängen auf Empfängerseite sowie den Nachvollzug des Datenaustausches.

In diesem Dokument wird davon ausgegangen, dass im Rahmen der Konzeption eines verbundspezifischen eXTra Kommunikationsstandards bereits grundlegende Entscheidungen getroffen wurden, z.B. welche Topologie beim Datenübermittlungsverbund gegeben ist und damit zusammenhängend welche Rolle das oder die eXTra-Empfangssysteme einnehmen sollen. Ebenso wird angenommen, dass bereits geklärt ist, welches Betriebsmodell zum Einsatz gelangen soll. Diese Entscheidungen werden also nicht diskutiert, sondern deren Auswirkungen behandelt.

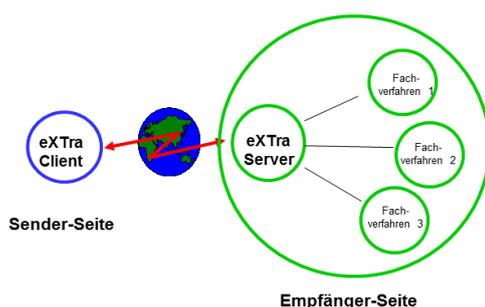
Ziel dieses Dokuments ist es, in Form einer Themensammlung Hinweise und Empfehlungen von Praktikern für Praktiker zu geben, um Hilfestellung bei der Ausgestaltung und Implementierung eines verbundspezifischen eXTra Kommunikationsstandards zu leisten. Dies geht von der Einordnung der verschiedenen Betriebsmodelle und deren Unterstützung durch den eXTra- Kommunikationsstandard über die Profilierung bis hin zur Verwendung der einzelnen eXTra Sprachmittel.

## 2. Rahmenbedingungen eines verbundspezifischen eXtra-Standards

### 2.1. Die vorgegebene Topologie des Datenübermittlungsverbundes

Die Topologie des Datenübermittlungsverbundes hat maßgeblichen Einfluss auf die Profilierung. Maßgeblich sind vor allem folgende Fragen:

- Befindet sich der eXtra-Server des Empfangssystems und die daran angegliederten Fachverfahren an einem Ort oder an unterschiedlichen Orten bzw. sollen die angegliederten Fachverfahren mit eXtra-Sprachmitteln gezielt adressiert werden können? Gleichbedeutend damit ist die Frage, ob die Vertraulichkeit der fachlichen Daten eines Fachverfahrens bis zum jeweiligen Fachverfahren gewährleistet sein soll oder ob dies nur bis zum eXtra-Server des Empfangssystems erforderlich ist.
- Kann pro Sendeprozess immer nur ein einziges oder potentiell mehrere Fachverfahren bedient werden?
- Kann der eXtra-Server des Empfangssystems und die angeschlossenen Fachverfahren immer nur ein einziges oder auch mehrere Verarbeitungsobjekte entgegennehmen? Beispiel: Will ein Fachverfahren immer nur die Daten von genau einem oder von mehreren Unternehmen entgegennehmen und verarbeiten können? Analog in der Auswirkung ist die Frage, ob das Fachverfahren immer nur pro Lieferung die fachlichen Daten eines einzigen oder mehrerer Erzeuger verarbeiten kann.

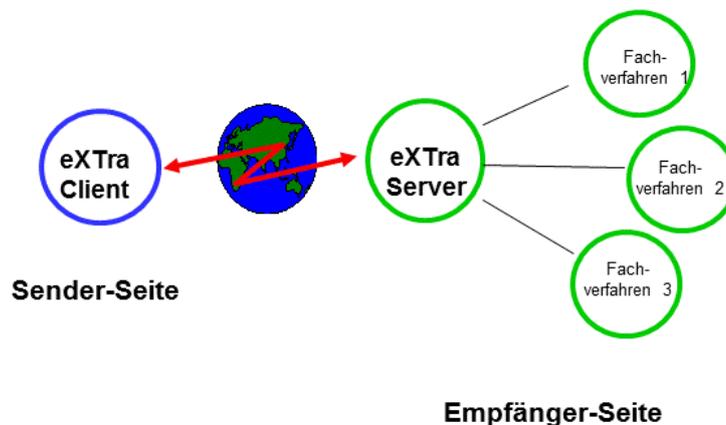


**Abbildung 1: Topologie 1**

Topologie 1: Das eXtra-Empfangssystem (technisch gesehen der eXtra-Server) und die angeschlossenen Fachverfahren befinden sich an einem Ort (bzw. die Vertraulichkeit muss „nur“ bis zum eXtra-Empfangssystem – dem eXtra-Server - gewährleistet sein).

Sofern sich pro Sendeprozess in einer Lieferung immer nur die fachlichen Daten für genau ein Fachverfahren befinden – egal; ob sich die Fachverfahren und das eXtra-Empfangssystem an einem Ort oder an mehreren Orten befinden und unabhängig davon, ob die fachlichen Daten für das jeweilige Fachverfahren oder das eXtra-Empfangssystem verschlüsselt wurden – genügt eine einzige eXtra-Ebene, nämlich die Transportebene.

Beispiel hierfür sind die Sofortmeldungen an die Rentenversicherung.



**Abbildung 2: Topologie 2**

Topologie 2: Das eXtra-Empfangssystem (technisch gesehen der eXtra-Server) und die angeschlossenen Fachverfahren befinden sich an unterschiedlichen Orten (möglicherweise jedes Fachverfahren an einem anderen Ort) bzw. die Vertraulichkeit muss bis zum jeweiligen Fachverfahren gewährleistet sein.

Sofern pro Sendeprozess in einer Lieferung die fachlichen Daten für mehrere Fachverfahren enthalten sein können – egal; ob sich die Fachverfahren und das eXtra-Empfangssystem an einem Ort oder an mehreren Orten befinden und unabhängig davon, ob die fachlichen Daten für das jeweilige Fachverfahren oder das eXtra-Empfangssystem verschlüsselt wurden – sind mindestens zwei eXtra-Ebenen erforderlich.

Beispiel hierfür sind alle Meldungen an den GKV-Kommunikationsserver mit zwei eXtra-Ebenen, der Transport- und der Paketebene.

Sofern pro Sendeprozess in einer Lieferung die fachlichen Daten nur für ein einziges Fachverfahren enthalten sein dürfen, das Fachverfahren zwar die Verarbeitung der Daten

mehrerer Unternehmen akzeptiert, jedoch die explizite Kennzeichnung der fachlichen Daten pro Unternehmen fordert – egal, ob sich die Fachverfahren und das eXTra-Empfangssystem an einem oder mehreren Orten befinden – sind mindestens zwei eXTra-Ebenen erforderlich: Die Transport- und die Message-Ebene .

Beispiel hierfür sind die Meldungen für die elektronische Betriebsprüfung an die Datenstelle der Träger der Rentenversicherung DSRV mit zwei eXTra-Ebenen, der Transport- und der Messageebene.

Drei eXTra-Ebenen wären sinnvoll, wenn in einer Lieferung mehrere Fachverfahren bedient werden sollen, wobei die Vertraulichkeit bis zum jeweiligen Fachverfahren gewährleistet sein muss (die erforderliche Unterteilung leistet die Package-Ebene) und zumindest ein Fachverfahren zwar die Belieferung von Daten mehrerer Unternehmen akzeptiert, aber eine unternehmensspezifische Verarbeitung verlangt (die erforderliche Unterteilung leistet die Message-Ebene).

## 2.2. Betriebsmodelle und deren Unterstützung mit dem eXTra-Standard

Im Folgenden werden mehrere Betriebsmodelle und deren charakteristische Eigenschaften vorgestellt sowie die jeweils in Frage kommenden Sprachmittel des eXTra Kommunikationsstandards aufgezählt, wobei entsprechende Hinweise und Empfehlungen gegeben werden.

Grob unterteilt gibt es Betriebsmodelle mit asynchron verarbeitenden/konsumierenden Fachverfahren auf Empfänger- und/oder Senderseite (dazu unten im Punkt 2.2.1) und das Betriebsmodell Dialogverfahren mit synchron verarbeitenden/konsumierenden Fachverfahren auf Empfängerseite (dazu unten im Punkt 2.2.2).

In einem Betriebsmodell auf Basis des eXTra Kommunikationsstandards gibt es 1 bis n verschiedene Prozesse, die je nach Betriebsmodell in einer bestimmten Sequenz ablaufen. Im eXTra Kommunikationsstandard gibt es folgende Prozesse:

- Einen Sendeprozess (mit fachlichen Daten),
- einen Holprozess (mit der Standardnachricht **DataRequest**),
- einen Bestätigungsprozess (mit der Standardnachricht **ConfirmationOfRequest**) und

- Auskunftsprozesse (mit den Standardnachrichten **ListRequest/ListResponse** oder **StatusRequest/StatusResponse**).

Ein Auskunftsprozess kann bei allen Betriebsmodellen an beliebiger Stelle in einer Sequenz von Prozessen stattfinden.

Ein Prozess besteht je nach Ausprägung des Elementes `Scenario` nur aus einem eXtra-Request (`scenario=fire-and-forget`) oder aus einem eXtra-Request gefolgt von einem eXtra-Response (`scenario=request-with-acknowledgement` bzw. `scenario=request-with-response`). Jeder Prozess beginnt mit einem eXtra-Request, den der Sender (der eXtra-Client) an das eXtra-Empfangssystem (den eXtra-Server) sendet.

Zu den Betriebsmodellen mit asynchron verarbeitenden Fachverfahren auf Empfänger- und/oder Senderseite gehören:

- Der Sendebetrieb (vgl. unten 2.2.1.1): Das Fachverfahren auf Senderseite übermittelt fachliche Daten an das zugeordnete Fachverfahren auf Empfängerseite. Das Fachverfahren auf Empfängerseite erzeugt keine Rückmeldungen für den Sender. Der Sender erhält keine Information darüber, ob das Fachverfahren auf Empfängerseite die fachlichen Daten erhalten und verarbeiten konnte bzw. ob - und wenn ja welche - Fehler aufgetreten sind. Beim Betriebsmodell „Sendebetrieb“ gibt es also ausschließlich Sendeprozesse, keine Holprozesse.
- Der Holbetrieb (vgl. unten 2.2.1.2): Das Fachverfahren auf Empfängerseite erzeugt fachliche Daten und stellt sie für das zugeordnete Fachverfahren auf Senderseite zum Abholen bereit. Das Fachverfahren auf Senderseite holt die bereitgestellten fachlichen Daten ab. Es bestätigt weder den erfolgreichen Abholprozess noch erzeugt es Rückmeldungen für das erzeugende Fachverfahren auf Empfängerseite. Beim Betriebsmodell „Holbetrieb“ gibt es also ausschließlich Holprozesse, keine Sendeprozesse.
- Der Sende-Holbetrieb oder Sende-Hol-Bestätigungsbetrieb (vgl. unten 2.2.1.3): Das Fachverfahren auf Senderseite übermittelt wie beim Sendebetrieb fachliche Daten an das zugeordnete Fachverfahren auf Empfängerseite. Das Fachverfahren auf

Empfängerseite erzeugt Rückmeldungen für den Sender und stellt sie zum Abholen bereit. Die Senderseite holt die bereitgestellten Rückmeldungen ab und bestätigt evtl. den erfolgreichen Abholprozess. Das Betriebsmodell „Sende-Holbetrieb“ ist durch Sende- und nachfolgende Holprozesse und das Betriebsmodell „Sende-Hol-Bestätigungsbetrieb“ ist durch Sende-, nachfolgende Hol- und abschließenden Bestätigungsprozess(e) gekennzeichnet.

- Der Hol-Bestätigungsbetrieb oder Hol-Bestätigungs-Sendebetrieb (vgl. unten 2.2.1.4):  
Wie beim Holbetrieb erzeugt das Fachverfahren auf Empfängerseite fachliche Daten für das zugeordnete Fachverfahren auf Senderseite und stellt sie zum Abholen bereit. Zusätzlich zum Holbetrieb bestätigt die Senderseite den erfolgreichen Abholprozess. Möglicherweise sendet das Fachverfahren auf Senderseite darüber hinaus eine Rückmeldung an das erzeugende Fachverfahren auf Empfängerseite, dass die bereitgestellten fachlichen Daten erhalten wurden und verarbeitet werden konnten (oder bei der Verarbeitung Fehler aufgetreten sind). Das Betriebsmodell „Hol-Bestätigungsbetrieb“ ist durch Hol- und nachfolgende Bestätigungsprozesse und das Betriebsmodell „Hol-Bestätigungs-Sendebetrieb“ durch Hol-, Bestätigungs- und abschließende Sendeprozesse gekennzeichnet.
- Der Sende-Hol-Bestätigungs- und Hol-Bestätigungs-Sendebetrieb (vgl. unten 2.2.1.5):  
Sowohl das Fachverfahren auf Senderseite wie auch das Fachverfahren auf Empfängerseite erzeugen fachliche Daten für das zugeordnete Fachverfahren auf der jeweils anderen Seite. Auf jeder Seite gibt es also sowohl erzeugende wie auch konsumierende/verarbeitende Fachverfahren. Jedes konsumierende/verarbeitende Fachverfahren erzeugt jeweils Rückmeldungen mit Informationen über den Erfolg bzw. Misserfolg bei der Verarbeitung der fachlichen Daten.

## 2.2.1. Betriebsmodelle mit asynchron verarbeitenden Fachverfahren

### 2.2.1.1. Das Betriebsmodell Sendebetrieb

Beim Sendebetrieb wird zwischen einem „einfachen Sendebetrieb“ mit dem Szenario „fire-and-forget“ und dem „Sendebetrieb“ mit dem Szenario „request-with-acknowledgement“ unterschieden.

#### Einfacher Sendebetrieb mit Sendeprozess und Szenario „fire-and-forget“

Der eXTra Sender erhält vom eXTra-Empfangssystem weder eine Empfangsbestätigung (mittels acknowledgement) noch eine spätere Verarbeitungsbestätigung durch das Fachverfahren auf Empfängerseite. Der einfache Sendebetrieb verläuft also in Form einer Einbahnstraße.

Dieses Betriebsmodell ist nur dann empfehlenswert, wenn

- der Sender keine Haftung für die rechtzeitige, vollständige oder korrekte Lieferung seiner fachlichen Daten übernehmen muss,
- die fachlichen Daten sehr einfach strukturiert sind und/oder nur selten übermittelt werden müssen,
- die fachlichen Daten höchstwahrscheinlich korrekt sind und sehr selten Reklamationen erfolgen,
- die Notwendigkeit der Korrektur bzw. der Nachlieferung fachlich unvollständiger oder inkorrektur Daten nicht existiert bzw. der Aufwand hierfür sehr gering ist bzw. vertretbar erscheint,
- das Risiko nennenswerter Kosten wegen verspäteten, unvollständigen oder inkorrekten fachlichen Daten nicht existiert bzw. vertretbar erscheint.

Für den einfachen Sendebetrieb kommen folgende eXTra-Sprachmittel in Frage:

- Bei Bedarf die Plugins `Contacts`, `DataSource` und `DataTransforms`.
- Dagegen werden bei dieser Betriebsform keine eXTra-Standardnachrichten benötigt.

### Hinweis:

Beim eXTra-Standard gibt es keine Vorgabe, wie die DFÜ-Ebene ausgestaltet werden soll. Insofern kann dieses Betriebsmodell so realisiert werden, dass auf der DFÜ-Ebene z.B. E-Mail oder auch http(s) zum Einsatz kommt. Im Fall von http(s) erhält der Sender auf der DFÜ-Ebene zwar evtl. eine http(s)-response, jedoch definitiv keine eXTra-Response.

### Sendebetrieb mit Sendeprozess und Scenario "request-with-acknowledgement"

Dieses Betriebsmodell ist empfehlenswert, wenn

- es für den Sender wichtig ist, die rechtzeitige und vollständige Lieferung seiner fachlichen Daten nachweisen zu können,
- der Sender keine Haftung für die korrekte und vollständige Lieferung seiner fachlichen Daten übernehmen muss,
- die fachlichen Daten einfach strukturiert sind und/oder nur selten übermittelt werden müssen,
- die fachlichen Daten höchstwahrscheinlich korrekt sind und sehr selten Reklamationen erfolgen,
- der Aufwand für die Korrektur inkorrektur Daten gering ist bzw. vertretbar erscheint,
- das Risiko nennenswerter wegen durch inkorrektur fachlicher Daten nicht existiert bzw. vertretbar erscheint.

Für den Sendebetrieb kommen - wie für den einfachen Sendebetrieb - folgende eXTra-Sprachmittel in Frage:

- Bei Bedarf die Plugins `Contacts`, `DataSource` und `DataTransforms`,
- von den eXTra-Standardnachrichten kommt bei Bedarf lediglich **RepeatResponse** bzw. **RepeatResponseRequest** in Frage.

#### **2.2.1.2. Das Betriebsmodell Holbetrieb**

Der einfache Holbetrieb mit Anforderungsprozess und Szenario „request-with-response“ hat folgende charakteristische Merkmale:

Der Sender fordert vom Erzeuger - formal der Empfängerseite, technisch dem eXTra-Server - fachliche Daten an, z.B. Rechnungen, die das erzeugende Fachverfahren irgendwann

bereitgestellt oder auch sofort erzeugt hat. Der Erzeuger verlangt keine Bestätigung vom Abholer, dass dieser die fachlichen Daten erfolgreich abgeholt und/oder verarbeitet hat. Der Datenfluss verläuft beim einfachen Holprozess also in Form einer Einbahnstraße.

Dieses Betriebsmodell ist dann empfehlenswert, wenn

- der Erzeuger - formal die Empfängerseite, technisch der eXtra-Server - z.B. auf Grund gesetzlicher Vorgaben nicht nachweisen muss, dass der Abholer - formal der Sender, technisch der eXtra-Client - tatsächlich die fachlichen Daten erhalten hat,
- es für den Erzeuger irrelevant ist, ob und wann der Abholer die fachlichen Daten abgeholt hat,
- die fachlichen Daten höchstwahrscheinlich korrekt sind und sehr selten Reklamationen durch den Abholer erfolgen,
- der Aufwand für die Korrektur inkorrektur Daten gering ist bzw. vertretbar erscheint,
- das Risiko nennenswerter Kosten wegen inkorrektur fachlicher Daten nicht existiert bzw. vertretbar erscheint.

Für den einfachen Holbetrieb kommen folgende eXtra-Sprachmittel in Frage:

- Bei Bedarf die Plugins `Contacts`, `DataSource` und `DataTransforms`.
- Die eXtra-Standardnachrichten **DataRequest** und evtl. **RepeatResponse** bzw. **RepeatResponseRequest**, sowie je nach Bedarf **ListRequest**, **ListResponse** und gegebenenfalls die entsprechenden **ListOf**- Nachrichten.

### **Empfehlung:**

Da die erzeugende Seite (formal die Empfängerseite, der eXtra-Server) keine Information von der konsumierenden Seite erhält (formal der Senderseite, der eXtra-Client), ob und wann die bereitgestellten Daten abgeholt wurden, muss die erzeugende Seite festlegen, wie lange sie die bereitgestellten Daten zum Abholen zur Verfügung stellt. Diesen entscheidenden Betriebsparameter muss die erzeugende Seite – der eXtra-Server - der konsumierenden Seite – dem eXtra-Client - verbindlich bekanntgeben.

### 2.2.1.3. Das Betriebsmodell Sende- Holbetrieb

Dieses Betriebsmodell ist dann empfehlenswert, wenn

- es für den Sender wichtig ist, die rechtzeitige, vollständige und korrekte Lieferung seiner fachlichen Daten nachweisen zu können,
- der Sender die Haftung für die rechtzeitige, vollständige und korrekte Lieferung seiner fachlichen Daten übernehmen muss,
- die zu sendenden fachlichen Daten komplex sind und/oder häufig übermittelt werden müssen,
- Reklamationen durch den Empfänger wegen unvollständigen oder fehlerhaften Daten möglichst vermieden werden sollen bzw. möglichst effizient gemeldet und bearbeitet werden können,
- Korrektur- bzw. Nachlieferungen durch den Sender möglichst effizient und noch rechtzeitig erfolgen können, um das Risiko nennenswerter Kosten wegen unvollständigen oder inkorrekten fachlichen Daten möglichst gering zu halten,
- beim Holbetrieb die Rückmeldungen einfach strukturiert sind und deshalb Fehler in den Rückmeldungen nahezu ausgeschlossen werden können oder wenn die Rückmeldungen Fehlermeldungen enthalten, die von Sachbearbeitern individuell bearbeitet werden müssen.

Für den Sende- Holbetrieb kommt das folgende Spektrum der eXtra-Sprachmittel in Frage:

- Bei Bedarf die Plugins `Contacts`, `DataSource` und `DataTransforms`.
- Die eXtra-Standardnachrichten **DataRequest** und **RepeatResponse** bzw. **RepeatResponseRequest**, sowie je nach Bedarf **ListRequest/ListResponse**, **StatusRequest/StatusResponse** und gegebenenfalls die entsprechenden **ListOf**-Nachrichten.

Beim Sende-Hol-Bestätigungsbetrieb kommt noch die Standardnachricht **ConfirmationOfReceipt** dazu.

#### **2.2.1.4. Das Betriebsmodell Hol- Bestätigungsbetriebs und Hol- Bestätigungs-Sendebetrieb**

Der Holbetrieb mit Anforderungsprozess und `scenario=request-with-response`, sowie anschließendem Bestätigungsprozess ist folgendermaßen gestaltet:

Der Sender fordert vom Erzeuger - formal der Empfängerseite, technisch dem eXtra-Server - fachliche Daten (z.B. Rechnungen) an, die das Fachverfahren auf Erzeugerseite irgendwann bereitgestellt oder auch sofort erzeugt hat. Der Erzeuger verlangt eine Bestätigung vom Abholer, dass dieser die fachlichen Daten erfolgreich abgeholt hat.

Dieses Betriebsmodell mit Bestätigungsprozess ist dann empfehlenswert, wenn

- der Erzeuger - formal die Empfängerseite, technisch der eXtra-Server - z.B. auf Grund gesetzlicher Vorgaben nachweisen muss, dass der Abholer - formal der Sender, technisch der eXtra-Client - tatsächlich die fachlichen Daten erhalten hat,
- es für den Erzeuger wichtig ist, ob und wann der Konsument die fachlichen Daten abgeholt hat,
- die abgeholten fachlichen Daten höchstwahrscheinlich korrekt sind und sehr selten Reklamationen durch den Konsumenten erfolgen,
- der Aufwand des Erzeugers für die Korrektur inkorrektur fachlicher Daten gering ist bzw. vertretbar erscheint,
- das Risiko nennenswerter Kosten beim Erzeuger wegen inkorrektur fachlicher Daten nicht existiert bzw. vertretbar erscheint.

Für den Hol- Bestätigungsbetrieb kommen folgende eXtra-Sprachmittel in Frage:

- Bei Bedarf die PlugIns `Contacts`, `DataSource` und `DataTransforms`.
- Die eXtra-Standardnachrichten **DataRequest**, **ConfirmationOfReceipt** und evtl. **RepeatResponse** bzw. **RepeatResponseRequest**, sowie je nach Bedarf **ListRequest**, **ListResponse** und gegebenenfalls die entsprechenden **ListOf**- Nachrichten.

Eine qualitative Steigerung ist der Hol-Bestätigungs-Sendebetrieb, also der Holbetrieb mit Anforderungsprozess, anschließendem Bestätigungsprozess und abschließender Rückmeldung durch das konsumierende/verarbeitende Fachverfahren auf Senderseite.

Der Sender fordert vom Erzeuger, formal der Empfängerseite, fachliche Daten an, die das Fachverfahren auf Erzeugerseite irgendwann bereitgestellt oder auch sofort erzeugt hat. Der Erzeuger verlangt eine Bestätigung vom Abholer, dass dieser die fachlichen Daten erfolgreich abgeholt hat. Nach Verarbeitung der abgeholtten fachlichen Daten erzeugt und versendet das konsumierende/verarbeitende Fachverfahren auf Senderseite jeweils eine Rückmeldung als Verarbeitungsquittung, in dem es dem Erzeuger die erfolgreiche oder fehlerhafte Verarbeitung mitteilt.

Dieses Betriebsmodell des Hol-Bestätigungs-Sendebetriebs, also der Holbetrieb mit Bestätigung und abschließender Rückmeldung, ist dann empfehlenswert, wenn

- der Erzeuger - formal die Empfängerseite, technisch der eXtra-Server - z.B. auf Grund gesetzlicher Vorgaben nachweisen muss, dass der Abholer - formal der Sender, technisch der eXtra-Client - tatsächlich die fachlichen Daten erhalten hat,
- es für den Erzeuger wichtig ist, ob und wann der Abholer die fachlichen Daten abgeholt hat,
- die fachlichen Daten komplex sind und/oder häufig bereitgestellt werden müssen,
- Reklamationen durch das konsumierende/verarbeitende Fachverfahren wegen unvollständigen oder fehlerhaften Daten möglichst vermieden werden sollen bzw. möglichst effizient gemeldet und bearbeitet werden können,
- Korrektur- bzw. Nachlieferungen durch das erzeugende Fachverfahren möglichst effizient und immer noch rechtzeitig erfolgen sollen, um das Risiko nennenswerter Kosten wegen unvollständigen oder inkorrekten fachlichen Daten möglichst gering zu halten.

Hierzu sind zwar keine weiteren zusätzlichen eXtra-Sprachmittel gegenüber dem Hol-Bestätigungsbetrieb ohne fachliche Rückmeldungen durch das konsumierende Fachverfahren auf Senderseite erforderlich, jedoch treten in analoger Weise die zusätzlichen Frage- und Problemstellungen rund um die Rückmeldungen wie beim Betriebsmodell des Sende-Holbetriebs auf, lediglich mit vertauschten Seiten.

### 2.2.1.5. Das Betriebsmodell Sende-Hol Bestätigungs- und Hol-Bestätigungs-Sendebetrieb

Beim Sende-Hol-Bestätigungsbetrieb und Hol-Bestätigungs-Sendebetrieb gibt es zusätzlich das Szenario, dass der Beginn der Prozesskette nicht beim Fachverfahren auf Senderseite, sondern beim Fachverfahren auf Empfängerseite liegt, welches fachliche Daten zum Abholen für den Sender erzeugt und bereitstellt. Beide konsumierende/verarbeitende Fachverfahren erzeugen Rückmeldungen, mit denen sie jeweils das erzeugende Fachverfahren über den Erfolg oder Misserfolg der Verarbeitung informieren.

Dieses Betriebsmodell ist dann empfehlenswert, wenn

- es für das erzeugende Fachverfahren wichtig ist, die rechtzeitige, vollständige und korrekte Lieferung bzw. Bereitstellung der fachlichen Daten nachweisen zu können,
- das erzeugende Fachverfahren die Haftung für die rechtzeitige, vollständige und korrekte Lieferung bzw. Bereitstellung der fachlichen Daten übernehmen muss,
- die fachlichen Daten komplex sind und/oder häufig übermittelt bzw. bereitgestellt werden müssen,
- Reklamationen durch das konsumierende/verarbeitende Fachverfahren wegen unvollständigen oder fehlerhaften Daten möglichst vermieden werden sollen bzw. möglichst effizient gemeldet und bearbeitet werden können,
- Korrektur- bzw. Nachlieferungen durch das erzeugende Fachverfahren möglichst effizient und immer noch rechtzeitig erfolgen sollen, um das Risiko nennenswerter Kosten wegen unvollständigen oder inkorrekten fachlichen Daten möglichst gering zu halten

Für beide Betriebsmodelle, dem Sende-Hol-Bestätigungsbetrieb und dem Hol-Bestätigungs-Sendebetrieb, kommt das gesamte Spektrum der eXtra-Sprachmittel in Frage:

- Bei Bedarf die Plugins `Contacts`, `DataSource` und `DataTransforms`.
- Die eXtra-Standardnachrichten **DataRequest**, **ConfirmationOfReceipt** und **RepeatResponse**, bzw **RepeatResponseRequest**, sowie je nach Bedarf **ListRequest**, **ListResponse**, **StatusRequest**, **StatusResponse** und gegebenenfalls die entsprechenden **ListOf**- Nachrichten.

Die Empfehlungen, Klarstellungen und Hinweise beim Sende-Hol-Bestätigungsbetrieb sowie beim Hol-Bestätigungs-Sendebetrieb gelten in identischer Weise.

Bei diesem Betriebsmodell ist nicht nur der Fall eines technischen Fehlers beim Empfänger nach einem erfolgreichen Sendeprozess (z.B. beim Entschlüsseln oder Dekomprimieren der fachlichen Daten einer Lieferung/Package/Message) relevant, sondern in analoger Weise das Problem technischer Fehler beim Sender nach einem erfolgreichen Abholprozess.

Dies ist die analoge Problematik zum sog. AcknowledgementUpdate auf Empfängerseite nach einem erfolgreichen Sendeprozess und sollte deshalb vergleichbar behandelt werden.

**Empfehlung:**

In diesem Fall sollte das Abholsystem auf Senderseite ebenso eine Rückmeldung für die betroffene Lieferung/Package/Message in Form eines sog. AcknowledgementUpdate mit Bezug auf die eXtra-Response der ausgelieferten Lieferung/Package/Message erzeugen und so für die Empfängerseite eine entsprechende Fehlermeldung bereitstellen (zur Problematik des AcknowledgementUpdates siehe auch 4.2.1).

**Empfehlung:**

Weiterhin sollte das Abholsystem auf Senderseite jeweils eine Rückmeldung in Form einer Verarbeitungsquittung erzeugen (2.2.1.3 und 2.2.1.4).

Auf diese Weise bleibt die Symmetrie zwischen Sender- und Empfängerseite gewahrt. Das erzeugende Fachverfahren erhält immer Rückmeldungen in Form von Verarbeitungsquittungen aber auch bei technischen Fehlern für alle seine gesendeten oder bereitgestellten Lieferungen/Packages/Messages, unabhängig davon, ob das erzeugende Fachverfahren auf Sender- oder Empfängerseite angesiedelt ist.

### 2.2.2. Betriebsmodell Dialogbetrieb mit synchron verarbeitenden Fachverfahren

Das Betriebsmodell Dialogbetrieb hat folgende charakteristische Merkmale:

- Das Fachverfahren auf Senderseite übermittelt fachliche Daten an das zugeordnete Fachverfahren auf Empfängerseite und erhält sofort (synchron in dieser Anschaltung) dessen Rückmeldung. Im Falle einer Anfrage, Anforderung einer Auskunft oder von fachlichen Daten durch den Sender enthält die Rückmeldung des Empfängers die entsprechenden fachlichen Daten. Für den Fall, dass der Sender fachliche Daten zur Verarbeitung durch das Fachverfahren auf Empfängerseite übermittelt, enthält die Rückmeldung bereits das Ergebnis der Verarbeitung.

Beim Dialogbetrieb geschieht der auslösende Sendeprozess, der eXtra-Request, mit dem Szenario „request-with-response

Für den Dialogbetrieb kommen folgende eXtra-Sprachmittel in Frage:

- Bei Bedarf die Plugins `Contacts`, `DataSource` und `DataTransforms`
- Die eXtra-Standardnachrichten erscheinen hier nicht als erforderlich

### **3. Gestaltung eines verbundspezifischen eXtra-Standards**

#### **3.1. Gestaltung der Schemadateien eines verbundspezifischen eXtra-Standards**

##### **3.1.1. Profilierung und Verwendung der Schemadateien**

Da eine Profilierung stets eine Restriktion darstellt, können eXtra-Nachrichten prinzipiell sowohl mittels unprofilierter wie auch profilierter Schemadateien validiert werden. Die Verwendung profilierter Schemadateien verbessert die Prozesssicherheit, da sie die syntaktische Korrektheit der Nachrichten eines konkreten Datenübermittlungsverbundes über den XML-Parser sicherstellt.

In Szenarien, in denen der fachliche Kontext unbekannt oder irrelevant ist, ist unter Umständen die Verwendung der unprofilierten Schemata notwendig bzw. vorteilhaft. So kann beispielsweise ein Service-Provider, der eXtra-Dienste für mehrere Datenübermittlungsverbünde anbietet, in einer äußeren, ersten Stufe die prinzipielle eXtra-Konformität mittels unprofilierter Schemadateien sicherstellen und in einer inneren, nachgelagerten zweiten Stufe die fachliche Konformität anhand profilierter Schemadateien überprüfen. Dabei ist zwingend auf eine konsistente Verwendung der unprofilierten Schemata zu achten. Profilierte Schemata dürfen nicht auf einer höheren Version beruhen.

##### **3.1.2. Prozessübergreifende oder prozessspezifische Schemadateien**

Der unprofilierter eXtra-Basisstandard stellt (neben den Schemadateien für die Plugins, die Components und die Standardnachrichten) zwei Schemadateien zur Verfügung: Eine für einen eXtra-Request und eine für eine eXtra-Response. Die Schemadateien gelten in gleicher Weise für alle Prozesse, sind also prozessübergreifend definiert.

Will man jedoch darüber hinaus eine maximale Prozesssicherheit bei jedem Prozess eines Datenübermittlungsverbundes (senden, holen, bestätigen) für beide Seiten erreichen bzw. die syntaktische Korrektheit bei jedem Prozess möglichst über den XML-Parser verifizieren, dann bieten sich prozessspezifische Schemadateien an. D.h. im Rahmen der Profilierung werden für jeden Prozess jeweils zwei spezifische Schemadateien generiert: Eine für den eXtra-Request und eine für die eXtra-Response. Damit kann mittels XML-Parser validiert werden, dass jede eXtra Nachricht exakt nur die für den jeweiligen Prozess zulässigen Werte der Elemente bzw. Attribute enthält.

### **Empfehlung:**

Wenn das Betriebsmodell eines Datenübermittlungsverbundes mehrere Prozesse umfasst, z.B. den Sende-, Hol- und Bestätigungsprozess oder den Hol- und Bestätigungsprozess, dann wird empfohlen, bei der Profilierung prozessspezifische Schemadateien zu generieren.

Beispiele für profilierte und darüberhinaus prozessspezifische Schemadateien sind die Schemadateien des GKV- oder des DSRV- Kommunikationsservers (siehe die registrierten Verfahren auf der eXTra-Homepage [www.extra-standard.de](http://www.extra-standard.de)).

### **3.1.3. Schemadateien für eXTra-Request und eXTra-Response**

Für den Vorgang eines eXTra-Request und der dazugehörigen Antwort mittels eXTra-Response gibt es unterschiedliche Schemadateien. Im unprofilierten eXTra-Basisstandard Transport sind die jeweils korrespondierenden eXTra-Objekte, z.B. der eXTra-Request `TransportHeader` und der eXTra-Response `TransportHeader` (mit Ausnahme der `ResponseDetails`) strukturell identisch definiert. Das Gleiche gilt für die Definition der drei Ebenen, auch sie sind im unprofilierten eXTra-Basisstandard eXTra-Transport strukturell identisch definiert.

Im Rahmen der Profilierung besteht nun die Möglichkeit, die Schemadateien für eXTra-Request und eXTra-Response wie auch die verschiedenen Ebenen in den Schemadateien jeweils unterschiedlich zu definieren.

### **Empfehlung für eine eXTra-Response auf einen vorangegangenen eXTra-Request:**

Die jeweils korrespondierenden eXTra-Objekte (z.B. des eXTra-Request `TransportHeader` und des eXTra-Response `TransportHeader`) werden strukturell identisch profiliert. Damit können die im laufenden Betrieb beim eXTra-Request vergebenen Werte der verschiedenen Elemente in einem eXTra-Request-Header problemlos beim eXTra-Response in die entsprechenden Elemente des eXTra-Response-Headers übertragen werden.

Die Möglichkeit der unterschiedlichen Festlegung wird deshalb nicht empfohlen.

### **Empfehlung für die Betriebsmodelle mit einem Holprozess:**

Für diese Betriebsmodelle werden für den eXtra-Response-Vorgang die Header aller eXtra-Ebenen, z.B. der `TransportHeader` und der `PackageHeader` strukturell so profiliert, dass die untere eXtra-Ebene (z.B. die `Package`-Ebene) mindestens die Elemente enthält wie die obere eXtra-Ebene (z.B. die eXtra-Transport-Ebene), evtl. jedoch mehr Elemente. Damit können bei einem Holprozess die im laufenden Betrieb beim eXtra-Request mittels **DataRequest** im eXtra-Request `TransportHeader` spezifizierten Werte problemlos beim dazugehörigen eXtra-Response zunächst in die entsprechenden Elemente des eXtra-Response `TransportHeader` und anschließend in die des `PackageHeader` übertragen werden.

### **3.1.4. Relation der Schemadateien des eXtra-Basisstandards gegenüber denjenigen eines verbundspezifischen eXtra-Standards**

Die Neuerungen in den Schemadateien des eXtra-Basisstandards Transport werden im Zuge der Weiterentwicklung des eXtra-Standards kommentiert, wie im Dokument „Überblick strukturierte Kommentare“ dargelegt.

#### **Empfehlung:**

Bei der Entwicklung der verbundspezifischen Schemadateien sollten die im Zuge der Profilierung gegenüber dem Basisstandard vorgenommenen Änderungen nach den gleichen Regeln dokumentiert werden.

#### **Empfehlung:**

In den verbundspezifischen Schemadateien sollte auch der Bezug zur Version des verwendeten Basisstandards dokumentiert werden, z.B.: „Ausgangspunkt für die Profilierung ist die Version x.y des eXtra-Basisstandards.“

Dies gilt insbesondere für eXtra-Server, die mehrere Datenübermittlungsverbünde bedienen.

### 3.1.5. Identifikation des Profils

Das Root-Element eines eXtra-Requests führt das Pflichtattribut `profile`. Es wird zur Hinterlegung einer ID verwendet, die auf das dem Vorgang zugrundeliegende eXtra-Profil hinweist. Das Format der ID ist das eines URI (Universal Resource Identifier).

#### **Empfehlung:**

Für jede Profilierung soll durch die zuständigen Verantwortlichen eine ID festgelegt und veröffentlicht werden. Die ID soll die Form eines http-URL unter Verwendung einer der den Verantwortlichen zur Verfügung stehenden Domain haben. Die ID wird im Root Element des eXtra-Requests eingetragen.

## 3.2. Gestaltung der Ebenen eines verbundspezifischen eXtra-Standards

### 3.2.1. Sendeprozess

Beim Sendeprozess ist der Einfluss der gegebenen Topologie eines Datenübermittlungsverbundes, der Einsatz von Dienstleistern und der notwendige Datenschutz auf die notwendige Gestaltung der eXtra-Ebenen besonders deutlich.

Wenn die Rolle des physikalischen Empfängers durch einen Dienstleister erbracht wird und/oder der physikalische Empfänger die zentrale Annahmestelle für mehrere logische Empfänger bzw. für mehrere Fachverfahren ist, oder der Datenschutz/die Vertraulichkeit bis zum logischen Empfänger oder dem jeweiligen Fachverfahren gewahrt werden soll, dann sind mehrere eXtra-Ebenen erforderlich. Der Einfluss dieser Faktoren auf die konkrete Ausgestaltung wird nachfolgend mit Beispielen erläutert.

Beispiel für die Nutzung nur einer Ebene ist die Übermittlung von Sofortmeldungen an die Rentenversicherung. Hier ist die Datenstelle der Träger der Rentenversicherung (DSRV) zugleich der physikalische wie auch der logische Empfänger. In einer Lieferung dürfen nur Fachnachrichten für ein einziges Fachverfahren enthalten sein. Die Fachnachrichten (die Sofortmeldungen) dürfen von mehreren Unternehmen stammen, müssen jedoch für die Rentenversicherung auf Anwendungs-Ebene verschlüsselt werden. Um diese Anforderungen

zu erfüllen, genügt eine eXTra-Ebene, die Transport-Ebene. Die verschlüsselten Fachnachrichten befinden sich im `TransportBody`.

Beispiel für den Einsatz von zwei Ebenen ist ebenfalls die Rentenversicherung bei der Übermittlung für das Fachverfahren der elektronisch unterstützten Betriebsprüfung (euBP). Hier ist zwar die DSRV wie bei den Sofortmeldungen ebenfalls zugleich der physikalische wie auch der logische Empfänger. In einer Lieferung dürfen die Fachnachrichten mehrerer Unternehmen enthalten sein, aber das Fachverfahren euBP will eine unternehmensspezifische Belieferung ihrer Fachnachrichten – die Begründung für den Einsatz von zwei eXTra-Ebenen, wobei wegen der unternehmensspezifischen Belieferung die zweite Ebene nicht die Package-, sondern die Message-Ebene ist.

Würde die DSRV in einer Lieferung auch Fachnachrichten für mehrere Fachverfahren zulassen, z.B. Sofortmeldungen und euBP-Meldungen, dann wäre hier als dritte eXTra-Ebene die Package-Ebenen erforderlich.

Ein Beispiel für den Einsatz von zwei Ebenen – der Transport- und der Package-Ebene - ist der Kommunikationsserver der gesetzlichen Krankenkassen, der GKV-Kommunikationsserver. Der GKV-Kommunikationsserver wird durch die ITSG betrieben. Der GKV-Kommunikationsserver ist der physikalische Empfänger für viele Fachverfahren unterschiedlicher logischer Empfänger, den sog. Datenannahme- und Verteilstellen, den DAVn. Der Datenschutz muss durch eine DAV-spezifische Verschlüsselung gewährleistet werden. Zur Erleichterung der Senderseite, insbesondere für Dienstleister, sind die Fachverfahren der GKV als Massenverfahren ausgelegt, d.h. eine Lieferung, ein eXTra Dokument, das an den GKV-Kommunikationsserver gesendet wird, darf Fachnachrichten an mehrere logische Empfänger (den DAVn) und gegebenenfalls für mehrere Fachverfahren enthalten. Diese Punkte führen dazu, dass zwei eXTra-Ebenen zum Einsatz kommen, die Transport- und die Package-Ebene. Eine unternehmensspezifische Belieferung der Fachverfahren ist nicht vorgesehen, da sonst zusätzlich zur Transport- und der Package-Ebene auch die Message-Ebene erforderlich wäre.

**Anzahl der Ebenen beim Sendeprozess mittels eXtra-Request und `Scenario=request-with-acknowledgement` und zugeordneter eXtra-Response.**

**Empfehlung:**

Wenn der Sendeprozess mittels eXtra-Request und `Scenario=request-with-acknowledgement` z.B. zwei eXtra-Ebenen umfasst, dann sollte auch die Antwort der Empfängerseite in Form einer eXtra-Response zwei Ebenen umfassen.

Die allgemeine Regel hierfür lautet, dass für jedes mittels eXtra-Request gesendete eXtra-Objekt in der dazugehörenden eXtra-Response auch eine korrespondierende Antwort zur Verfügung gestellt wird. Es sollte eine 1 zu 1 Beziehung zwischen den eXtra-Objekten eines eXtra-Request zu denen der dazugehörigen eXtra-Response bestehen.

Mit dieser Vorgehensweise erhält der Sender beim Betriebsmodell Sendebetrieb oder Sende-Holbetrieb den notwendigen Einblick auf den Zustand der Empfängerseite (welche Packages oder Messages wurden von der Empfängerseite angenommen und welche nicht), der wiederum Voraussetzung ist für einen Sendebetrieb mit teilweiser Annahme (siehe 4.1.1).

### **3.2.2. Holprozess**

Ein Holprozess wird mit einem eXtra-Request eingeleitet, wobei die konkrete Anforderung mit der Standardnachricht **DataRequest** formuliert wird, die sich wiederum im Datenteil der Transport-Ebene befindet. Beim eXtra-Request ist also nur eine Ebene – die Transport-Ebene – erforderlich.

Die Standardnachricht **DataRequest** ist so ausgelegt, dass sich eine konkrete Anforderung auf mehrere Rückmeldungen bzw. Fachnachrichten beziehen kann, gegebenenfalls von mehreren logischen Empfängern und/oder Fachverfahren. Wenn auf Grund einer derartigen Anforderung mittels **DataRequest** die Antwort in Form einer eXtra-Response möglichst alle davon betroffenen Rückmeldungen bzw. Fachnachrichten enthalten soll, dann ist dafür eine zweite Ebenen - die Package-Ebene - erforderlich. Damit ist eine saubere Trennung (z.B. von unterschiedlichen) Rückmeldungen möglich, die der Sender dann jeweils zu einer ursprünglichen Lieferung zuordnen kann.

Dies ist ein Beispiel dafür, dass sich die Anzahl notwendiger eXtra-Ebenen bei einem eXtra-Request mittels `Scenario=request-with-response` durchaus von denen beim zugeordneten eXtra-Response unterscheiden kann.

### 3.2.3. Bestätigungsprozess

Der Prozess der Bestätigung einer erfolgreichen Abholung von Rückmeldungen bzw. Fachnachrichten wird mit einem eXTra-Request eingeleitet, wobei die konkrete Bestätigung mit der Standardnachricht **ConfirmationOfReceipt** oder **ListOfConfirmationOfReceipt** formuliert wird, die sich wiederum im Datenteil der Transport-Ebene befindet. Beim eXTra-Request ist also nur eine Ebene – die Transport-Ebene – erforderlich.

Da der eXTra-Request mit `Scenario=request-with-acknowledgement` formuliert wird, sollte die Anzahl der Ebenen bei einer eXTra-Response identisch zum auslösenden eXTra-Request sein, besteht also ebenfalls nur aus einer Ebene, der Transport-Ebene (siehe 3.1.2).

## 3.3. Gestaltung der Header eines verbundspezifischen eXTra-Standards

### Gestaltung der Header im Rahmen der Profilierung

#### Klarstellung

Im eXTra-Standard spielen der `Header` und der `Body` eine besondere Rolle. Diese beiden Strukturelemente sind Pflichtelemente. Sie müssen im Rahmen der Profilierung als Pflichtelemente erhalten bleiben, d.h. sie dürfen in der Schemadatei weder als optional noch als nicht unterstützt (`ignored`) deklariert werden.

Im eXTra Basisstandard Transport gibt es im `Header` sowohl Pflicht-Elemente als auch optionale Elemente. Die im `Header` befindlichen Pflichtelemente müssen als solche erhalten bleiben, dürfen also beim Profilierungsvorgang weder zum optionalen Element umdefiniert noch als nicht unterstützt deklariert werden (`Ignored`).

Dagegen kann im Rahmen der Profilierung festgelegt werden, dass ein optionales Element im eXTra-Header zum Pflichtelement wird, oder dass ein optionales Element nicht unterstützt wird (`Ignored`).

Weiterhin kann spezifiziert werden, wie bezüglich der optionalen Elemente der `Header` auf Transport-, Package- und Message-Ebene definiert ist. Die Festlegung könnte ebenenspezifisch unterschiedlich sein, z.B. könnte das Element `Procedure` in den `RequestDetails` auf Transport-Ebene als optional oder `ignored` definiert werden, dagegen auf Package-Ebene als Pflichtelement. Damit wäre der `Header` auf den verschiedenen Ebenen strukturell gleich, im Detail jedoch unterschiedlich definiert.

**Klarstellung:**

Der Header ist strukturell für alle Fachverfahren eines Datenübermittlungsverbundes gleich, da die Profilierung verbundspezifisch, aber nicht fachverfahrenspezifisch erfolgt.

**Empfehlung:**

Bei den Betriebsmodellen mit Hol-Prozess ist zumindest für die eXTra-Response der Header strukturell für jede eXTra-Ebene gleich.

**Empfehlung:**

Wenn ein optionales Element im Rahmen der Profilierung als „Ignored“ definiert wird, ist damit folgende Vorstellung verknüpft:

Auf Senderseite: Der Sender hat die Wahl, ob er das so definierte Element

- weglässt oder
- als leeres Element weitergibt oder
- mit einem Wert versieht.

Auf Empfängerseite ist die Wirkung immer gleich:

Das Element - ob es fehlt, leer oder mit einem Wert belegt ist – wird akzeptiert, aber nicht ausgewertet. Wird das Element angegeben, muss es vom Empfänger akzeptiert werden.

**Das Element `Testindicator` und seine Bedeutung auf den unterschiedlichen Ebenen**

Für das Element `Testindicator` gibt es vier Ausprägungen mit folgender Bedeutung:

- none: kein Testfall, sondern Produktivfall
- receive: Test bis zur Annahme durch das eXTra-Empfangssystem der jeweiligen eXTra-Ebene auf Empfängerseite
- accept: Test bis zum Ende der Verarbeitung durch das eXTra-Empfangssystem der jeweiligen eXTra-Ebene auf Empfängerseite, d.h. gegebenenfalls mit der Entschlüsselung und Dekomprimierung der Daten sowie Verifikation der Signaturen und Erzeugung des Strukturelements Logging
- process: Test der jeweiligen eXTra-Ebene (mit allen Verarbeitungsschritten auf dieser eXTra-Ebene, gegebenenfalls einschließlich der Entschlüsselung und Dekomprimierung

der fachlichen Nachrichten sowie Verifikation der Signaturen und Erzeugung des Strukturelements Logging) und Weitergabe an die nächste Ebene bzw. bei der untersten eXtra-Ebene an das zugeordnete Fachverfahren

### **Dabei existieren folgende Regeln:**

Ein Produktivfall auf der eXtra-Transport-Ebene kann im Zuge der Weiterbearbeitung auf den unteren Ebenen, z.B. auf der eXtra Package- oder eXtra-Message-Ebene, zum Testfall werden. D.h. das Produktivsystem auf den höheren eXtra-Ebenen kann zum Transport eines Testfalls der unteren eXtra-Ebenen verwendet werden.

Das Umgekehrte ist jedoch nicht zulässig: Ein Testfall auf der eXtra-Transport-Ebene kann auf der eXtra-Package- oder eXtra-Message-Ebene nicht zum Produktivfall werden (ein Testfall bleibt ein Testfall). Analoges gilt für einen Testfall auf der eXtra-Package-Ebene, dieser kann auf der eXtra-Message-Ebene nicht zum Produktivfall werden

### **Die Bedeutung des Elementes Testindicator auf den unterschiedlichen Ebenen:**

Transport-Ebene:

- receive: Test des Transports bis zur Annahme der Daten beim physikalischen Empfänger durch die eXtra-Transport-Ebene
- accept: Test des Transports zum physikalischen Empfänger bis zum Ende der Verarbeitung auf Transport-Ebene, d.h. ohne Weitergabe zur nächsten eXtra-Ebene bzw. zum log. Empfänger
- process: Test der Transport-Ebene und Weitergabe an die nächste eXtra-Ebene bzw. Weitergabe an das Fachverfahren beim logischen Empfänger

Package-Ebene:

- receive: Test des Transports bis zur Annahme der Daten durch die eXtra-Package-Ebene auf Empfängerseite
- accept: Test des Transports bis zum Ende der Verarbeitung auf Package-Ebene, d.h. ohne Weitergabe zur Message-Ebene bzw. den logischen Empfänger
- process: Test der Transport- und Package-Ebene und Weitergabe an die Message-Ebene bzw. Weitergabe an das Fachverfahren beim logischen Empfänger

#### Message-Ebene:

- `receive`: Test des Transports bis zur Annahme der Daten durch die eXtra-Message-Ebene auf Empfängerseite
- `accept`: Test des Transports bis zum Ende der Verarbeitung auf Message-Ebene, d.h. ohne Weitergabe zum Fachverfahren
- `process`: Test der Transport-, Package- und Message-Ebene und Weitergabe an das Fachverfahren beim Endempfänger.

#### Beispiele im Zusammenspiel mit drei eXtra-Ebenen:

##### Beispiel 1: Test nur mit der Transport-Ebene

- `TransportHeader`, `Testindicator` = `receive` oder `accept`, Testfall verbleibt in der TransportEbene
- `PackageHeader`, `Testindicator` irrelevant, da der Testfall nicht an die Package-Ebene weitergereicht wird
- `MessageHeader`, `Testindicator` irrelevant, da die Transport-Ebene den Testfall nicht an die unteren Ebenen weiterreicht

##### Beispiel 2: Test nur mit der Transport- und Package-Ebene

- `TransportHeader`, `Testindicator` = `process`, Testfall wird weitergereicht
- `PackageHeader`, `Testindicator` = `receive` oder `accept`, Testfall verbleibt in der Package--Ebene
- `MessageHeader`, `Testindicator` irrelevant, da die Packet-Ebene den Testfall nicht an die Message-Ebene weiterreicht

##### Beispiel 3: Test mit allen drei eXtra-Ebenen

- `TransportHeader`, `Testindicator` = `process`, Testfall wird weitergereicht
- `PackageHeader`, `Testindicator` = `process`, Testfall wird weitergereicht
- `MessageHeader`, `Testindicator` = `receive` oder `accept`, Testfall verbleibt in der Message-Ebene

Beispiel 4: Test mit Verarbeitung des Testfalls durch das Fachverfahren.

- `TransportHeader, Testindicator = process`, Testfall wird weitergereicht
- `PackageHeader, Testindicator = process`, Testfall wird weitergereicht
- `MessageHeader, Testindicator = process`, Testfall wird an das Fachverfahren weitergereicht und dort verarbeitet

Beispiel 5: Test mit Verarbeitung des Testfalls durch das Fachverfahren unter Verwendung des Produktivsystems auf der Transport- und Package-Ebene

- `TransportHeader, Testindicator = none`, wird auf der Transport-Ebene als Produktivfall behandelt und zur nächsten Ebene weitergereicht
- `PackageHeader, Testindicator = none`, wird auch auf der Package-Ebene als Produktivfall behandelt und zur nächsten Ebene weitergereicht
- `MessageHeader, Testindicator = process`, Testfall wird an das Fachverfahren weitergereicht und dort verarbeitet

### **Empfehlung:**

Das Element `Testindicator` ist im eXtra-Basisstandard Transport als optionales Element definiert. Im Rahmen der Profilierung sollte das Element nicht als Ignored deklariert werden. Ein eXtra-spezifisches Datenübermittlungsverfahren sollte prinzipiell Testmöglichkeiten zur Verfügung stellen.

### **Das Element `scenario` und dessen Definition auf verschiedenen Ebenen**

Scenario ist im eXtra Kommunikationsstandard das entscheidende Element, mit dem der Sender den gewünschten Kommunikationstypus definiert, d.h. ob und welche Antwort er von der Empfängerseite erwartet.

#### **Empfehlung:**

- Auf der Transport-Ebene sollte das Element `scenario` auf Grund seiner zentralen Bedeutung als Pflichtelement definiert werden. Damit ist auf Empfängerseite die eXtra-Instanz auf Transport-Ebene exakt darüber informiert, welche Reaktion die Senderseite bei einem Sendeprozess von der Empfängerseite erwartet:
  - Keine eXtra-Response, keine Reaktion vom eXtra-Empfangssystem bei `scenario=fire-and-forget`. Der Sendeprozess ist als Einbahnstrasse vom Sender zum Empfänger ausgelegt.
  - Eine eXtra-Response, eine Rückmeldung ohne fachlichen Daten – semantisch eine Empfangsbestätigung vom eXtra-Empfangssystem - bei `scenario=request-with-acknowledgement`
  - Eine eXtra-Response, eine Rückmeldung mit fachlichen Daten bei `scenario=request-with-response` vom verarbeitenden Fachverfahren auf Empfängerseite
  
- Auf der Package- und Message-Ebene kann das Element `scenario` formal ebenso wie auf der Transport-Ebene als Pflichtelement oder als ein optionales Element definiert sein.

### 3.4. Sicherheit, Datenschutz

#### **Empfehlung:**

Die Übermittlung eines eXTra Dokuments sollte – da sie üblicherweise über das Internet erfolgt – auf der DFÜ-Ebene prinzipiell verschlüsselt erfolgen, z.B. mittels https. Denn aus Sicht des eXTra Kommunikationsstandards wird der eXTra-TransportHeader der eXTra-Transport-Ebene für die eXTra-Empfangsinstanz des physikalischen Empfängers prinzipiell unverschlüsselt übermittelt.

#### **Hinweis:**

Bei schützenswerten (z.B. personenbezogenen) Daten ist die Verschlüsselung auf Anwendungsebene prinzipiell immer dann erforderlich, wenn die fachlichen Daten auf DFÜ-Ebene nicht verschlüsselt gesendet werden.

#### **Empfehlung:**

Selbst wenn die schützenswerten fachlichen Daten auf der DFÜ-Ebene verschlüsselt werden (z.B. mittels https), ist eine Verschlüsselung auf Anwendungsebene aus Datenschutzgründen immer dann empfehlenswert oder sogar gefordert, wenn die Empfangsinstanz des physikalischen Empfängers keine Kenntnis der übermittelten Nutzdaten erhalten soll, sondern ausschließlich die fachliche Verarbeitungsinstanz - das Fachverfahren - des logischen bzw. Endempfängers. Dies gilt insbesondere dann, wenn die Empfangsinstanz des physikalischen Empfängers durch einen Dritten (z.B. einem Dienstleister) zur Verfügung gestellt wird oder wenn die eXTra-Empfangsinstanz des physikalischen Empfängers und des logischen Empfängers bzw. Endempfängers - das Fachverfahren - an anderen Orten angesiedelt sind. Auch aus Datensicherheitsgründen ist eine Verschlüsselung auf Anwendungsebene sinnvoll, da diese nicht bereits auf der DFÜ-Ebene (z.B. der https-Instanz) des physikalischen Empfängers, sondern erst in dessen internen Netz terminiert werden, wodurch ein höheres Sicherheitsniveau erreicht wird.

## 4. Der laufende Betrieb – dynamische Aspekte

### 4.1. Verfügbarkeit des eXTra Systems auf Empfängerseite, die Nachricht ExtraError

Wenn das eXTra-System auf Empfängerseite nicht verfügbar ist, sei es wegen einer angekündigten und geplanten Wartungsmaßnahme oder wegen eines schwerwiegenden Fehlers, dann sollte die Empfängerseite die Nichtverfügbarkeit des eXTra-Systems möglichst nicht mit einer technischen Fehlermeldung (z.B. des DFÜ-Systems), sondern mit einer verständlichen Nachricht mitteilen. Für diesen Zweck steht die Nachricht ExtraError zur Verfügung. ExtraError ist eine spezielle Quittungsnachricht, mit der die Empfängerseite den Sender über die temporäre oder permanente Nichtverfügbarkeit des eXTra-Systems informieren kann. Darüber hinaus können mit der ExtraError Nachricht auch schwerwiegende Fehler im eXTra-Protokoll berichtet werden, z.B. wenn der Request nicht als eXTra-Request erkannt wurde.

#### **Empfehlung:**

Da die ExtraError Nachricht den Systemzustand des eXTra-Systems bzw. grobe Fehler im eXTra-Protokoll in verständlicher Weise mitteilt, wirkt sie sich auch als Serviceentlastung einer Hotline aus. Deshalb wird empfohlen, ExtraError zu implementieren.

### 4.2. Zusammenspiel der eXTra Senderseite mit der eXTra Empfängerseite

#### **Zeitstempel**

Die beiden Zeitstempel, im Element `Timestamp` in den `RequestDetails`, sowie den `ResponseDetails` in den eXTra-Headern zeigen auf, wann ein eXTra-Request bzw. ein eXTra-Response stattgefunden hat.

Für den Fall, dass die Empfängerseite eine Abgabepflicht zu einem bestimmten Stichtag und Stichzeitpunkt formuliert hat, kann der Sender mit dem Zeitstempel in den `ResponseDetails` - dem Eingangsstempel auf Empfängerseite - den Nachweis der tatsächlichen (rechtzeitigen) Abgabe führen.

**Hinweis:**

In der Praxis kommt es immer wieder vor, dass die Uhren auf Sender- wie auf Empfängerseite auseinanderlaufen. Deshalb kann es vorkommen, dass das eXTra-Empfangssystem z.B. eine Lieferung mit einem Zeitstempel erhält oder eine Auskunft geben soll zu einem Vorgang, der in der Zukunft liegt. Für solche Fälle muss zwischen Sender- und Empfängerseite geregelt sein, welche Uhr gelten soll, die der Sender-oder die der Empfängerseite.

**Empfehlung:**

Sowohl Sender- wie auch Empfängerseite müssen mit dem Phänomen der „auseinander laufenden Uhren“ zurechtkommen. Die wichtigste Konsequenz ist, dass die Senderseite anerkennt, dass die Uhr der Empfängerseite maßgeblich ist. Dies gilt insbesondere dann, wenn der Sender Abgabepflichten zu definierten Zeitpunkten erfüllen muss.

**4.2.1. Der Sendeprozess****Grenzen der Parallelität bzw. Notwendigkeit der Serialisierung des Sendeprozesses**

Eine unerlässliche Serialisierung des Sendeprozesses beim Sender ist bei allen Sendeprozessen bezüglich eines Fachverfahrens mit spezifiziertem Datentyp erforderlich, wenn es

- im Fachverfahren die Funktion der Stornierung einer Lieferung eines Datentyps gibt,
- im Fachverfahren einen Begriff gibt, der den Ausgabestand einer Lieferung eines Datentyps definiert und das Fachverfahren die Belieferung in einer lückenlosen aufsteigenden Reihenfolge erwartet (z.B. die laufende Dateinummer bei der GKV),
- eine simple zeitorientierte Verarbeitungsregel innerhalb des Fachverfahrens gibt, wonach die neueste Lieferung eine bestehende ältere Lieferung zum gleichen vorgegebenen Zeitraum überschreibt bzw. annulliert (z.B. zum gleichen Veranlagungszeitraum bei der Finanzverwaltung).

Die Frage ist, ob auf Grund einer erforderlichen Serialisierung der Sendeprozesse eines Fachverfahrens mit spezifiziertem Datentyp parallele Sendeprozesse dieses Datentyps ausgeschlossen oder unter bestimmten Umständen tolerabel und beherrschbar sind. Lässt der Empfänger parallele Sendeprozesse des gleichen Datentyps zu, so müssen die Systeme auf Empfangsseite (eXTra-Server und das zugeordnete Fachverfahren) das Phänomen

beherrschen, dass Überholvorgänge und temporäre Lücken auftreten könnten. Ein Überholvorgang kann z.B. dann leicht auftreten, wenn die Datenmenge der vorletzten Sendung groß (z.B. 10 MB), die der letzten Sendung jedoch klein ist (z.B. wenige KB). Die Systeme auf Empfangsseite dürfen nun die serielle Belieferung nicht sofort nach einem erfolgreichen Empfang einer Sendung überprüfen, sondern müssen zulassen, dass eine Lücke in der Belieferung innerhalb einer gewissen Toleranzzeit (z.B. innerhalb von einer oder wenigen Stunden) vom Sender geschlossen werden kann.

Eine Serialisierung des Sendeprozesses beim Sender ist bei Sendeprozessen unterschiedlicher Fachverfahren aus fachlichen Gründen nicht erforderlich. Es kann jedoch sein, dass der Empfänger die zulässige Parallelität von Sendeprozessen eines Senders aus technischen Gründen einschränkt, im Extremfall sogar untersagt (z.B. aus Kapazitätsgründen seiner Infrastruktur).

### **Annahmetransaktion oder teilweise Annahme eines Sendeprozesses**

Für das Betriebsmodell Sendebetrieb und Sende-Holbetrieb ist die Frage interessant, wie das eXtra-Empfangssystem auf Empfängerseite ausgelegt ist: Realisiert es eine Annahmetransaktion oder akzeptiert es auch eine teilweise Annahme?

eXtra-Annahmetransaktion bedeutet, dass das eXtra-Empfangssystem entweder die gesamte Lieferung, d.h. den gesamten TransportBody mit allen Daten bzw. mit allen Packages/Messages erfolgreich übernehmen konnte oder nichts bzw. kein(e)) einzige(s) Package/Message (Prinzip „Alles oder Nichts“).

Teilweise Annahme heißt, alle fehlerfrei angenommenen Packages/Messages werden vom eXtra-Empfangssystem zur Verarbeitung an das jeweilige Fachverfahren weitergeleitet. Die fehlerhaft oder nicht vollständig angenommenen Packages/Messages werden mit einer entsprechenden Fehlermeldung in den `ResponseDetails` an den eXtra-Sender zurückgemeldet.

Randbedingung ist, dass das jeweils kleinste eXtra-Objekt (je nachdem, wie viele eXtra-Ebenen eine Lieferung umfasst, ist dies eine Message, ein Package oder der TransportBody) immer vollständig übernommen wurde. Die teilweise Annahme bezieht sich also immer auf vollständige eXtra-Objekte, die jedoch nicht alle übernommen werden konnten.

**Empfehlung:**

Besteht eine Lieferung nur aus der eXTra-Transport-Ebene, so wird empfohlen, dass das eXTra-Empfangssystem eine Annahmetransaktion realisiert, denn das eXTra-Empfangssystem hat keinerlei Kenntnis über die Struktur und das Format der übermittelten fachlichen Daten, kann also nicht zurückmelden, bis wohin die fachlichen Daten übernommen werden konnten.

**Empfehlung:**

Besteht eine Lieferung aus mehreren eXTra-Ebenen, liegt es bei einem konkreten Datenübermittlungsverbund nicht von vornherein auf der Hand, ob eine Annahmetransaktion oder die teilweise Annahme günstiger sein würde.

Insofern verbleibt als Empfehlung lediglich der Hinweis, dass die Beschreibung des Empfangsprozesses eines konkreten Datenübermittlungsverbundes unbedingt eine Klarstellung enthalten sollte, welches Annahmemodell – Annahmetransaktion oder teilweise Annahme – umgesetzt wurde.

**Hinweis:**

Bei den beiden eXTra-Servern der gesetzlichen Krankenkassen (dem GKV KommServer) und der Datenstelle der Träger der Rentenversicherung DSRV (beim Fachverfahren „elektronisch unterstützte Betriebsprüfung“) ist das Betriebsmodell des Sende- Holbetriebs mit teilweiser Annahme realisiert.

**Zusammenspiel des eXTra-Request mit eXTra-Response, spezifiziert mit dem Element scenario**

Scenario gibt es in drei Ausprägungen:

- „Fire-and-forget“
- „request-with-acknowledgement“
- „request-with-response“

**Verwendung von Scenario=fire-and-forget**

Die Wirkung von „Fire-and-forget“ ist einfach: Der Sender verzichtet beim Sendeprozess auf eine eXTra-Response des Empfängers. Der Informationsfluss ist als Einbahnstraße vom Sender zum Empfänger organisiert – geeignet für das Betriebsmodell einfacher Sendebetrieb.

**Hinweis und Empfehlung:**

Interessant ist die Frage, wie das eXtra-Empfangssystem auf einen Sendeprozess mit `Scenario=fire-and-forget` reagiert. Die Antwort ist den profilierten Schemadateien zu entnehmen. Ist in der Schemadatei festgelegt, dass das Element `Scenario` nur oder auch den Wert `fire-and-forget` haben darf, dann darf das Empfangssystem keine eXtra-Response generieren. Ist jedoch in der Schemadatei definiert, dass das Element `Scenario` nur den Wert `request-with-acknowledgement` haben darf, dann hat sich der Sender nicht schemakonform verhalten. In diesem Fall könnte sich die Empfängerseite schemakonform verhalten und mit einer eXtra-Response antworten. Aus diesem Grund wird empfohlen, dass der eXtra-Client so realisiert ist, dass er prinzipiell eine eXtra-Response definiert behandeln kann. Dies gilt insbesondere dann, wenn das eXtra Empfangssystem die ExtraError Nachricht unterstützt. Denn dann muss der eXtra-Client prinzipiell bei jedem eXtra-Request – unabhängig vom Wert des Elementes `Scenario` - damit rechnen, dass er eine ExtraError Nachricht erhalten könnte.

**Verwendung von Scenario=request-with-acknowledgement**

Beim Betriebsmodell Sendebetrieb oder Sende-Holbetrieb erwartet der Sender bei einem Sendeprozess mit `Scenario=request-with-acknowledgement` eine Antwort der Empfängerseite in Form einer eXtra-Response. Die Antwort – das Acknowledgement - muss die Empfangsseite so schnell liefern können, dass z.B. beim DFÜ-Protokoll http(s) beim Warten des Senders auf die eXtra-Response kein Timeout erfolgt.

**Empfehlung:**

In einem Datenübermittlungsverbund ist die Festlegung des Timeouts ein wesentlicher Bestandteil der Vereinbarungen zwischen der Sender- und der Empfängerseite. Dieser Wert sollte für alle beteiligten Instanzen auf Sender- wie Empfängerseite bekannt und verbindlich geregelt sein, also sowohl für die DFÜ-Ebene als auch für die eXtra-spezifischen Instanzen gelten. Möglicherweise differiert der Timeout-Wert für einen Sendeprozess mit `request-with-acknowledgement` gegenüber einem Sende- oder Holprozess mit `request-with-response`.

**Klarstellung:**

Die Antwort der Empfängerseite – das Acknowledgement – erzeugt das eXtra-Empfangssystem, nicht das im Element `Procedure` genannte verarbeitende Fachverfahren

auf Empfängerseite. Das Acknowledgement ist immer eine Statusmeldung des eXtra-Empfangssystems – es ist nie eine fachliche Nachricht.

Die Bedeutung eines Acknowledgements kann maximal Auskunft über den Erfolg (oder Misserfolg) aller Bearbeitungsschritte im eXtra-Empfangssystem (einschließlich Übernahme der fachlichen Nachrichten in die lokale Datenhaltung, deren Schemaprüfung, Entschlüsselung und Dekomprimierung) bis zur Weitergabe an das verarbeitende Fachverfahren geben. Dies ist dann möglich, wenn die Datenmenge der gesendeten fachlichen Nachrichten überschaubar ist, z.B. weniger als 1 MB umfasst.

Im minimalen Fall kann das Acknowledgement lediglich Auskunft über die erfolgreiche Übernahme der fachlichen Nachrichten in die lokale Datenhaltung geben. Dies ist insbesondere dann der Fall, wenn sich die zulässige Datenmenge an fachlichen Nachrichten im größeren MB- oder sogar GB-Bereich befinden darf und die anschließende Schemaprüfung/Entschlüsselung/Dekomprimierung zu viel Zeit in Anspruch nehmen würde.

#### **Empfehlung:**

Bei einem realen eXtra-System ist die Bedeutung eines Acknowledgements in der Regel zwischen dem oben dargestellten Minimum und dem Maximum angesiedelt. Deshalb sollte das eXtra-Empfangssystem die Bedeutung eines Acknowledgements möglichst exakt darlegen und beschreiben über welche Bearbeitungsschritte im eXtra-Empfangssystem der Sender Auskunft erhalten kann.

Des Weiteren definiert der Wert des Timeouts zusammen mit den Bearbeitungsschritten, über die ein Acknowledgement Auskunft geben soll, implizit die Datenmenge, die ein Sendeprozess umfassen darf.

Beides - Größe des Timeout und maximal zulässige Datenmenge beim Sendeprozess - sind wesentliche Betriebsparameter, die zwischen Sender- und Empfängerseite verbindlich geregelt sein sollten.

### Verwendung von Scenario=request-with-response

Beim Dialogbetrieb oder beim Betriebsmodell Sende-Holbetrieb erwartet der Sender bei einem Sendeprozess mit `Scenario=request-with-response` eine Antwort des Fachverfahrens auf Empfängerseite in Form einer eXTra-Response. Die Antwort – genauso wie bei `scenario=request-with-acknowledgement` - muss das Fachverfahren auf Empfangsseite so schnell liefern können, dass z.B. beim DFÜ-Protokoll http(s) beim Warten des Senders auf die eXTra-Response kein Timeout erfolgt.

### Klarstellung:

Die Antwort der Empfängerseite erzeugt bei einem Sendeprozess mit `Scenario=request-with-response` nicht das eXTra-Empfangssystem, sondern das im Element `Procedure` genannte verarbeitende Fachverfahren auf Empfängerseite. Die vom Fachverfahren generierte Antwort in Form einer eXTra-Response ist immer eine fachliche Nachricht, die sich im Body der untersten eXTra-Ebene (z.B. der Package-Ebene) befindet.

### Hinweis:

Im Gegensatz zur Antwort der Empfängerseite auf Grund eines eXTRA-Request mit `Scenario=request-with-acknowledgement` ist bei einem eXTRA-Request mittels `Scenario=request-with-response` neben den Instanzen des eXTra-Empfangssystems auch das spezifizierte Fachverfahren mit eingebunden.

Aus diesem Grund limitiert die festgelegte Timeout-Größe für eine eXTra-Response implizit die Datenmenge beim Betriebsmodell mit Holprozessen bzw. den Datenumfang der Antwort beim Betriebsmodell Dialogbetrieb.

Beides - die Datenmenge bei Holprozessen wie auch der Datenumfang der Antwort beim Dialogbetrieb - sollte explizit und verbindlich zwischen Sender- und Empfängerseite geregelt sein.

### Das Element scenario auf mehreren eXTra-Ebenen

#### Empfehlung:

- Unabhängig davon, ob der Sendeprozess nur eine eXTra-Ebene oder mehrere eXTra-Ebenen umfasst, sollte im Rahmen der Profilierung das Element `Scenario` auf Transport-Ebene als Pflichtelement definiert sein. Auf diese Weise ist explizit sichergestellt, dass die

eXTra-Empfangsinstanz auf Transportebene immer weiß, welche Reaktion der Sender bei einem eXTra-Request erwartet.

- Ist das Element `scenario` auf Transport-Ebene dagegen als optional definiert, müsste die Empfängerseite eine Interpretation des Defaultwertes für `scenario` liefern. Da Interpretationsspielräume möglichst vermieden werden sollten, wird davon abgeraten, statt expliziter formaler Definitionen interpretationsabhängige Defaultwerte zuzulassen.
- Auf der Package- und Message-Ebene kann das Element `scenario` formal als Pflichtelement oder als ein optionales Element definiert sein. Bei einem Sendeprozess sollte der Sender das Element `scenario` auf jeder Ebene mit dem gleichen Wert (z.B. überall mit `request-with-acknowledgement` oder mit `request-with-response`) belegen.
- Ist das Element `scenario` auf der Package- und Message-Ebene als optional definiert, wird aber bei einem Sendeprozess vom Sender nicht mit einem Wert belegt, so sollte beim eXTra-Empfangssystem auf der Package- und Message-Ebene diejenige Belegung gelten, die der Sender auf der Transport-Ebene formuliert hat.

### Wiederholtes Senden bei einem Sendeprozess mit `scenario=request-with-acknowledgement`

Annahme: Der Sender hat eine fachliche Nachricht mit einem eXtra-Request der Art `scenario=request-with-acknowledgement` abgesendet und als Rückantwort keine eXtra-Response, sondern eine Fehlermeldung der DFÜ-Ebene (z.B. einen Timeout der http(s)-Instanz) erhalten. Für den Sender ist nun nicht entscheidbar, ob der Empfänger die vollständige Nachricht nicht empfangen und ablegen konnte, der Sender die Nachricht also noch einmal senden kann. Oder ob der Empfänger die Nachricht zwar empfangen und ablegen konnte, aber die eXtra-Response (aus welchem Grund auch immer) nicht zurückgeben konnte, z.B. weil der Proxy oder die DFÜ-Ebene auf einen Timeout lief.

Kann der Sender nun den Sendeprozess einfach wiederholen? Die Antwort auf diese Frage kann nur das verarbeitende Fachverfahren auf Empfängerseite geben.

Wenn das Fachverfahren auf Empfängerseite zweifellos erkennen kann, dass es die fachlichen Daten bereits erhalten hat und die erneute Belieferung ignorieren kann, dann könnte der Sender in diesem Fall den Sendeprozess einfach wiederholen.

Wenn das Fachverfahren auf Empfängerseite eine wiederholte Belieferung nicht zweifelsfrei erkennen kann oder wenn die fachlichen Daten einen Ausgabestand enthalten müssen und die Belieferung in einer streng aufsteigenden Sequenz gefordert ist (wie z.B. beim Meldewesen der gesetzlichen Krankenkassen oder der Rentenversicherung, wo als Ausgabestand eine laufende Datenummer verlangt ist), dann ist von einer wiederholten Sendung abzuraten, denn es ist nicht ausgeschlossen, dass der Empfänger die Nachricht dadurch doppelt erhalten würde.

Die Unsicherheit des Senders, wie er auf die Situation einer fehlenden eXtra-Response reagieren soll, kann mit der eXtra-Standardnachricht **RepeatResponse** oder alternativ **RepeatResponseRequest** behoben werden. Mit der eXtra-Standardnachricht **RepeatResponse** bzw. **RepeatResponseRequest** kann der Sender vom Empfänger exakt die damalige Antwort auf seinen eXtra-Request mit `scenario=request-with-acknowledgement` anfordern, die der Sender aus irgendwelchen Gründen nicht erhalten hat. Mit dieser Antwort der Empfängerseite kann der Sender jetzt auf die Fehlermeldung sofort automatisch, also ohne menschliche Interaktion (z.B. Anruf bei der Hotline), gezielt und korrekt reagieren (siehe auch [ÜMSG Kapitel 4]).

### Wiederholtes Senden bei einem Sendeprozess mit `scenario=request-with-response`

Annahme: Der Sender hat eine fachliche Nachricht mit einem eXTra-Request der Art `scenario=request-with-response` abgesendet und als Rückantwort keine eXTra-Response, sondern eine Fehlermeldung der DFÜ-Ebene, z.B. einen Timeout der http(s)-Instanz erhalten. Dann ist es für den Sender – siehe oben - nicht entscheidbar, welcher Zustand auf Empfängerseite vorliegt.

Die Frage ist nun, wie die Unsicherheit des Senders zu bewerten ist. Entscheidend hierfür ist, welche Bedeutung der eXTra-Request mit `scenario=request-with-response` hatte. Wurde mit diesem eXTra-Request eine Auskunftsfunktion des Fachverfahrens auf Empfängerseite angesprochen – also eine Funktion, die den Systemzustand des Fachverfahrens auf Empfängerseite nicht verändert – so kann der Sender den ursprünglichen eXTra-Request einfach wiederholen.

Soll jedoch mit diesem eXTra-Request eine Veränderung des Systemzustands des Fachverfahrens auf Empfängerseite durch eine Verarbeitung der übermittelten fachlichen Daten bewirkt werden, so kann es sein, dass – je nach Ausgestaltung des Fachverfahrens auf Empfängerseite – der Sender auf die nicht erhaltene Response nicht verzichten kann, wenn er weiterhin in jedem Fall korrekte eXTra-Requests senden will bzw. wenn er vermeiden will, dass eine Wiederholung des ursprünglichen eXTra-Request zu einem fehlerhaften Systemzustand des Fachverfahrens auf Empfängerseite führen würde. Für diesen Fall stehen die beiden eXTra-Standardnachrichten **RepeatResponse** bzw. alternativ **RepeatResponseRequest** zur Verfügung, mit der der Sender vom Empfänger exakt die damalige Antwort auf seinen eXTra-Request mit `scenario=request-with-response` anfordern kann (siehe auch [ÜMSG Kapitel 4]).

### Technische Fehler in den Transport-Instanzen auf Empfängerseite

Eine weitere Frage ergibt sich für den Fall, dass nach einem erfolgreichen Sendeprozess mittels `scenario=request-with-acknowledgement` (signalisiert mit einem positiven Acknowledgement in der eXTra-Response auf Empfängerseite) technische Fehler auftreten (z.B. beim Entschlüsseln oder Dekomprimieren einer eXTra-Lieferung/eXTra-Package/eXTra-Message), so dass das gewünschte Fachverfahren auf Empfängerseite die fachlichen Daten gar nicht erhält. Wie kann nun die Empfängerseite den Sender darüber informieren, dass die in der ursprünglichen eXTra-Response bereits abgegebene positive Empfangsbestätigung mittels Acknowledgement sich nachträglich leider als zu optimistisch herausgestellt hat, weil ein Fehler

aufgetreten ist, der die Weitergabe der fachlichen Daten an das Fachverfahren verhindert hat? Die Idee ist nun, dass die Empfängerseite die ursprüngliche Empfangsbestätigung - das ursprüngliche Acknowledgment - quasi korrigiert und dem Sender als sog. AcknowledgmentUpdate so zur Verfügung stellt, dass dieser den Bezug zum ursprünglichen Sendeprozess herstellen kann.

**Hinweis:** Dieses Fehlerszenario ist bei Sendeprozessen mittels `scenario=request-with-response` kein zusätzlich zu lösendes Fehlerbild, weil die gesendeten fachlichen Daten vom Fachverfahren verarbeitet werden sollen und deshalb alle Transportinstanzen auf Empfängerseite mit eingebunden sind - also auch diejenige, bei der der technische Fehler vor Übergabe an das Fachverfahren aufgetreten ist.

**Hinweis:** Das AcknowledgmentUpdate ist genauso wie ein Acknowledgment Bestandteil einer eXTra-Response mit leerem Body auf der untersten eXTra-Ebene. Die festgestellte Fehlerursache wird in den neu gebildeten `ResponseDetails` der untersten eXTra-Ebene, z.B. der `Package`-Ebene, erläutert. In den `ResponseDetails` sollten die Attribute `@highestweight` und `@weight` auf "ERROR" gesetzt werden, wobei die Elemente `Code` und `Text` die entsprechende Fehlernummer und den erläuternden Text enthalten sollten.

**Empfehlung:** In einem AcknowledgmentUpdate sollten alle diejenigen Fehlerursachen gemeldet werden können, die auf dem Weg der fachlichen Daten vom eXTra-Empfangssystem bis hin zur erfolgreichen Abgabe an das Fachverfahren auftreten können. Die Empfängerseite sollte deshalb so konstruiert sein, dass z.B. beim Betriebsmodell Sende-Holbetrieb der Sender nach einem erfolgreichen Sendeprozess mittels `scenario=request-with-acknowledgment` immer garantiert eine Rückmeldung erhält, entweder im positiven Fall eine Rückmeldung des Fachverfahrens über die Verarbeitung seiner fachlichen Daten oder im Fehlerfall eine Rückmeldung in Form eines AcknowledgmentUpdate.

Eine analoge Empfehlung gibt es auch für Sendeprozesse mit `scenario=fire-and-forget`, sofern Sender und Empfänger vereinbart haben, dass das Fachverfahren auf Empfängerseite zu jeder empfangenen fachlichen Nachricht auch eine zugehörige Rückmeldung erzeugt und entsprechend zeitversetzt zur Abholung bereitstellt. Der Bezug zur ursprünglichen Lieferung kann vorzugsweise auf zwei verschiedenen Wegen (Alternativen) hergestellt werden: Über die `RequestID` des eXTra-Request der ursprünglichen Lieferung (Alternative1) oder über die `ResponseID` der eXTra-Response zur ursprünglichen Lieferung (Alternative2).

### Alternative1: Bezug zur ursprünglichen Lieferung mittels RequestID

Voraussetzung für diese Alternative ist, dass die Senderseite die `RequestID` grundsätzlich eindeutig generiert, sodass jedem eXtra-Request eine eindeutige `RequestID` zugeordnet werden kann. Kann dies die Senderseite eines Datenübermittlungsverbundes nicht garantieren, so sollte die Alternative2 - Bezug über die `ResponseID` - gewählt werden.

Die eXtra-Response mit einem `AcknowledgementUpdate` ist strukturell genauso aufgebaut wie eine eXtra-Response auf einen Sendeprozess mit `Scenario=request-with-acknowledgement`. Im Idealfall (wenn zum Zeitpunkt des Fehlers, der zu einem `AcknowledgementUpdate` führte, die eXtra-Header der eXtra Response noch verfügbar sind), enthält das `AcknowledgementUpdate` alle Meta-Daten der ursprünglichen Lieferung, d.h. den eXtra-Request `TransportHeader` der ursprünglichen Lieferung mit den ursprünglichen `RequestDetails` (und damit mit der damaligen `RequestID`, dem `TimeStamp`) und – sofern die ursprüngliche Lieferung eXtra-Package(s)/ eXtra-Messages enthielt – diejenigen eXtra-Request `PackageHeader/ MessageHeader` der ursprünglichen Lieferung des eXtra-Package/ Message, bei dem der technische Fehler auftrat (mit den damaligen `RequestDetails`).

Im Zuge der Generierung des `AcknowledgementUpdate` wird/werden also die Meta-Daten -der ursprünglichen Lieferung herangezogen, jetzt aber um die neu gebildeten `ResponseDetails` ergänzt. In den `ResponseDetails` auf der untersten eXtra-Ebene wird die `ResponseID` neu vergeben, der aktuelle `TimeStamp` erzeugt und die eigentliche Fehlermeldung in das Element `Report` eingetragen, außerdem werden die Attribute `@highestweight` und `@weight` auf "ERROR" gesetzt. Zusätzlich könnte im Element `Originator` die Instanz eingetragen werden, bei der der technische Fehler auftrat.

Mit dieser idealtypischen Vorgehensweise ist eine perfekte Zuordnung des `AcknowledgementUpdate` zu der ursprünglichen Lieferung möglich.

In der Realität eines Datenübermittlungsverbundes sind möglicherweise zum Zeitpunkt des auslösenden Fehlers die kompletten Meta-Daten der eXtra-Header des ursprünglichen Sendeprozesses mit einem positiven `Acknowledgement` nicht mehr verfügbar, z.B. dann, wenn der logische Empfänger remote an das eXtra Empfangssystem angebunden ist und der Transfer vom physikalischen zum logischen Empfänger nicht mit eXtra-Mitteln durchgeführt wird. Je mehr Meta-Daten des ursprünglichen Sendeprozesses zum Fehlerzeitpunkt noch verfügbar sind, umso aussagekräftiger kann die neue eXtra-Nachricht mit einem `AcknowledgmentUpdate` gebildet werden. Im Minimum sind neben der `RequestID` die

SenderID der ursprünglichen Lieferung erforderlich. Von Vorteil ist es außerdem, wenn beim Sendeprozess das PlugIn `DataSource` mit dem Dateinamen der fachlichen Daten - dem `RequestFileName` - mitgegeben wurde und dieses beim `AcknowledgementUpdate` mit dazugestellt werden kann. Die anderen Informationen der eXtra-Header müssen notfalls neu gebildet werden.

### **Alternative2: Bezug zur ursprünglichen Lieferung mittels ResponseID**

Voraussetzung für diese Alternative ist, dass die Empfängerseite die `ResponseID` grundsätzlich eindeutig generiert, sodass jedem eXtra-Request/eXtra-Response eine eindeutige `ResponseID` zugeordnet werden kann. Weil die Voraussetzung einer eindeutigen `ResponseID` nur von einer Stelle - dem eXtra-Empfangssystem - erbracht werden muss, ist sie wesentlich leichter sicherzustellen als die bei der Alternative 1 geforderte Eineindeutigkeit der `RequestID` über alle Sender hinweg.

Im Fall eines zum `AcknowledgementUpdate` führenden Fehlers würde als `ResponseID` jetzt die `ResponseID` der ursprünglichen eXtra-Response wieder verwendet werden. Damit geht die Eineindeutigkeit der `ResponseID` verloren: Die `ResponseID` wird zum eindeutigen Identifikator nicht mehr nur eines einzigen eXtra-Prozesses, sondern der gesamten eXtra-Prozesskette, beginnend beim Sendeprozess über den Holprozess bis zum Bestätigungsprozess.

Die obigen Ausführungen zu den Meta-Daten und zur Bildung der eXtra-Header für das `AcknowledgementUpdate` sowohl in idealtypischer Weise wie auch im Minimalfall gelten in analoger Weise auch hier für die Alternative2.

**Hinweis:** Der Bedeutungswandel der `ResponseID` hin zu einem Identifikator der gesamten Prozesskette hat Einfluss auf den Bestätigungsprozess, sofern Sender und Empfänger vereinbart haben, dass der Sender jede abgeholte eXtra-Nachricht bestätigen muss. Falls die Bestätigung mittels **ConfirmationOfReceipt** bislang mit dem `ResponseFilename` erfolgt ist, dann sollte sie stattdessen jetzt ebenfalls mit der `ResponseID` erfolgen. Die Angabe der `ResponseID` ist dann erzwungen, wenn in der neuen eXtra-Nachricht des Typs `AcknowledgementUpdate` das ursprüngliche PlugIn `DataSource` mit dem `RequestFileName` nicht mitgegeben werden kann.

### Empfehlung:

Die Senderseite sollte genauso wie die Empfängerseite die `RequestID` respektive die `ResponseID` prinzipiell eindeutig über alle Empfänger, Fachverfahren und Datentypen hinweg vergeben. Dies könnte in Form einer  $\textcircled{1}$ UUID geschehen oder alternativ in streng aufsteigender zeitlicher Reihenfolge.

Wenn die Empfängerseite die `ResponseID` in Form einer UUID bildet, ist damit zwar die Eineindeutigkeit sichergestellt, aber keine zeitliche Reihenfolge verknüpft. Dies ist bei einer Anforderung mittels **ListRequest** oder **DataRequest** zu berücksichtigen, wenn in der Query mit einem Aufsetzpunkt „gib mir alles seit der letzten Abholung“ gearbeitet werden soll. Der Aufsetzpunkt sollte dann nicht über die `ResponseID`, sondern z.B. mit Hilfe des `ResponseCreationTimeStamp` gebildet werden, evtl. in Kombination mit `State=AVAILABLE`. Ist die `ResponseCreationTimeStamp` auf Grund der Profilierung nicht in der Query zulässig, so bleibt in diesem Fall nur die Möglichkeit, über die Standardnachricht **ListRequest** ein `AcknowledgementUpdate` zu erkennen.

Wenn die Empfängerseite die `ResponseID` in streng aufsteigender Reihenfolge vergibt, dann ist damit gewährleistet, dass die Sender bei der Alternative1 ein `AcknowledgementUpdate` auch immer erhalten werden, wenn sie den **DataRequest** mit einem Aufsetzpunkt der letzten abgeholten `ResponseID` formulieren, z.B. mit `ResponseID GT xyz`. Würde die `ResponseID` im `AcknowledgementUpdate` nicht neu gebildet, sondern wie bei der Alternative2 aus dem damaligen `Acknowledgement` der ursprünglichen eXtra-Response übernommen werden, würde damit in zeitlicher Hinsicht ein Schritt zurück in die Vergangenheit erfolgen. Um das Problem zu umgehen, dass der Sender das `AcknowledgementUpdate` nicht mehr erhalten würde, weil inzwischen die `ResponseID` der zuletzt abgeholten Rückmeldung größer ist als die `ResponseID` des damaligen `Acknowledgement`, hat der Sender zwei Möglichkeiten. Entweder formuliert er die **DataRequest** Anforderung ohne Angabe eines Aufsetzpunktes – also im Sinn eines „gib mir alles was Du hast“ - oder er nutzt die Auskunftsfunktion des **ListRequests** evtl. in Kombination mit `State=AVAILABLE`, um auch ein `AcknowledgementUpdate` zu erfassen.

Empfehlung: Die Problematik der Formulierung eines geeigneten Aufsetzpunktes mit der Standardnachricht **DataRequest** kann deutlich entschärft werden, wenn der neuen eXtra-Nachricht des Typs `AcknowledgementUpdate` ein eigener Datentyp zugeordnet wird, der in der Standardnachricht **DataRequest** in der Query als Kriterium (z.B. als `@Property="DataType"`

EQ=AcknowledgementUpdate) formuliert werden kann und der bei der Auslieferung im eXtra-ResponseHeader im Element `Datatype` zurückgegeben wird.

Wenn im Rahmen der Profilierung der Standardnachricht **DataRequest** in der Query als Kriterium neben `@Property="DataType"` auch `@Property="Procedure"` zulässig ist, dann könnte der Sender bei Bedarf jeweils nur die technischen Fehler abholen, die bei einem bestimmten Verfahren aufgetreten sind. Die Belastung der Empfängerseite durch Abfragen technischer Fehler könnte dadurch stark reduziert werden.

#### 4.2.2. Der Holprozess

##### Die Vorbereitung des Holprozesses mit der Auskunftsfunktion ListRequest – ListResponse

Ab der Version 1.4 der Schemadatei der eXtra-Standardnachrichten stehen die Standardnachrichten **ListRequest** und **ListResponse** zur Verfügung, die eine Auskunftsfunktion über den Hol- und Bestätigungsbetrieb der jüngsten Vergangenheit realisieren. Damit hat der Sender die Möglichkeit, sich über den Zustand des Auslieferungssystems der Empfängerseite (dem eXtra-Delivery-Server) zu erkundigen. Mit einem **ListResponse** erhält der Sender ein „Inhaltsverzeichnis“ der Rückmeldungen bzw. fachlichen Nachrichten, die der im **ListRequest** angegebenen Query entsprechen. In der Query kann der Sender den gewünschten Status der Rückmeldungen bzw. fachlichen Nachrichten im Datenbestand des eXtra-Delivery-Servers angeben:

- `State=AVAILABLE` bezeichnet die noch nicht abgeholten Rückmeldungen bzw. fachlichen Nachrichten
- `State=FETCHED` bezeichnet die bereits abgeholten, aber noch nicht bestätigten Rückmeldungen bzw. fachlichen Nachrichten
- `State=CONFIRMED` bezeichnet die bereits abgeholten und bestätigten Rückmeldungen bzw. fachlichen Nachrichten

##### Hinweis:

`State=Confirmed` macht nur dann Sinn, wenn die Vorhaltezeit von bereitgestellten eXtra-Nachrichten nicht mit der Bestätigung des erfolgreichen Abholens endet, sondern darüberhinaus erst n Tage später zu Ende geht oder eine generelle Konvention besteht, dass

die Vorhaltezeit prinzipiell n Wochen beträgt, innerhalb derer Daten unabhängig vom Abholstatus abgeholt werden können.

Mit dem so erworbenen „Inhaltsverzeichnis“ kann der Sender im Normalbetrieb bei einem nachfolgenden **DataRequest**-Aufruf beim Sende-Hol-Betriebsmodell ganz gezielt die verfügbaren und gewünschten (noch nicht abgeholt) Rückmeldungen angeben bzw. beim Hol-Betriebsmodell gezielt die gewünschten bereitgestellten (noch nicht abgeholt) fachlichen Nachrichten auswählen.

Will die Empfängerseite die Senderseite verpflichten, den Holprozess und den Bestätigungsprozess als einen integrierten Prozess auszugestalten (um das Problem von „Leichen“ im Datenbestand des eXTra-Delivery-Servers zu vermeiden, das dadurch entsteht, dass Sender zwar Rückmeldungen abholen, aber „vergessen“, zu bestätigen), dann ist dafür die Verfügbarkeit der Standardnachrichten **ListRequest** und **ListResponse** zusammen mit **DataRequest** der Version 1.3 Voraussetzung. Denn nur so hat der Sender die Chance, eine zwar abgeholte Rückmeldung bzw. fachliche Nachricht erneut abzuholen, falls aus irgendeinem Grund die so abgeholte Rückmeldung bzw. fachliche Nachricht nicht beim zugeordneten Fachverfahren auf Senderseite ankam.

Im Katastrophenfall mit Datenverlust kann der Sender mit den Angaben im **ListResponse** erneut die bereits abgeholt und ggf. bereits bestätigten Rückmeldungen oder fachlichen Nachrichten mit einem **DataRequest** der Version 1.3 abholen und damit seinen Datenbestand wiederherstellen (weitere Details siehe [ÜMSG] Kapitel 3.2) – sofern der Katastrophenfall noch innerhalb der vereinbarten Vorhaltezeit liegt.

### **Hinweis:**

Die beim **ListResponse** im Element `Entry` aufgelisteten Eigenschaften, z.B. `ReceiverID`, `Procedure`, `Datatype`, `ResponseFileName`, `ResponseCreationTimeStamp` sind diejenigen, die bei einem nachfolgenden **DataRequest** im eXTra-Package- bzw. eXTra-MessageHeader ausgeliefert werden.

Die Empfängerseite versucht beim **ListResponse** für jede Rückmeldung bzw. fachliche Nachricht das Maximum an verfügbaren Informationen zur Verfügung zu stellen.

**Hinweis:**

Wenn der Sender das „Inhaltsverzeichnis“ für Rückmeldungen bzw. fachlichen Nachrichten mittels **ListRequest** anfordert, die er noch nicht abgeholt hat, also mit Status AVAILABLE, dann wird es beim **ListResponse** im Element `Entry` zumindest keine `RequestID` geben können, weil es noch keinen entsprechenden **DataRequest**-Aufruf gab. Ob zu diesem Zeitpunkt bereits eine `ResponseID` existiert, ist vom Zusammenspiel zwischen Fachverfahren und eXTra Delivery Server auf Empfängerseite abhängig.

**Zusammenspiel des eXTra-Servers auf Empfängerseite mit dem verarbeitenden Fachverfahren**

Beim Betriebsmodell Sende-Holbetrieb ist das Zusammenspiel des eXTra-Servers auf Empfängerseite mit dem verarbeitenden Fachverfahren von besonderem Interesse. Im Zuge eines Sendeprozesses leitet das eXTra-Empfangssystem die empfangenen fachlichen Nachrichten an das verarbeitende Fachverfahren des logischen Empfängers weiter. Das Fachverfahren verarbeitet die so erhaltenen fachlichen Nachrichten und erzeugt die Verarbeitungsquittung (siehe 4.2), die der Sender später mit einem Holprozess, einem eXTra-Request mit einem **DataRequest**-Aufruf anfordern kann.

Für die Aufbewahrung der so erzeugten Verarbeitungsquittung bis zu deren Anforderung durch den Sender mittels **DataRequest** und der Bestätigung der erfolgreichen Abholung mittels **ConfirmationOfReceipt** gibt es zwei Alternativen:

## Alternative 1:

Das verarbeitende Fachverfahren bewahrt die erzeugte Verarbeitungsquittung zumindest bis zum Zeitpunkt der Bestätigung der erfolgreichen Abholung durch den Sender mittels **ConfirmationOfReceipt** bis auf Weiteres auf (wenn die Standardnachrichten **ListRequest** und **ListResponse** unterstützt werden, auch noch über den Zeitpunkt eines **ConfirmationOfReceipt** Aufrufs hinaus). Damit muss der Sender beim **DataRequest**-Aufruf das Fachverfahren als Auslieferungsinstanz im Element `Procedure` benennen und das eXTra System auf Empfängerseite muss einen **DataRequest** Aufruf immer an das so benannte Fachverfahren weiterreichen. Analoges gilt für den **ConfirmationOfReceipt** Aufruf. Das verarbeitende Fachverfahren muss somit unterschiedliche Verarbeitungsarten kennen: Zum einen die Verarbeitung von fachlichen Nachrichten und zum anderen die Datenhaltung von

Verarbeitungsquittungen, beginnend mit der Zwischenspeicherung über die Auslieferung bis hin zur Löschung in Folge eines **ConfirmationOfReceipt** Aufrufs.

Das eXTra-System auf Empfängerseite ist damit im Wesentlichen ein Vermittlungssystem zwischen der Senderseite und den Fachverfahren auf Empfängerseite. Die wesentliche Arbeit leistet das jeweilige Fachverfahren beim logischen Empfänger bzw. Endempfänger mit der Verarbeitung und der Datenhaltung der Verarbeitungsquittungen.

#### Alternative 2:

Im Unterschied zu Alternative 1 hat das verarbeitende Fachverfahren das Ergebnis der Verarbeitung, die Verarbeitungsquittung, sofort dem eXTra-System auf Empfängerseite zur Verfügung gestellt - quasi im Vorgriff auf die spätere Anforderung des ursprünglichen Senders mittels **DataRequest**. Es gibt also innerhalb des eXTra-Systems auf Empfängerseite noch eine weitere Instanz, die die Ergebnisse der Verarbeitungsläufe der verschiedenen Fachverfahren entgegennimmt und für eine spätere Anforderung durch den ursprünglichen Sender bereitstellt – das eXTra-Auslieferungssystem, den eXTra-Delivery-Server. Aus diesem Grund übergibt das eXTra-Empfangssystem die Anforderung in Form eines **DataRequest** an den eXTra-Delivery-Server, welcher die Anforderung bearbeitet und – im positiven Fall - die Verarbeitungsquittung ausliefert. Aus diesem Grund muss sich beim Sendeprozess das Element `Procedure` (der Name des Fachverfahrens) und beim späteren Holprozess mittels **DataRequest** das Element `Property @name=Procedure` (der Name des eXTra-Auslieferungssystems, der eXTra-Delivery Server) unterscheiden, weil damit andere Instanzen auf Empfängerseite adressiert werden.

Das eXTra-System auf Empfängerseite ist bei der Alternative 2 funktional deutlich mächtiger als bei Alternative1. Durch die Datenhaltung und Verwaltung der Rückmeldungen bzw. fachlichen Nachrichten aller angeschlossenen Fachverfahren auf Empfängerseite werden diese deutlich entlastet. Die auf Empfängerseite zu leistende Arbeit wird bei dieser Alternative auf die jeweiligen Fachverfahren bei den logischen Empfängern bzw. Endempfängern und dem zentralen eXTra-Delivery Server mit der Datenhaltung und Verwaltung aller Rückmeldungen und fachlichen Nachrichten aufgeteilt.

### **Der Holprozess: Abholtransaktion oder teilweise Abholung**

Beim Betriebsmodell Sende-Holbetrieb oder Holbetrieb stellt sich beim Holprozess für das Abholsystem auf Senderseite die Frage, ob hier im Zusammenspiel mit dem Auslieferungssystem auf Empfängerseite (dem eXTra Delivery Server) eine Abholtransaktion oder eine teilweise Abholung angemessen ist.

Die eXTra-Standardnachricht **ConfirmationOfReceipt** lässt zu, dass beim Holbetrieb das abholende System auf Senderseite nur die erfolgreich übernommenen fachlichen Daten an das Auslieferungssystem auf Empfängerseite (dem eXTra Delivery Server) zurückmeldet. Folgt man den Möglichkeiten der Standardnachricht **ConfirmationOfReceipt**, ist das Abholsystem auf Senderseite so ausgelegt, dass es keine Abholtransaktion realisiert hat, sondern eine teilweise Abholung. Ein zusätzliches eXTra-Sprachmittel, um die nicht erfolgreich abgeholten fachlichen Daten zu melden, erscheint unnötig, da der Sender diese Daten mit einem erneuten **DataRequest**-Aufruf erneut anfordern kann.

### **Hinweis:**

Im Zusammenspiel mit den beiden eXTra-Servern der gesetzlichen Krankenkassen (dem GKV KommServer) und der Datenstelle der Träger der Rentenversicherung DSRV kann der Sender das Betriebsmodell des Sende-, Hol- und Bestätigungsbetriebs mit teilweiser Abholung realisieren.

### **Parametrierung der Standardnachricht DataRequest**

### **Hinweis:**

Fehlt in der `Query` der Standardnachricht **DataRequest** die Angabe für `ReceiverID/ Procedure/ Datatype/ ResponseID/ ResponseFileName`, so wird dies jeweils als „keine Einschränkung“ interpretiert. Im Extremfall ist das Element `Query` leer. In diesem Fall erhält der Sender von allen mit dem eXTra-Auslieferungssystem (dem eXTra-Delivery Server) in Beziehung stehenden logischen Empfängern bzw. Endempfängern (alle vom eXTra-Delivery Server bedienten `ReceiverIDs`) alle für ihn bereitgestellten (und noch nicht als abgeholt bestätigten) fachlichen Nachrichten bzw. Daten. Umgangssprachlich formuliert sagt der Sender dem eXTra-Delivery Server auf Empfängerseite „gib mir alles was du hast“.

**Hinweis:**

Wieviele fachliche Nachrichten bzw. Rückmeldungen der Sender in der zugehörigen eXtra-Response nach einem **DataRequest** mit leerer *Query* erhält, ist für ihn prinzipiell nicht vorhersehbar. Er kann jedoch die maximale Anzahl auszuliefernder fachlichen Nachrichten oder Paketen bzw. die maximale Datengröße mit dem Element **Control** festlegen.

**Hinweis:**

Ab der Version 1.4 der Schemadatei der eXtra-Standardnachrichten kann der Sender das Problem der nicht vorhersehbaren Menge an Rückmeldungen umgehen, indem er vor dem eigentlichen Holprozess mittels **DataRequest** einen **ListRequest** Aufruf abgibt und mit der Antwort des **ListResponse** gezielt die gewünschten Rückmeldungen im **DataRequest** V1.3 angibt (siehe [ÜMSG] Kapitel 3.2).

Auch für die Empfängerseite bieten die beiden Standardnachrichten **ListRequest** und **ListResponse** Vorteile. Denn ein **DataRequest** mit leerer *Query*, also der Anforderung „gib mir alles was du hast“, könnte auf der Empfängerseite aufwändig zu bearbeiten sein. Insbesondere dann, wenn überwiegend solche Anforderungen mit „keine Daten vorhanden“ beantwortet werden, hat dies eine unerwünschte Systembelastung auf Empfängerseite zur Folge.

**Hinweis:**

Voraussetzung für die Nutzung der *ResponseID/ ResponseFilename* in der *Query* eines **DataRequest** für die Selektion einer Menge von fachlichen Nachrichten bzw. Daten mittels Vergleichsoperator GT, GE, LT, LE ist, dass der eXtra-Server auf Empfängerseite die Begriffe zumindest pro logischem Empfänger bzw. Endempfänger (*ReceiverID*) streng aufsteigend vergibt.

**Hinweis:**

Nutzt man in der *Query* als Selektionskriterium z.B. *@name="ResponseID"*, so ist damit nicht eindeutig festgelegt, ob damit die *ResponseID* der Transport-Ebene oder der Package-Ebene der eXtra-Response gemeint ist. Ursache hierfür ist die Auslieferung von fachlichen Nachrichten bzw. Rückmeldungen auf Grund eines **DataRequest** in Form von eXtra-Packages

oder eXtra-Messages (siehe oben Hinweis zu **DataRequest**). Mit dem Selektionskriterium `@name="Layer"` kann die erforderliche Eindeutigkeit erzielt werden.

### **Antwort der Empfängerseite (eXtra-Response) auf einen DataRequest**

#### **Klarstellung:**

Eine eXtra-Response als Antwort auf eine Anforderung einer Rückmeldung bzw. einer fachlichen Nachricht (mittels eXtra-Request und **DataRequest**) mit `Result=ok` in den `ResponseDetails` heißt lediglich, dass das eXtra-Auslieferungssystem (der eXtra-Delivery Server) diese Anforderung korrekt erfüllen und die geforderte Rückmeldung bzw. fachliche Nachricht ausliefern konnte. `Result=ok` bedeutet nicht notwendigerweise, dass das Fachverfahren auf Empfängerseite alle ursprünglich gesendeten fachlichen Daten annehmen und korrekt verarbeiten konnte. Eine Rückmeldung, die mit `Result=ok` ausgeliefert wurde, könnte auch in den fachlichen Daten (im eXtra-Body) Fehlermeldungen des Fachverfahrens enthalten.

### **Gestaltung des PackageHeader in der eXtra-Response auf einen DataRequest**

Das Ausgangsproblem sei wie folgt: Die Anforderung mittels **DataRequest** erfolgt nur auf der Transport-Ebene eines eXtra-Requests – die Auslieferung=eXtra-Response der bereitgestellten Informationen, z.B. von Rückmeldungen, aber auf der Package-Ebene. Insofern stellt sich die Frage, welche Werte die verschiedenen Elemente im Response `PackageHeader` erhalten sollen. Dies gilt im Einzelnen für die Elemente `ReceiverID` sowie `RequestID`, `Procedure` und `DataType` in den `RequestDetails`.

Das Element `ReceiverID` im `PackageHeader` erhält nicht die `ReceiverID` im `TransportHeader` der eXtra-Response, sondern die `ReceiverID` des logischen Empfängers bzw. Endempfängers, dessen Fachverfahren die jeweilige Rückmeldung erzeugt hat.

Das Element `RequestID` in den `RequestDetails` im `PackageHeader` erhält die `RequestID` aus dem `TransportHeader` der eXtra-Response (die wiederum identisch ist zur `RequestID` im `TransportHeader` vom auslösenden eXtra-Request).

Bei den Elementen `Procedure` und `DataType` in den `RequestDetails` gibt es mehrere Varianten, welche Werte sie annehmen können:

- Variante 1: Das Element `Procedure` enthält den Namen des Fachverfahrens auf Empfängerseite, das die Rückmeldung tatsächlich erzeugt hat, und das Element `DataType` den Namen der Rückmeldung, den das Fachverfahren auf Empfängerseite vergeben hat  
=> empfehlenswerte Variante,
- Variante 2: Kopie aus der Anforderung mittels **DataRequest** und der dortigen `Query` mit `Procedure` und `DataType`  
=> nahezu so gut wie Variante 1 - aber nur, falls in der Anforderung mittels **DataRequest** die `Query` tatsächlich auch so differenziert spezifiziert wurde
- Variante 3: `Procedure` und `DataType` bleiben leer

### **Empfehlung:**

Empfehlenswert ist die Variante 1, weil das eXtra-Abholssystem auf Senderseite das erzeugende Fachverfahren und den Namen der Rückmeldung unabhängig von der jeweiligen Spezifikationstiefe beim **DataRequest** immer erfährt (unabhängig davon, ob beim **DataRequest** in der `Query` `Procedure` und/oder `DataType` leer oder angegeben war), ohne die erhaltenen fachlichen Daten auswerten zu müssen. Damit kann das eXtra-Abholssystem immer unmittelbar entscheiden, an welches zugeordnete Fachverfahren auf Senderseite die Rückmeldung weitergeleitet werden soll.

Variante 2 ist nur dann empfehlenswert, wenn sichergestellt werden kann, dass in der Anforderung mittels **DataRequest** die `Query` tatsächlich auch immer `Procedure` und `DataType` enthält.

Variante 3 wird nicht empfohlen, weil das eXtra-Abholssystem auf Senderseite nicht unmittelbar entscheiden kann, welche Rückmeldung es erhalten hat und an welches Fachverfahren die Rückmeldung weitergeleitet werden soll.

### Wiederholtes Abholen einer erfolgreich abgeholten und evtl. bereits bestätigten

#### Rückmeldung bzw. fachlichen Nachricht

Solange der Sender die Abholung von Rückmeldungen oder fachlichen Nachrichten mittels **ConfirmationOfReceipt** nicht bestätigt hat, kann der Sender beliebig oft den Holprozess mit einem **DataRequest** Aufruf wiederholen.

Hat der Sender die Abholung allerdings bereits bestätigt, benötigt aber die bereits bestätigten Rückmeldungen bzw. fachlichen Nachrichten erneut (weil z.B. aus irgendeinem Grund die bereits abgeholte Rückmeldung bzw. fachliche Nachricht nicht beim zugeordneten Fachverfahren auf Senderseite ankam), so stehen ihm ab der Version 1.4 der Schemadatei der eXtra-Standardnachrichten die Standardnachrichten **ListRequest** Version 1.0 (mit einer `Query` und `State=CONFIRMED`), **ListResponse** Version 1..0 (die das entsprechende „Inhaltsverzeichnis“ ausliefert) und **DataRequest** Version 1.3 (mit einer `Query` und `State=CONFIRMED`, sowie einer Liste der gemäß „Inhaltsverzeichnis“ verfügbaren und gewünschten Rückmeldungen bzw. fachliche Nachrichten) zur Verfügung (siehe [ÜMSG Kapitel 3.2]).

#### **4.2.3. Der Bestätigungsprozess**

Der Bestätigungsprozess verfolgt primär zwei Ziele:

- Zum einen kann damit beim Betriebsmodell Sende- Hol- Bestätigungsbetrieb oder Hol- Bestätigungsbetrieb die Empfängerseite eine effiziente und ökonomische Datenhaltung des Auslieferungssystems (des eXtra-Delivery-Servers) umsetzen, weil der eXtra-Delivery-Server die bestätigten Rückmeldungen bzw. fachlichen Nachrichten aus seinem Datenbestand löschen kann.
- Zum anderen kann damit die Empfängerseite einen evtl. vom Gesetzgeber geforderten Nachweis der Auslieferung von Rückmeldungen an den Sender bzw. den Nachweis erbringen, dass der Sender die bereitgestellten Rückmeldungen auch erhalten hat.

#### Transaktionssicherheit

Darüberhinaus kann mit Hilfe des Hol- und Bestätigungsprozesses das eXtra-Gesamtsystem – bestehend aus eXtra-Client und eXtra-Server – den Fachverfahren auf beiden Seiten die sprachlichen Mittel für ein hohes Maß an Prozesssicherheit bieten, das einer Transaktionssicherheit entspricht. Dies gilt insbesondere für asynchron arbeitende

Fachverfahren beim Betriebsmodell Sende- Hol-Bestätigungsbetrieb, für die der Nachweis der Termintreue, der Vollständigkeit und Korrektheit wichtig ist.

Das erzeugende Fachverfahren auf Senderseite kann nachweisen,

- wann die Empfängerseite die fachlichen Nachrichten in ihre Sphäre übernommen hat,
- dass es einer ggf. bestehenden Meldepflicht rechtzeitig und vollständig nachkam und
- dass das konsumierende Fachverfahren auf Empfängerseite alle gesendeten fachlichen Nachrichten bearbeitet hat.

Das konsumierende Fachverfahren auf Empfängerseite wiederum kann nachweisen,

- dass es alle erhaltenen fachlichen Nachrichten bearbeitet hat,
- dass es das dazugehörige Verarbeitungsergebnis der Senderseite zur Verfügung gestellt hat und die Senderseite das Verarbeitungsergebnis nachweislich erhalten hat.

### Automatische Bestätigung

Will die Empfängerseite die Senderseite verpflichten, den Holprozess zusammen mit dem Bestätigungsprozess als einen integrierten Prozess auszugestalten, also die Bestätigung mittels **ConfirmationOfReceipt** im Zuge des Holprozesses sofort und automatisch zu generieren (um das Problem von „Leichen“ im Datenbestand des eXtra Delivery Server zu vermeiden, das dadurch entsteht, dass Sender zwar Rückmeldungen abholen, aber „vergessen“ zu bestätigen), dann ist dafür die Verfügbarkeit der Standardnachrichten **ListRequest** und **ListResponse** zusammen mit **DataRequest** der Version 1.3 Voraussetzung. Denn nur so hat der Sender die Chance, eine abgeholte Rückmeldung bzw. fachliche Nachricht erneut abzuholen, falls aus irgendeinem Grund die so abgeholte Rückmeldung bzw. fachliche Nachricht nicht beim zugeordneten Fachverfahren auf Senderseite ankam.

### Hinweis:

Hat der Sender beim Holprozess mittels **DataRequest** Rückmeldungen von mehreren logischen Empfängern/ Endempfängern erhalten, sollte er beim anschließenden Bestätigungsprozess mittels **ConfirmationOfReceipt** berücksichtigen, dass der `ResponseFileName` bzw. die `ResponseID idR` nur eindeutig innerhalb eines logischen Empfängers/ Endempfängers ist. Deshalb sollte der Sender bei der Standardnachricht **ConfirmationOfReceipt** zusätzlich die jeweiligen `ReceiverIDs` des logischen Empfängers/ Endempfängers mit angeben, um eine eindeutige Zuordnung von `ResponseFileName` bzw. `ResponseID` zu logischem Empfänger/ Endempfänger zu gewährleisten.

**Hinweis:**

Hat der Sender beim Holprozess mittels **DataRequest** Rückmeldungen von mehreren logischen Empfängern/ Endempfängern erhalten und will er mit einem Prozess alle abgeholten Rückmeldungen explizit bestätigen, so steht für eine derartige „Sammelbestätigung“ die Standardnachricht **ListOfConfirmationOfReceipt** mit mehreren **ConfirmationOfReceipt**-Nachrichten zur Verfügung.

**Hinweis:**

konnten bei einem Abholprozess alle fachlichen Daten, Packages oder Messages erfolgreich übernommen werden, so bietet die Standardnachricht **ConfirmationOfReceipt** eine einfache und sehr effiziente Möglichkeit, den gesamten Abholprozess mit einer einzigen Angabe zu bestätigen. Dies gelingt mit dem Element `Property`, dem Attribut `@name="RequestID"` oder „`ResponseID`“ und dem Attribut `@layer="Transport"`, wobei die `RequestID` oder `ResponseID` demjenigen `TransportHeader` zu entnehmen ist, der bei der Auslieferung der angeforderten Rückmeldungen bzw. fachlichen Nachrichten durch das eXtra-Auslieferungssystem (den eXtra-Delivery-Server) auf Empfängerseite mittels eXtra-Response zurückgeliefert wurde.

**4.2.4. Nachvollzug und Recherche**

Die beiden eXtra-Elemente `RequestID` und `ResponseID` im eXtra Header dienen dazu, jedem eXtra-spezifischen Vorgang einen eindeutigen Identifikator zuzuordnen. Der Sender vergibt die `RequestID` in seiner Begriffswelt, der Empfänger die `ResponseID` ebenfalls aus seiner Begriffswelt. Damit haben die `RequestID` und `ResponseID` die Bedeutung eines Tickets, mit dem jeder einzelne eXtra-Prozessschritt - sowohl beim Sender als auch beim Empfänger - effizient nachvollzogen werden kann.

**Empfehlung:**

`RequestID` und `ResponseID` sollten eindeutig vergeben werden, am besten über alle logischen Sender/ logischen Empfänger, Fachverfahren und Datentypen hinweg.

#### Alternative 1: Verwendung von UUIDs

UUIDs sind laut Definition eindeutig. Allerdings ist hiermit kein Zeitbezug gegeben. Dies ist bei Verwendung der Standardnachrichten **ListRequest** und **DataRequest** dann zu berücksichtigen, wenn mit einem Aufsetzpunkt „gib mir alles seit dem letzten Abholprozess“ gearbeitet wird.

#### Alternative 2: Verwendung von streng aufsteigenden Informationen

Um das Problem des Überlaufs zu umgehen, sollte der Begriff genügend Stellen umfassen. Ein probates Mittel ist die Integration des Zeitstempels, z.B. könnte der Begriff 20-stellig sein und folgendermaßen aufgebaut sein: JJJJMMTTHHMMSSnnnnnn mit einer 6-stelligen laufenden Nummer nach dem Zeitstempel.

Sowohl für die Sender- als auch die Empfängerseite ist es von Vorteil, wenn die `RequestID` und `ResponseID` jeweils über alle logischen Sender/ logischen Empfänger/ Endempfänger, Fachverfahren und Datentypen eindeutig vergeben wird, weil damit auch die zeitliche Sequenz aller eXtra-spezifischen Prozesse abgebildet werden kann..

#### **Empfehlung:**

Die beiden eXtra-(Schlüssel-)Begriffe `RequestID` und `ResponseID` sollten beide konsequent in den Systemen des Senders wie auch des Empfängers abgespeichert werden, um für alle Prozesse einen Nachvollzug der gesamten Bearbeitungs- und Prozesskette gewährleisten zu können, so wie es Zielsetzung der beiden eXtra-Standardnachrichten **StatusRequest/ StatusResponse** ist.

Im Idealfall bleiben diese beiden Begriffe `RequestID` und `ResponseID` auf dem gesamten Weg der Daten erhalten, vom eXtra-Empfangssystem über das verarbeitende Fachverfahren bis hin zum eXtra-Delivery-Server bei der Auslieferung von Rückmeldungen des Fachverfahrens. Dies bedingt allerdings in der Regel entsprechende Anpassungen all der Instanzen, die in den Weg der Daten vom eXtra-Empfangssystem bis hin zum eXtra-Delivery-Server involviert sind: Dies dürfte erst als Endergebnis eines längeren Entwicklungsprozesses erreichbar sein.

#### 4.2.4.1. Die Auskunftsfunktionen **StatusRequest** und **StatusResponse** sowie **ListRequest** und **ListResponse**

##### **StatusRequest – StatusResponse**

Mit den beiden eXtra-Standardnachrichten **StatusRequest** und **StatusResponse** werden eXtra-spezifische Sprachmittel zur Verfügung gestellt, mit denen Recherchen des Sendebetriebs der jüngsten Vergangenheit ermöglicht werden. Damit sollen Fragestellungen wie z.B. „was ist mit meiner Sendung mit der RequestID/ResponseID =xyz vor 3 Wochen geworden, weil dafür die Rückmeldung überfällig ist“ technisch formuliert werden können.

Analoges gilt für eine effektive Unterstützung für den Nachvollzug und die Diagnose insbesondere seltener, ungeklärter oder rätselhafter Vorgänge mit der Angabe der Stelle, an der z.B. die Verarbeitungskette beim Empfänger aus welchem Grund abbrach. Im Ergebnis bewirken die beiden Standardnachrichten eine deutliche Reduzierung des Service-Aufwands für den Sender wie auch für den Empfänger und dessen Hotline.

Wenn der Sender auf Grund eines technischen Defekts oder einer sonstigen Katastrophe einen Datenverlust erlitten hat und nun auf einen Sicherungsbestand vor n Tagen zurückgreifen muss, dann kann er mit den beiden Standardnachrichten in Erfahrung bringen, was sich im Sendebetrieb seit seiner letzten Datensicherung alles ergeben hat. **StatusRequest** und **StatusResponse** realisieren somit eine Auskunftsfunktion über den Sendebetrieb. Die ergänzende Auskunftsfunktion über den Hol- und Bestätigungsbetrieb wird über die beiden Standardnachrichten **ListRequest** und **ListResponse** zur Verfügung gestellt.

In der Praxis wird die Empfängerseite die Auskunftsfähigkeit zeitlich begrenzen, z.B. auf sechs Wochen. Da derartige Auskünfte evtl. nur sehr aufwändig erbracht werden können und andererseits bei verstärkter Nutzung einer DDos-Attacke gleichkommen können, ist in Erwägung zu ziehen, ob hierfür ein anderer Dienstendepunkt angeboten wird, an dem eine sog. Whitelist gepflegt wird.

##### **Hinweis:**

Bei der Formulierung der `Query` eines **StatusRequest** ist es ratsam, Begriffe aus der Sphäre des Senders zu verwenden, z.B. die `RequestID` oder den `FileName` im `Plugin DataSource`,

weil im Allgemeinen unbekannt ist, ob und wie weit die Bearbeitung der ursprünglichen Lieferung in den Systemen der Empfängerseite vorangeschritten ist. Dies herauszufinden ist ja gerade die Motivation für die Abgabe eines **StatusRequest**.

### **StatusResponse – Antwort auf einen StatusRequest**

Der Empfänger meldet als Antwort auf einen **StatusRequest** den zuletzt erreichten Zustand der betroffenen Sendungen zurück. Je nachdem wie die Empfängerseite ausgestaltet ist, kann der zuletzt erreichte Zustand durch

- a) das eXTra-Empfangssystem (den eXTra-Distribution-Server) im Minimalausbau => „Annahme und Weiterleitung ist erfolgt“,
- b) das Fachverfahren => „Verarbeitung ist erfolgt“ oder
- c) sogar durch das eXTra-Auslieferungssystem (den eXTra-Delivery-Server) im Endausbau => „Bereitstellung der Rückmeldungen, der Verarbeitungsergebnisse ist erfolgt“

gemeldet werden.

Im Fall a) im Minimalausbau, wenn nur der eXTra-Distributions-Server den zuletzt erreichten Zustand melden kann, lautet im fehlerfreien Normalfall die Antwort sinngemäß „eXTra-Distribution-Server: Weitergabe an das Fachverfahren erfolgreich durchgeführt“. Im Fehlerfall könnte die Antwort z.B. „eXTra-Distribution-Server: Fehler beim Versuch der Weitergabe an das Fachverfahren“ lauten.

Im Fall b), wenn das Fachverfahren in die Rückmeldung zu einem **StatusRequest** integriert ist, lautet im fehlerfreien Normalfall die Antwort sinngemäß z.B. „Fachverfahren: Daten erfolgreich erhalten“ oder „Fachverfahren: Daten erfolgreich verarbeitet“. Im Fehlerfall könnte die Antwort z.B. „Fachverfahren: Fehler beim Verarbeiten der Daten“ lauten.

Im Fall c), wenn der Weg der gesendeten fachlichen Daten über das Fachverfahren bis hin zum eXTra Delivery-Server verfolgt werden kann, lautet im fehlerfreien Normalfall die Antwort sinngemäß z.B. „eXTra Delivery-Server: Rückmeldung bzw. fachliche Nachricht vom Fachverfahren xyz erfolgreich erhalten“ (das entspricht dem Zustand `State=AVAILABLE` bei der Standardnachricht **ListRequest** oder **ListResponse**) oder „eXTra Delivery-Server: Rückmeldung bzw. fachliche Nachricht erfolgreich ausgeliefert“ (das entspricht dem Zustand `State=FETCHED` bei der Standardnachricht **ListRequest** oder **ListResponse**) oder „eXTra

Delivery-Server: Bestätigung für erfolgreiche Abholung der Rückmeldung bzw. fachlichen Nachricht erhalten“ (das entspricht dem Zustand `State="CONFIRMED"` bei der Standardnachricht **ListRequest** oder **ListResponse**). Im Fehlerfall könnte die Antwort z.B. lauten: „eXTra-Delivery-Server: letzte Bestätigung mittels **ConfirmationOfReceipt** und `RequestID xyz` fehlerhaft“.

### **4.3. Zusammenspiel der beiden Fachverfahren auf Sender- und Empfängerseite**

#### **Allgemeines zu Rückmeldungen und Verarbeitungsquittungen**

##### **Hinweis:**

das Format und die Struktur einer Rückmeldung obliegen allein dem jeweiligen Fachverfahren, da die Rückmeldung eine fachliche Nachricht ist.

##### **Klarstellung:**

Eine Rückmeldung, sei es eine Zustell- oder eine Verarbeitungsquittung, stammt ursächlich immer von einem Fachverfahren – nie von einer eXTra-Instanz.

Erzeugt ein (älteres) Fachverfahren im Fehlerfall lediglich einen Fehlercode, dann ist es denkbar, dass eine eXTra-Instanz aus dem Fehlercode eine verständliche Fehlermeldung bildet.

#### **Die semantische Qualität einer Rückmeldung**

Bei allen Betriebsmodellen, bei denen die Empfängerseite Rückmeldungen zur Verfügung stellt, ist für das Fachverfahren auf Senderseite entscheidend, welche semantische Qualität die Rückmeldung beinhaltet: Ist sie als Zustellquittung oder als Verarbeitungsquittung zu verstehen?

Zustellquittung heißt, dass das Fachverfahren auf Empfängerseite den Erhalt der fachlichen Daten bestätigt bzw. die Annahme einiger/aller fachlichen Daten verweigert. Eine Aussage bezüglich der korrekten oder fehlerhaften Verarbeitung der fachlichen Daten ist damit nicht enthalten. In qualitativer Hinsicht ist also eine Zustellquittung nur geringfügig besser als ein Acknowledgement beim Sendeprozess.

Mit einer Verarbeitungsquittung bestätigt das Fachverfahren auf Empfängerseite nicht nur den Erhalt der fachlichen Daten, sondern teilt auch mit, ob es die fachlichen Daten verarbeiten konnte bzw. welche Fehler an welcher Stelle aufgetreten sind. Eine Verarbeitungsquittung ist also qualitativ deutlich besser als eine Zustellquittung.

**Empfehlung:**

Auf Grund der deutlich höheren Aussagekraft wird empfohlen, dass das Fachverfahren eine Verarbeitungsquittung und nicht nur eine Zustellquittung erzeugt.

**Empfehlung:**

Um für beide Seiten eine möglichst hohe Prozesssicherheit zu gewährleisten, wird empfohlen, dass das verarbeitende Fachverfahren auf Empfängerseite prinzipiell immer eine Rückmeldung, eine Verarbeitungsquittung erzeugt.

**Empfehlung:**

Besteht eine Lieferung aus mehreren eXTra-Ebenen und enthält sie mehrere Packages und/oder Messages, so erzeugen die verarbeitenden Fachverfahren auf Empfängerseite für jede eXTra-Instanz der untersten Ebene (für jedes eXTra-Package bzw. eXTra-Message) jeweils (mindestens) eine Rückmeldung.

Von einer Zusammenfassung der Verarbeitungsergebnisse mehrerer eXTra Packages oder eXTra-Messages zu einer einzigen Rückmeldung wird abgeraten, denn dann wäre eine Zuordnung einer Rückmeldung zu genau einer gesendeten Lieferung/eines eXTra-Packages/einer eXTra-Message nicht möglich.

Enthält eine Lieferung z.B. fünf eXTra-Packages oder zehn eXTra-Messages, so erzeugt das oder die unterschiedlichen Fachverfahren (mindestens) fünf Rückmeldungen auf Package-Ebene bzw. (mindestens) zehn Rückmeldungen auf Message-Ebene.

Damit ist sichergestellt, dass in der eXTra-Response auf einen **DataRequest** mehrere Rückmeldungen in Form von eXTra-Messages oder eXTra-Packages ausgeliefert werden können.

## 5. Das eXTra-Gesamtsystem: Verbindliche Festlegungen, Betriebsparameter und Leistungsumfang

In diesem Dokument wird an mehreren Stellen die Empfehlung ausgesprochen, die Bedingungen, denen das eXTra-Gesamtsystem eines Datenübermittlungsverbundes genügt, verbindlich zu regeln bzw. die Belegung verschiedener Betriebsparameter bekannt zu geben. Neben diesen Empfehlungen werden wichtige Merkmale des eXTra-Gesamtsystems in der folgenden Übersicht zusammengefasst.

### Betriebsparameter gleichermaßen für die unterlagerten DFÜ-Systeme und die eXTra-Instanzen

- Zeitspanne (Wert des Timeout), innerhalb der eine eXTra-Response mit `Scenario=Request-with-Acknowledgement` erfolgen muss. Implizit wird damit die maximale Größe einer Lieferung bestimmt.
- Zeitspanne (Wert des Timeout), innerhalb der eine eXTra-Response mit `Scenario=Request-with-Response` erfolgen muss. Implizit wird damit die maximale Größe einer Auslieferung bzw. der maximale Umfang/ maximale Detaillierung von Auskunftsfunktionen bestimmt.

### eXTra-spezifische Festlegungen

- eXTra Empfangssystem: Unterstützung der ExtraError-Nachricht?
- eXTra-Empfangssystem: Unterstützung der MTOM-Funktionalität?
- Maßgeblicher Zeitstempel: Welche Uhr ist maßgeblich, wenn die Uhren des Senders und Empfängers auseinanderlaufen?
- Sendebetrieb: Festlegung der maximalen Größe einer Lieferung mit allen eXTra-Objekten und gegebenenfalls der maximalen Größe eines einzelnen eXTra-Objektes (Package oder Message)
- Holbetrieb: Festlegung der maximalen Größe einer Auslieferung mit allen eXTra-Objekten und gegebenenfalls der maximalen Größe eines einzelnen eXTra-Objektes (Package oder Message)

- Holbetrieb: Wie gezielt kann der Sender eXtra-Nachrichten mit der Standardnachricht **DataRequest** abholen? Bietet die Empfängerseite nur ein "gib mir alles was du hast" oder auch die Möglichkeit, gezielt genau eine spezifizierte eXtra-Nachricht abzuholen?
- Bedeutung eines Acknowledgements beim Sendeprozess: Welche Prozessschritte im eXtra-Empfangssystem werden mit einem Acknowledgement bestätigt: Übernahme in die lokale Datenhaltung, Verifikation der eXtra-Nachricht, Entschlüsselung, Dekomprimierung, Weitergabe an das Fachverfahren?
- Unterstützung des sog. AcknowledgementUpdate? Bezug zur ursprünglichen Lieferung mittels RequestID oder ResponseID der ursprünglichen Lieferung? Zusätzlich Bekanntgabe, wie die eXtra-Header des AcknowledgementUpdates gebildet werden und wie die Sender ein AcknowledgementUpdate erkennen und abholen können. Wird einem AcknowledgementUpdate ein eigenständiger Datentyp zugewiesen, sodass der Sender eine eXtra-Nachricht des Typs AcknowledgementUpdate mittels Standardnachricht **DataRequest** gezielt abholen kann?
- Zulässiges Maß von parallelen Sendeprozessen: Für alle logischen Empfänger bzw. für alle Fachverfahren und gegebenenfalls für genau einen logischen Empfänger oder genau ein Fachverfahren?  
Extremfall: keine Parallelität, ausschließlich serieller Betrieb.
- Zulässiges Maß von parallelen Holprozessen: Von allen logischen Empfängern bzw. von allen Fachverfahren und gegebenenfalls von genau einem logischen Empfänger oder genau einem Fachverfahren?  
Extremfall: keine Parallelität, ausschließlich serieller Betrieb.
- eXtra-Empfangssystem: Realisierung des Sendebetriebs als Annahmetransaktion oder als teilweise Annahme?
- Sendebetrieb: Toleranzzeit, innerhalb derer Lücken in der Belieferung vom Sender geschlossen werden müssen (z.B. bei laufender Dateinummer)
- Unterstützung der Standardnachricht **ConfirmationOfReceipt**: Wünscht bzw. verlangt die Empfängerseite die Bestätigung erfolgreich abgeholter eXtra-Nachrichten?
- Vorhaltezeitraum für bereitgestellte fachliche Nachrichten, innerhalb derer der Sender diese abholen muss bzw. nach dem die fachlichen Nachrichten gelöscht werden.
- Unterstützung der Standardnachrichten **ListRequest** und **ListResponse**: Kann der Sender bereits abgeholte und evtl. bereits bestätigte fachliche Nachrichten erneut abholen?

- Vorhaltezeitraum für bereits abgeholte und bestätigte fachliche Nachrichten, nach dem die fachlichen Nachrichten gelöscht werden (relevant für die Standardnachrichten **ListRequest/ListResponse** mit folgendem **DataRequest**).
- Unterstützung der Standardnachricht **StatusRequest**: Wie detailliert kann die **StatusRequest**-Query sein?
- Unterstützung der Standardnachricht **StatusResponse**: Bis zu welcher Instanz auf Empfängerseite kann der Status einer gesendeten eXTra-Nachricht nachverfolgt werden? Nur innerhalb des eXTra-Empfangssystems (eXTra-Distribution-Server) oder bis einschließlich zum Fachverfahren oder darüber hinaus sogar bis zur Auslieferung der bereitgestellten Rückmeldung (eXTra-Delivery-Server)?
- Sendebetrieb: Unterstützung der Standardnachricht **RepeatResponse** bzw. **RepeatResponseRequest**: Kann der Sender eine ausgebliebene eXTra-Response anfordern? Kann dies mit **RepeatResponse** geschehen, oder mit **RepeatResponseRequest**? Wie lange nach dem ursprünglichen Sendevorgang kann der Sender eine ausgebliebene eXTra-Response mit einer dieser beiden Standardnachrichten anfordern?

## 6. Anhang

### 6.1. Referenzen

Kurzname	Quelle
ANWLF	<i>eXtra Anwenderleitfaden mit Beispielen aus der Praxis</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
BEST	<i>eXtra Best Practices</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
DSIG	<i>eXtra Design Guidelines</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
EINF	<i>Einführung in den eXtra Standard</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
EMSG	<i>eXtra Standardnachrichten, Schnittstellenbeschreibung</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
EXSEC	<i>Sicherheit und Verfügbarkeit in einem eXtra spezifischen Datenübermittlungsverbund</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
EXWS	<i>eXtra und Webservices</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
FTPVSHHTTP	<i>ftp vs http</i> , Daniel Haxx, zu finden unter <a href="http://daniel.haxx.se/docs/ftp-vs-http.html">http://daniel.haxx.se/docs/ftp-vs-http.html</a>
ISO29115-11	<i>Entity authentication assurance framework</i> , ISO/IEC 29115, ursprüngliche Fassung November 2011, zu finden unter <a href="https://www.oasis-open.org/committees/download.php/44751/285-17Attach1.pdf">https://www.oasis-open.org/committees/download.php/44751/285-17Attach1.pdf</a>  aktuelle Fassung vom April 2013, zu finden unter <a href="https://www.iso.org/obp/ui/#iso:std:45138:en">https://www.iso.org/obp/ui/#iso:std:45138:en</a>
IFACE	<i>eXtra Transport Schnittstellenbeschreibung</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
KOMP	<i>eXtra Kompendium</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
MTAB	<i>eXtra Mustertabellen</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
NIST_SP800 63	<i>Electronic Authentication Guideline</i> , National Institute of Standards and Technology, USA NIST Special Publication 800-63-2, 26. August 2013: überarbeitete Fassung, zu finden unter <a href="http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf">http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf</a>  NIST Special Publication 800-63 Version 1.0.2, April 2006: ursprüngliche

	Fassung zu finden unter <a href="http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf">http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf</a>
RFC2119	<i>Request for Comments: 2119</i> , S. Bradner, Harvard University, March 1997, <a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a>
PROF	<i>eXTra Profilierung</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
UMSG	<i>eXTra Standardnachrichten, Überblick</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
VERS	<i>eXTra Versionierung</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
VMRVSPoC	<i>RV-Vorgehensmodell, Anbindung von Fachverfahren an den SPoC</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
VMXUV	Drei Dokumente, die das Vorgehensmodell des XUV-Standards zeigen: <i>XUV_Standard_1.0_Begleitdokument.pdf</i> <i>Webservices.pptx</i> <i>Nachrichtenaustausch.pptx</i> zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
XBSP13	<i>XML-Beispiele auf Basis von eXTra V1.3</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
XBSP14	<i>XML-Beispiele auf Basis von eXTra V1.4</i> , zu finden unter <a href="http://www.extra-standard.de">http://www.extra-standard.de</a>
XENC	<i>XML Encryption</i> , <a href="http://www.w3.org/TR/xmlenc-core/">http://www.w3.org/TR/xmlenc-core/</a>
XML	<i>XML Recommendation 1.0, 3rd Edition</i> , <a href="http://www.w3.org/XML">http://www.w3.org/XML</a>
XSD	<i>XML Schema Definition</i> , <a href="http://www.w3.org/TR/xmlschema-0/">http://www.w3.org/TR/xmlschema-0/</a>
XSIG	<i>XML Signature</i> , <a href="http://www.w3.org/TR/xmldsig-core/">http://www.w3.org/TR/xmldsig-core/</a>
XSL	<i>XML Stylesheet Language</i> , <a href="http://www.w3.org/TR/1999/REC-xslt-19991116">http://www.w3.org/TR/1999/REC-xslt-19991116</a> , <a href="http://www.w3.org/TR/xslt20/">http://www.w3.org/TR/xslt20/</a>

## 6.2. Glossar

Begriff	Erklärung
Authentifizierung	<p>Authentifizierung ist der Nachweis (Verifizierung) einer behaupteten Eigenschaft eines Systems, eines Dokumentes oder einer Information. Bei einer Authentifizierung zwischen zwei Parteien authentisiert sich die Eine (z.B. der Sender), während die Andere (z.B. der Empfänger) die Erstere authentifiziert.</p> <p>Eine in der EDV weit verbreitete Form ist die Authentifizierung mittels UserID, Passwort, oder auch mittels Zertifikat und Signatur</p>
Body	<p>Im eXTra-Datenmodell enthält dieser Bereich entweder die gesamte nächsttiefere Ebene oder im Fall der untersten Ebene die fachlichen Nutzdaten.</p>
Clearing-Stelle	<p>Zentrale annehmende Stelle von Daten, welche die Funktion eines Bindegliedes zwischen fachlichem Sender und fachlichem Empfänger darstellt und für beide Seiten in der Regel für Rechtssicherheit und einen geordneten Betrieb sorgt.</p>
Datenübermittlungsverbund	<p>Heute können in einem Datenübermittlungsverbund definierte Nachrichten über ein konkretes Datenübermittlungsverfahren ausgetauscht werden. In der Regel definiert der Empfänger der Daten (zumeist eine Behörde, ein Verband oder eine Institution) sowohl das Datenübermittlungsverfahren als auch die Nachrichten. Die Teilnehmer, die Datenlieferanten, müssen üblicherweise beim Empfänger registriert sein.</p> <p>Die Idee von eXTra ist es für beliebige Datenübermittlungsverbände ein einheitliches Datenübermittlungsverfahren, bzw. über ein generisches Konzept – der <i>Profilierung</i> - eine Familie verwandter Datenübermittlungsverfahren zur Verfügung zu stellen.</p> <p>Das Ergebnis der Profilierung ist ein <i>verbundspezifischer eXTra-Standard</i>.</p>
DFÜ Ebene	<p>Die DFÜ Ebene repräsentiert innerhalb des abstrakten Architekturmodells eines Datenübermittlungssystems, sowie bei eXTra eine Ebene, bei der ein <i>DFÜ Sender</i> Daten mit einem <i>DFÜ Protokoll</i> an einen <i>DFÜ Empfänger</i> sendet. Weiterhin sind in der DFÜ Ebene in der Regel Sicherheitsmaßnahmen integriert, die insbesondere Angriffen aus der Internet-Welt entgegen wirken sollen.</p> <p>In eXTra wird die Ausgestaltung der DFÜ Ebene nicht behandelt; insofern trifft eXTra keinerlei Aussagen zum DFÜ-Protokoll oder zu den dort angesiedelten Sicherheitsmaßnahmen.</p>
ELSTER	<p>Die elektronische Steuererklärung ELSTER ist ein Datenübermittlungsverfahren der deutschen Steuerverwaltungen aller Länder und des Bundes mit dem ein Steuerpflichtiger oder in dessen Auftrag ein Dienstleister Erklärungs- und Anmeldesteuern an die Finanzverwaltung übermitteln kann. Auf Wunsch stellt die Finanzverwaltung Bescheide zum Abholen durch den Steuerpflichtigen bzw. in dessen Auftrag einem Dienstleister zum Abholen bereit.</p>
eSTATISTIK.core	<p>Datenübermittlungsverfahren des Statistischen Bundesamtes zur automatischen Übermittlung von statistischen Daten aus betrieblichen</p>

Begriff	Erklärung
	Anwendungen der Melder.
Fachverfahren	Mit Hilfe von eXTra kann ein Fachverfahren auf Senderseite – der fachliche Sender – fachliche Daten an das zugeordnete Fachverfahren auf Empfängerseite – den fachlichen Empfänger – übermitteln.  In umgekehrter Datenflussrichtung kann ein erzeugendes Fachverfahren auf Empfängerseite dem anfordernden Fachverfahren auf Senderseite fachliche Daten zum Abholen bereitstellen.
Header	In der Informationstechnik werden Metadaten am Anfang einer Datei oder eines Datenblocks als Header (auch: Dateikopf) bezeichnet. Diese können verwendet werden, um beispielsweise das Dateiformat zu beschreiben oder weitere Angaben zu den Daten zu machen.
Kommunikationsszenario	Ein Kommunikationsszenario definiert das erwartete Verhalten eines Empfängers auf einer Ebene.
Kommunikationsvorgang	Ein Kommunikationsvorgang definiert den Ablauf der Kommunikation zwischen Sender und Empfänger auf einer Ebene des eXTra-Kommunikationsmodells. Er legt fest, wie die Rollen Sender und Empfänger verteilt werden und ob eine synchrone Response möglich ist.
Komprimierung	Datenkompression oder Datenkomprimierung ist die Anwendung von Verfahren zur Reduktion des Speicherbedarfs von Daten.
Logischer Empfänger	Der logische Empfänger ist ein Akteur, der auf Empfänger-Seite der <i>Logistikebene</i> zugeordnet ist.
Logischer Sender	Der logische Sender ist ein Akteur, der auf Sender-Seite der <i>Logistikebene</i> zugeordnet ist.
Logistikebene	Die Logistik- oder auch Paketebene repräsentiert bei eXTra eine Ebene, die auf Senderseite durch den <i>logischen Sender</i> für die Bündelung mehrerer fachlichen Nachrichten für einen Endempfänger zu einem Paket zuständig ist, bzw. auf Empfängerseite durch den <i>logischen Empfänger</i> die Verteilung der Paketinhalte auf die Endempfänger/ Verwerter übernimmt.
Logging	Eine Logdatei beinhaltet das automatisch erstellte Protokoll aller oder bestimmter Aktionen von Prozessen auf einem Computersystem. Das Fortschreiben dieser Datei nennt man Logging.
Message	(Fach-)Nachricht, die ein Erzeuger mithilfe eines Fachverfahrens generiert und die ein Verwerter verarbeiten soll.  Eine eXTra-Message enthält im MessageBody in der Regel eine einzige Fachnachricht. Dadurch ist eine Fachnachricht für eXTra sicht- und greifbar.
Migration	Unter Migration versteht man im Rahmen der Informationstechnik den Umstieg eines wesentlichen Teils der eingesetzten Software beziehungsweise den Transfer von Daten aus einer Umgebung in eine andere, sowie die Umstellung von Hardware einer alten Technologie in neue Technologien unter weitgehender Nutzung vorhandener Infrastrukturen.
Nachrichtenebene	Die Nachrichtenebene repräsentiert bei eXTra eine Ebene, bei der z.B. auf

Begriff	Erklärung
	<p>Senderseite ein Erzeuger durch ein Fachverfahren eine fachliche Nachricht generiert, die in Form einer eXTra-Message übermittelt wird. Auf der Empfängerseite verarbeitet das korrespondierende Fachverfahren als Endempfänger und Verwerter die fachliche Nachricht.</p>
Package	<p>Ein Paket, das z.B. ein <i>logischer Sender</i> an einen <i>logischen Empfänger</i> übermittelt.</p> <p>Ein Package enthält im PackageBody entweder die fachlichen Daten oder die nächsttiefere Ebene, die Nachrichtenebene. Sind im PackageBody fachliche Daten enthalten, so ist es für eXTra nicht erkennbar, aus wievielen fachlichen Nachrichten der PackageBody zusammengesetzt ist.</p>
Paketebene	<p>Die Paket- oder Logistikebene erlaubt es, Einzelnachrichten zu Paketen zusammenzufassen und damit in verschiedener Hinsicht einheitlich zu behandeln. Sie unterstützt damit insbesondere die Massendatenverarbeitung.</p> <p>Ein derartiges Paket wird in Form eines eXTra-<i>Package</i> übermittelt.</p>
physikalischer Empfänger	<p>Der physikalische Empfänger ist ein Akteur, der auf Empfänger-Seite der <i>Transportebene</i> zugeordnet ist. Physikalischer Sender und –Empfänger stehen über ein konkretes Kommunikationssystem direkt miteinander in Verbindung und tauschen darüber eXTra-Dokumente aus.</p> <p>Die Ausgestaltung des konkreten Kommunikationssystems (und damit der verwendeten DFÜ-Protokolle und Netze) ist in eXTra nicht vorgegeben.</p>
physikalischer Sender	<p>Der physikalische Sender ist ein Akteur, der auf Sender-Seite der <i>Transportebene</i> zugeordnet ist. Physikalischer Sender und Empfänger stehen über ein konkretes Kommunikationssystem direkt miteinander in Verbindung und tauschen darüber eXTra-Dokumente aus.</p> <p>Die Ausgestaltung des konkreten Kommunikationssystems (und damit der verwendeten DFÜ-Protokolle und Netze) ist in eXTra nicht vorgegeben.</p>
Plug-Ins	<p>Softwarehersteller definieren Schnittstellen zu ihren Produkten, mit deren Hilfe Dritte Funktionserweiterungen (Plug-Ins) für diese Softwareprodukte programmieren können. In eXTra sind Plug-Ins optionale Erweiterungen des Datenmodells, die aber nicht unabhängig entwickelt werden können, sondern dem Standardisierungsprozess unterliegen</p>
Profilkonfiguration Profilierung	<p>Die Profilkonfiguration ist bei eXTra eine XML-Datei, die dazu dient aus dem allgemeinen eXTra-Basisschema für ein konkretes Fachverfahren bzw. einen konkreten <i>Datenübermittlungsverbund</i> auf formale Weise eine spezifische Schemadatei – ein eXTra-Subschema - zu generieren. Diesen Generierungsvorgang – die Profilierung - kann jedes Fachverfahren bzw. jeder Datenübermittlungsverbund selbst durchführen.</p>
Request-ID	<p>Anfragekennung.</p> <p>In der eXTra-Terminologie ist die Request-ID ein vom Sender vergebener eindeutiger Identifikator einer Anfrage.</p>
Response-ID	<p>Antwortkennung</p> <p>In der eXTra-Terminologie ist die Response-ID ein vom Empfänger vergebener eindeutiger Identifikator einer Antwort auf eine Anforderung mit einer eindeutigen RequestID.</p>

Begriff	Erklärung
Signierung	<p>Unter einer elektronischen Signatur versteht man Daten, mit denen man den Unterzeichner bzw. Signaturersteller identifizieren kann und sich die Integrität der signierten, elektronischen Daten prüfen lässt. Die elektronische Signatur erfüllt somit technisch gesehen unter bestimmten Bedingungen den gleichen Zweck wie eine eigenhändige Unterschrift auf Papierdokumenten. Den Vorgang nennt man Signierung.</p>
Topologie	<p>Die Topologie bezeichnet bei einem Computernetz die Struktur der Verbindungen mehrerer Geräte untereinander, um einen gemeinsamen Datenaustausch zu gewährleisten.</p> <p>Bei einem Datenübermittlungsverbund stellt die Topologie die Struktur der Verbindungen der einzelnen Teilnehmer bzw. Systeme des Datenübermittlungsverbundes dar.</p>
Transportebene	<p>Die Transportebene repräsentiert bei eXTra eine Ebene, bei der ein <i>physikalischer Sender</i> vollständige eXTra-Dokumente an einen <i>physikalischen Empfänger</i> sendet.</p> <p>Im TransportBody sind entweder die fachlichen Daten oder die nächsttiefere Ebene, die Paket- oder Nachrichtenebene enthalten. Sind im TransportBody fachliche Daten enthalten, so ist es für eXTra nicht erkennbar, aus wievielen fachlichen Nachrichten der TransportBody zusammengesetzt ist.</p>
URL	<p>Als Uniform Resource Locator (URL, engl. „einheitlicher Quellenanzeiger“) bezeichnet man eine Unterart von Uniform Resource Identifiern (URIs). URLs identifizieren eine Ressource über das verwendete Netzwerkprotokoll (beispielsweise http oder ftp) und den Ort (engl. location) der Ressource in Computernetzwerken.</p>
UUID	<p>Ein <b>Universally Unique Identifier (UUID)</b> ist ein Standard für Identifikatoren, der in der <u>Softwareentwicklung</u> verwendet wird. Er ist von der <u>Open Software Foundation (OSF)</u> als Teil des <u>Distributed Computing Environment (DCE)</u> standardisiert.</p> <p>Sinn und Zweck von UUIDs ist, Informationen in verteilten Systemen ohne zentrale Koordination eindeutig kennzeichnen zu können.</p>
Validierung	<p>In der Softwaretechnik bezeichnet Validierung (auch Plausibilisierung, als Test auf Plausibilität, oder engl. Sanity Check genannt) die Kontrolle eines konkreten Wertes darauf, ob er zu einem bestimmten Datentyp gehört oder in einem vorgegebenen Wertebereich oder einer vorgegebenen Wertemenge liegt.</p>
verbundspezifischer eXTra-Standard	<p>Die Idee von eXTra ist es für beliebige Datenübermittlungsverbünde ein einheitliches Datenübermittlungsverfahren, bzw. über ein generisches Konzept – der <i>Profilierung</i> - eine Familie verwandter Datenübermittlungsverfahren zur Verfügung zu stellen.</p> <p>Das Ergebnis der Profilierung ist ein verbundspezifischer eXTra-Standard.</p>
Verschlüsselung	<p>Verschlüsselung nennt man den Vorgang, bei dem die Repräsentation einer Informationseinheit wie etwa ein Text oder eine Bilddatei aus einer unverschlüsselten Form, dem sogenannten Klartext, in eine verschlüsselte Form, dem sogenannten Geheimtext, überführt wird. In der Regel erfolgen Ver- und Entschlüsselung mit Hilfe mathematischer Verfahren, die hierzu</p>

Begriff	Erklärung
	ein oder mehrere extern zugeführte Schlüssel verwenden.
W3C	Das World Wide Web Consortium (W3C, <a href="http://www.w3.org">http://www.w3.org</a> ) entwickelt Standards und Technologien für das Internet
XSLT Stylesheet	XSL Transformation, kurz XSLT, ist eine XML-basierte Sprache für die Transformation von XML-Dokumenten. Sie ist Teil des W3C-Standards Extensible Stylesheet Language (XSL).