Implementation Guide

Stoffsammlung

4. Oktober 2008

1. Profilierung

Verfahren
- Rolle der Basisschemata
- Rolle der Profilkonfiguration
- Das Ergebnis: die verbundspezifischen Schemata

Regeln der Profilierung

Auswahl der Ebenen:

Pflichtigkeit der 3 Ebenen oder Regeln der Verkürzung

Pflicht ist nur die Transport-Ebene

DÜ-Verfahren kann im Rahmen der Profilierung festlegen ob es eine Packet-und/ oder Message-Ebene geben muss bzw. kann

Gestaltung einer Ebene

Minimalumfang einer Ebene

Header und Body sind Pflicht (=Kernbereich)

DÜ-Verfahren kann im Rahmen der Profilierung festlegen ob es Signatures, Plug-in und Logging geben muss bzw. kann

Gestaltung des Headers

Ausgestaltung der Header

der Header ist strukturell für jede Ebene gleich

der Header ist strukturell für alle DÜ-Verfahren gleich

Ausgestaltung der Header

im Header gibt es Pflicht-Elemente und optionale Elemente

DÜ-Verfahren kann im Rahmen der Profilierung festlegen, wie bezüglich der optionalen Elemente der Header auf Transport-, Packet- und Message-Ebene definiert ist. Die Festlegung könnte ebenenspezifisch unterschiedlich sein.

DÜ-Verfahren kann im Rahmen der Profilierung festlegen, dass ein optionales Element zum Pflichtelement wird, oder dass ein optionales Element nicht unterstützt wird (Ignored).

DÜ-Verfahren kann im Rahmen der Profilierung dagegen **nicht** festlegen, dass ein Pflichtelement zum optionalen Element wird oder nicht unterstützt wird.

Wenn ein optionales Element 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 nicht ausgewertet; wird das Element angegeben, muss es vom Empfänger akzeptiert werden

Gestaltung des Body

Beginn der fachlichen Nutzdaten

die fachlichen Nutzdaten liegen im <body> der untersten Ebene, z.B. der Message-Ebene. Formal sind sie mit dem Tag <data> umschlossen, sodass entscheidbar ist, aus wie vielen Ebenen eine Lieferung besteht.

Frage: sollen die Nutzdaten formal noch klassifiziert werden können, um sie vor der Übergabe an die Verarbeitungsinstanz noch syntaktisch prüfen zu können?

in Zeichenfolge <charsequence>
<base64>,
<elementsequence>
<anyXML>

Festlegung

Unter <data> wird diese Möglichkeit als choice eröffnet

2. Wertelisten

Formale Absicherung von Werteliste über URI

Die Notwendigkeit der formalen Absicherung von Wertelisten – nach dem Vorschlag von Herrn Schäfer – sollte im Zusammenhang mit der kontrollierten Weiterentwicklung des Standards gesehen werden, also als ein Mittel Wildwuchs zu verhindern. Wo diese Notwendigkeit nicht besteht oder wenn sie aus Gründen der Flexibilität und Erweiterbarkeit hinderlich ist, sollte man auf eine formale Absicherung verzichten.

Festlegung:
Eine formale Absicherung erfolgt bei

- Profil
- Plugin

Ob eine formale Absicherung erfolgt obliegt dem jeweiligen Fachverfahren bei

- <szenario>
- <procedere>

Keine formale Absicherung erfolgt bei

- <datatyp>; hier wäre sogar eine Behinderung für die weitere Nutzung des Standards für ein bereits unterstütztes Verfahren gegeben.

3. Verschlüsselung

- Die Verschlüsselung wird gemäß w3c-Standard vorgenommen
- Der zu verschlüsselnde Bereich ist gemäß w3c-Standard mit den Tags <encrypted data> umschlossen
- Der zu verschlüsselnde Bereich ist immer ein ganzes Strukturelement, z.B. der gesamte Header, Body, Plugins, etc.
- Ausnahme ist der TransportHeader: dieser wird nicht verschlüsselt

4. Verschlüsselung + Komprimierung

Verschlüsselung + Komprimierung
im w3c-Standard ist dieser Fall mit „transforms" enthalten und wird entsprechend verwendet

Frage: ist dadurch der zu komprimierende und zu verschlüsselnde Bereich gleich?
Antwort: ja

Verschlüsselung, Komprimierung
die Information ob und welche Art der Verschlüsselung bzw. Komprimierung verwendet wurde geht auf dem Weg vom physikalischen zum logischen Empfänger verloren (das Tag <encrypted data> wird aus der XML-Struktur entfernt). Die der Datenübermittlung nachgeordneten Instanzen auf Empfänger-Seite benötigen aber diese Information (Nachvollzug).
vorläufig: da dies nur wenige DÜ-Verfahren benötigen, wird diese Information in ein Plug-in ausgelagert

Verschlüsselung/Komprimierung der einzelnen Strukturelemente
TransportHeader bleibt immer lesbar und wird nicht verschlüsselt/ komprimiert

alle anderen Strukturelemente auf Transportebene wie zB. Plug-in können – sofern sinnvoll - optional verschlüsselt/komprimiert werden

alle Strukturelemente auf Packet- und Message- Ebene können optional verschlüsselt/ komprimiert werden
Wenn ein Strukturelement, z.B. Body, verschlüsselt werden soll, dann immer als ganzes

Verschlüsselung von <body>
Tag <encrypted data> steht unmittelbar hinter <body>
Anmerkung: die Vertraulichkeit ist besser gewahrt, wenn auf der entsprechenden Ebene nicht erkennbar ist, welche Strukturelemente sich in der nächst tieferen Ebene befinden bzw. ob es noch eine weitere tiefere Ebene gibt.

Verwendung von <encrypted data>

Laut w3c-Standard kann <encrypted data> in 2 Fassungen angewendet werden:

a) Ersatz des Vorgänger-Tags und Verschlüsselung des Inhalts

b) Vorgänger-Tag bleibt erhalten und Inhalt wird verschlüsselt
Beispiel:

```
<body>
        <encrypted data>
                verschlüsselte Daten
        </encrypted data>
</body>
```

Festlegung:
In eXTra wird nur Alternative b verwendet

5. Komprimierung ohne Verschlüsselung

Komprimierung ohne Verschlüsselung

Alternative 1
im w3c-Standard ist diese Konstellation zulässig, bei ISIS-MTT ausgeschlossen

Anmerkung1: ein Konflikt mit ISIS-MTT wird nicht gesehen, da die Zielrichtung von ISIS-MTT die Signatur ist, die wiederum an dieser Stelle keine Rolle spielt

Anmerkung2: auch wenn w3c diese Konstellation zulässt, kann nicht davon ausgegangen werden, dass die derzeit vorhandenen Implementierungen diese Konstellation unterstützen

Alternative 2
Auslagerung der entsprechenden Information zur Komprimierung in Plugin

Festlegung: Alternative 2, Auslagerung in Plugin

6. Signaturen

Bezugspunkt ist vornehmlich der w3C-Standard xml signatures. Der eXTra-Standard ist jedoch so flexibel, dass bestehende Sicherheitsverfahren idR weiter verwendet werden können.

Der zu signierende Bereich ist maximal ein ganzes, minimal nur Teil eines Strukturelementes, das sich auf der gleichen Ebene wie das Strukturelement Signatures befindet, z.B. der gesamte oder Teil des Header, Body, Plugins, etc.
Im Gegensatz zur Verschlüsselung wird bei der Signatur beim TransportHeader keine Ausnahme gemacht, dieser kann ebenfalls signiert sein.

Im Strukturelement Signatures können mehrere Signaturen enthalten sein, z.B. die Signatur des Header und des Body

7. Logging

8. Plugins

Standard-Plugins
Komprimierung
Weitergabe der Information welche Art der Verschlüsselung/Komprimierung verwendet wurde

9. Kommunikationsvorgänge

Es gibt 2 Vorgänge: Request und Response
- die Ausgestaltung eines Headers für Request und Response ist unterschiedlich
- der Header für Response enthält alle Elemente von Request und zusätzlich" das Element „Responsedetails"

XML-Sprachmittel zur Unterscheidung von Request – Response:
Die Unterscheidung wird anhand verschiedener Namensräume getroffen

9.1. Zusammenspiel von Request und Response, ausgelöst durch das Element „scenario"

9.1.1. prinzipieller Umgang mit dem Element „scenario" in den requestdetails

Scenario gibt es in 3 Ausprägungen:
- „Fire-and-forget"
- „request-with-acknowledgement"
- „request-with-response"

Die Wirkung von „Fire-and-forget" ist einfach: der Sender verzichtet auf eine Response des Empfängers.

Die Bedeutung von „request-with-acknowledgement" bzw. „request-with-response" wird anhand von Szenarien diskutiert, die beispielhaft und etwas vereinfacht aus der Finanzverwaltung entnommen sind.

Folgendes Szenario1 sei gegeben:
Der Sendevorgang besteht nur aus der Transport-Ebene, die fachlichen Nutzdaten sind im Transport-body enthalten. Sie werden an die fachliche Verarbeitungsinstanz weitergereicht, die diese Daten in Batchläufen verarbeitet. Die Ergebnisse eines derartigen Batchlaufes, z.B. in Form von Fehlermeldungen, stehen also nicht sofort zur Verfügung, sondern erst zeitverzögert nach n Stunden.
Beispiel für dieses Szenario ist die Übermittlung von Lohnsteuerbescheinigungen an die Elster-Clearingstelle. Die Protokolle, ob die übermittelten Lohnsteuerbescheinigungen verarbeitet werden können, stehen erst nach mehreren Stunden zur Verfügung, können also als Rückmeldung des Sendevorgangs noch nicht rückgemeldet werden.

Folgendes Szenario2 sei gegeben:
Dieses Szenario ist die Fortsetzung des obigen Szenarios1: jetzt will der Sender die Protokolle abholen. Eingeläutet wird dieser Vorgang durch das senden einer Anforderung, die nur aus der Transport-Ebene besteht, die fachlichen Nutzdaten sind im Transport-body enthalten. Die fachliche Nutzdaten sind die Angaben, mit denen der Sender formuliert, was er genau will. Sie werden an die fachliche Verarbeitungsinstanz weitergereicht, die diese Anforderung sofort bearbeitet. Das Ergebnis der Bearbeitung kann entweder sein „noch keine Daten vorhanden" (dann muss der Sender die Anforderung später nochmals stellen) oder „hier sind die Daten". Die Transport-Ebene muss somit auf die Antwort der fachlichen Verarbeitungsinstanz warten, um diese zurückzumelden.
Während das Datenvolumen für das senden der Anforderung gering sein wird, könnte die Rückantwort – je nach Anforderung - sehr umfangreich ausfallen.
Beispiel für dieses Szenario ist die Anforderung des Prüfprotokolls von Lohnsteuerbescheinigungen. Der Sender kann hier immer nur das Protokoll genau einer Lieferung (mit der Ticketnummer xyz) abholen.

Folgendes Szenario3 „Dialoganfrage" sei gegeben:
Der Sendevorgang besteht nur aus der Transport-Ebene, die fachlichen Nutzdaten sind im Transport-body enthalten. Sie werden an die fachliche Verarbeitungsinstanz weitergereicht, die diese Daten sofort verarbeitet, weil der Sender eine Dialoganfrage gestellt hat und auf die Antwort wartet. Die Ergebnisse dieser Anfrage müssen quasi sofort zur Verfügung gestellt und an den Sender zurückgemeldet werden können. Die Transport-Ebene muss somit auf die Antwort der fachlichen Verarbeitungsinstanz warten, um diese zurückzumelden.
Beispiel für dieses Szenario ist die Steuerkontoabfrage.

Die Frage ist, wie die Transport-Ebene der Empfängerseite auf „request-with-acknowledgement" und „request-with-response" reagieren muss, um all diesen Szenarien gerecht zu werden.

Variante 1:
Die beiden Ausprägungen „request-with-acknowledgement" und „request-with-response" beziehen sich auf Aussagen der Transport-Ebene. Mit einem positiven „acknowledgement" bestätigt die Transport-Ebene auf Empfängerseite die erfolgreiche Annahme und Abspeicherung der Lieferung. Und mit einer positiven „response" wird signalisiert, dass alle Aktionen auf der Transport-Ebene (verifizieren von Signaturen, dekomprimieren, entschlüsseln, erzeugen von logging-Einträgen) erfolgreich durchlaufen wurden und die Daten an die nächst tiefere Ebene bzw. an die Verarbeitungsinstanz weitergereicht werden können.
Allerdings fehlt jetzt ein Sprachmittel, das der Transport-Ebene signalisiert, dass sie beim obigen Szenario 2 oder 3 auf die Antwort der fachlichen Verarbeitungsinstanz warten muss, um diese dem Sender zurück zu liefern. Dieses fehlende Sprachmittel könnte z.B. eine zusätzliche Ausprägung von scenario „request-with-result" sein.

Variante 2:
Nur die Ausprägung „request-with-acknowledgement" bezieht sich auf die Transport-Ebene. „request-with-response" wendet sich immer an die fachliche Verabeitungsinstanz, die eine Antwort der wartenden Transport-Ebene liefern soll, die diese wiederum an den ebenfalls wartenden Sender weiterreichen muss.
Mit einem positiven „acknowledgement" bestätigt die Transport-Ebene auf Empfängerseite zumindest die erfolgreiche Annahme und Abspeicherung der Lieferung.

Ob sie darüber hinaus auch das vollständige und erfolgreiche durchlaufen aller Aktionen auf der Transport-Ebene (verifizieren von Signaturen, dekomprimieren, entschlüsseln, erzeugen von logging-Einträgen) sinnvollerweise signalisieren kann, hängt von mehreren Faktoren ab, u.a. vom Datenvolumen der Nutzdaten und von der zu erwarteten Bearbeitungszeit für das durchlaufen der gesamten Aktionskette. Diese prinzipiellen Überlegungen muss sich jede konkrete Implementierung eines eXTra-Empfangssystems machen und festlegen, wie aussagekräftig und differenziert ein positives „acknowledgement" sein kann. Die konkrete Aussagekraft könnte im Tag <result> hinterlegt werden. Die Bandbreite kann minimal gehen von „Datenannahme und Abspeicherung erfolgreich" bis maximal zu „Datenannahme und Weitergabe an die nächste Ebene/Verarbeitungsinstanz erfolgreich".

Vergleich der beiden Varianten:
In der Variante 2 wurde eine zusätzliche Ausprägung von scenario im Sinne eines „request-and-result" vermieden, aber eine gewisse Unschärfe des eXTra-Standards bei der Bedeutung von „request-with-acknowledgement" in Kauf genommen.
Ursache dieser Unschärfe ist die Unsicherheit, ob die Transportebene in jedem Fall immer das vollständige und erfolgreiche durchlaufen aller Aktionen auf der Transport-Ebene sinnvollerweise signalisieren kann. Diese Unsicherheit ist bei beiden Variante in gleicher Weise vorhanden. Der Zugewinn des Sprachmittels „request-and-result" der Variante 1 ist somit recht zweifelhaft

Vorschlag:
Die Variante 2 ist die bessere Variante, da sie die Unschärfe des Standards bei der konkreten Implementierung eines eXTra-Empfangssystems egalisieren kann indem sie – wie gezeigt – die Bedeutung eines „acknowledgement" explizit mitteilt.

9.1.2. die Verwendung des Elements „scenario" auf den verschiedenen Ebenen

Unter 9.1.1 wurde anhand mehrerer Szenarien diskutiert, welche Auswirkung die verschiedenen Ausprägungen von scenario auf die Transport-Ebene des eXTra-Empfangssystems hat. Scenario ist das entscheidende Element, mit dem der Sender ausdrückt, ob und welche Rückmeldungen er von der Transport-Ebene erwartet. Auf Grund dieser zentralen Bedeutung ist das Element scenario ein Pflichtelement auf der Transport-Ebene.

Die Frage ist nun, welche Bedeutung das Element scenario hat, bzw. welchen Nutzen der Sender davon hat, wenn er scenario auf allen Ebenen angibt.

Entscheidend für den Sender sind zwei Dinge:

- eine Bestätigung für den Empfang der Lieferung durch das Empfangssystem, um den Nachweis erbringen zu können, dass er die Daten fristgerecht abgegeben hat. Wenn möglich sollte die Bestätigung auch beinhalten, dass auf Transport-Ebene die gesamte Aktionskette erfolgreich durchlaufen wurde und die Daten an die nächste Ebene/Verarbeitungsinstanz weitergeben werden können. Dies ist zwar für den Nachweis der fristgerechten Abgabe nicht erforderlich, beruhigt den Sender aber ungemein, was die Verarbeitungssicherheit angeht.
- Eine Bestätigung durch das fachliche Verarbeitungssystem, dass die Daten vollständig und korrekt verarbeiten werden konnten.

Der Zugewinn für den Sender, dass er weitere Zwischenmeldungen von den tieferen Ebenen über die weitere erfolgreiche Weiterleitung seiner Daten erhält, ist nur sehr

gering. Die zu lösende Komplexität, wie derartige Zwischenmeldungen an den Sender zurückgemeldet werden können, wäre dagegen erheblich.

Vorschlag:

- Auf der Transport-Ebene ist das Element scenario auf Grund seiner zentralen Bedeutung ein Pflichtelement
- Auf der Packet- und Message-Ebene ist das Element scenario formal ein optionales Element. Es ist jedoch ohne Bedeutung, wird nicht ausgewertet und bewirkt also keinerlei Aktionen in der Empfänger-sphäre.

9.1.3. Gestaltung der Response auf einen Sendevorgang

Eine Response des Empfängers gibt es nur dann, wenn im Transport-Header scenario = „request-with-acknowledgement" oder „request-with-response" angegeben wurde.

Zum besseren Verständnis des Zusammenspiels werden beispielhaft mehrer Szenarien dargelegt

Folgendes Szenario1 sei gegeben:
Der Sendevorgang besteht aus 1 Ebene, im Transport-Header wurde scenario = „request-with-acknowledgement" angegeben, d.h. es wird eine Rückantwort, eine Response der Transport-Ebene des Empfängers erwartet.

Frage: wie sieht diese Rückmeldung aus?

Vorschlag:
Bei scenario = „request-with-acknowledgement" besteht die Rückmeldung nur aus Transport-Header mit request-details und response-details, wobei die eigentliche Aussage z.B. „Datenempfang wird bestätigt" im Tag <report> hinterlegt ist.

Ein Element Transportbody ist eigentlich unnötig.
Aus syntaktischen Gründen muss der Transportbody (weil Pflichtelement) jedoch formal angegeben werden (body enthält keine Information: </body>)

Folgendes Szenario2 sei gegeben:
Der Sendevorgang besteht aus 2 Ebenen, im Transport-Header wurde scenario = „request-with-acknowledgement" und im Packet-Header fehlt scenario (gemäß Vorschlag von 9.1.2), d.h. es wird eine Rückantwort, eine Response nur von der Transport-Ebene des Empfängers erwartet.

Frage: wie sieht diese Rückmeldung aus?

Vorschlag:
Die Rückmeldung sieht genauso aus wie in Szenario1: die Rückmeldung besteht nur aus der Transport-Ebene. Der Transport-Header enthält die request-details und response-details, wobei die eigentliche Aussage z.B. „Datenempfang wird bestätigt" im Tag <report> hinterlegt ist.

Ein Element Transportbody ist bei der Rückmeldung eigentlich unnötig.
Aus syntaktischen Gründen muss der Transportbody (weil Pflichtelement) jedoch als leeres Element geliefert werden.

Die Packet-Ebene fehlt bei der Rückmeldung.

Folgendes Szenario3 „Dialoganfrage" sei gegeben:
Der Sendevorgang besteht nur aus der Transport-Ebene, die fachlichen Nutzdaten (eigentliche Dialoganfrage) sind im Transport-Body enthalten. Sie werden an die fachliche Verarbeitungsinstanz weitergereicht, die in den requestdetails mit dem Element procedure spezifiziert wird. Dass eine sofortige Antwort der fachlichen Verarbeitungsinstanz gefordert ist, wird durch scenario = „request-with-response" formuliert. Die Transport-Ebene muss somit auf die Antwort der fachlichen Verarbeitungsinstanz warten, um diese dem Sender zurückzumelden.

Die Topologie der Empfänger-Sphäre sei einfach, die Transport-Ebene und die fachliche Verarbeitungsinstanz sollen am gleichen Ort angesiedelt sein.
Im Normalfall besteht die Rückmeldung ebenfalls nur aus der Transport-Ebene.

Frage: wie sieht diese Rückmeldung aus?

Vorschlag:
Die Rückmeldung besteht nur aus der Transport-Ebene. Im Transport-Header sind die request-details und response-details enthalten, wobei die Aussage „alles ok und die Antwort der fachlichen Verarbeitungsinstanz liegt vor" oder „die fachliche Verarbeitungsinstanz ist zur Zeit nicht verfügbar" oder „es gab auf der Transport-Ebene folgende Fehler" im Tag <report> hinterlegt ist.
Die Antwort der fachlichen Verarbeitungsinstanz ist im Transport-Body enthalten.
Bezüglich der Ausgestaltung der Antwort gibt der eXTra-Standard nichts vor.

9.1.4. Gestaltung der Response beim Holvorgang

Der Holvorgang wird eingeleitet mit einer Anforderung Daten (z.B. Ergebnisprotokolle der fachlichen Verarbeitungsinstanz) rückzuliefern, die sich an die fachliche Verarbeitungsinstanz richtet. Die Anforderung wird mittels request und in den requestdetails mit scenario = „request-with-response" gestellt. Wenn die fachliche Verarbeitungsinstanz auf der Empfängerseite die Daten der ursprünglichen Lieferung bereits verarbeitet hat und ein Ergebnisprotokoll bereits vorliegt, kann die Anforderung durch die fachliche Verarbeitungsinstanz auch erfüllt werden. Ist dies noch nicht der Fall muss der Sender die Anforderung später nochmals stellen.

Folgendes Szenario1 sei gegeben:
Die ursprüngliche Lieferung bestand nur aus der Transport-Ebene.
Die Protokollanforderung des Senders wird nur auf der Transport-Ebene formuliert und lautet sinngemäß „gib mir die Rückmeldung zu genau der Lieferung mit der requestID/responseID = nnn". Auf der Empfängerseite ist die Rückmeldung der spezifizierten Lieferung durch die fachliche Verarbeitungsinstanz entweder bereits fertig bereitgestellt, oder noch nicht vorhanden. Die angeforderte Rückmeldung lautet beispielsweise sinngemäß „bei der Verarbeitung der spezifizierten Lieferung war alles ok" oder „beim folgenden Datensatz trat der Fehler Fnnn" auf.

Der Sender formuliert die Anforderung folgendermaßen: sie besteht aus dem Transportheader mit scenario = „request-with-response", procedure= Name_der_fachlichen_Verarbeitungsinstanz und Datatyp =Protokollanforderung und einem Transportbody, in dem unter dem Element data die Forderung „Rückmeldung zur Lieferung mit der requestID/responseID = nnn" des Datentyps y spezifiziert ist.

Frage: wie sieht die Response auf diese Anforderung aus, wenn die ursprüngliche Lieferung nur aus der Transport-Ebene bestand?

Vorschlag:
Die Rückmeldung besteht nur aus der Transport-Ebene.
Auf der Transport-Ebene sind im Transport-Header die request-details und response-details enthalten, wobei die Aussage „alles ok und das Ergebnisprotokoll der fachlichen Verarbeitungsinstanz liegt vor" oder „die fachliche Verarbeitungsinstanz ist zur Zeit nicht verfügbar" oder „es gab auf der Transport-Ebene folgende Fehler" im Tag <report> hinterlegt ist.
Die angeforderte Rückmeldung der fachlichen Verarbeitungsinstanz im Sinne von „alles ok" oder „beim folgenden Datensatz trat der Fehler Fnnn auf" befinden sich im Transport-Body unterhalb des Elements data.

Folgendes Szenario2 sei gegeben:
Die ursprüngliche Lieferung bestand nur aus der Transport-Ebene.
Die Anforderung des Senders wird nur auf der Transport-Ebene formuliert und lautet sinngemäß „gib mir alle Rückmeldungen, die sich auf Lieferungen vom Datentyp y beziehen, die zwischen Datum 1 und Datum 2 gesendet wurden". Auf der Empfängerseite sind die Rückmeldungen des Datentyps y durch die fachliche Verarbeitungsinstanz entweder bereits fertig bereitgestellt, oder noch nicht vorhanden. Die angeforderten Rückmeldungen beziehen sich auf eine oder mehrere Lieferungen, die in der Vergangenheit gesendet wurden und für die jetzt die Rückmeldungen der fachlichen Verarbeitungsinstanz angefordert werden im Sinne von „bei der damaligen Lieferung war alles ok" oder „beim folgenden Datensatz trat der Fehler Fnnn" in der fachlichen Verarbeitungsinstanz auf.

Der Sender formuliert die Anforderung folgendermaßen: sie besteht aus dem Transportheader mit scenario = „request-with-response", procedure= Name_der_fachlichen_Verarbeitungsinstanz und Datatyp =Ergebnisprotokoll und einem Transportbody, in dem unter dem Element data die Forderung „alle Rückmeldungen bezüglich der Lieferungen mit dem Datatyp=y vom Datum1 bis Datum2" spezifiziert ist. Im allgemeinen Fall kann es sich nun ergeben, dass alle, oder nur einige, oder noch keine Rückmeldungen vorliegen für die Lieferungen im angegebenen Zeitraum. Deshalb werden die Rückmeldungen in Packets bereitgestellt, wobei sich jedes Packet auf genau eine Lieferung bezieht.

Frage: wie sieht die Response auf diese Anforderung aus, wenn die ursprüngliche Lieferung nur aus der Transport-Ebene bestand?

Vorschlag:
Positiver Fall
Die Rückmeldung besteht im positiven Fall aus der Transport- und der Packet-Ebene.
Auf der Transport-Ebene sind im Transport-Header die request-details und response-details enthalten, wobei im Tag <report> ein „alles ok" hinterlegt ist.
Im Transport-Body befindet sich die Packet-Ebene mit mehreren Packets.
Auf der Packet-Ebene sind im Packet-Header ebenfalls die request-details und response-details enthalten Im Tag <report>ist ein „alles ok" hinterlegt ist, weil das Packet korrekt erzeugt und zur Verfügung gestellt wurde. Die angeforderten Rückmeldungen der fachlichen Verarbeitungsinstanz im Sinne von „alles ok" oder „beim folgenden Datensatz trat der Fehler Fnnn auf" befinden sich im Packet-Body unterhalb des Elements data.

Frage: wie sehen die request-details im Packet-Header der Response aus, wenn der Sender in der Anforderung doch gar keine Packet-Ebene und damit keine request-details mitgegeben hat?

Der eXTra-Standard schreibt hier lediglich vor, dass die request-details vorhanden sein müssen. Wie z.B. die requestID aussieht bleibt der jeweiligen Implementierung überlassen.

Negativer Fall
Im negativen Fall – wenn die Transport-Ebene die Anforderung nicht an die fachliche Verarbeitungsinstanz weiterreichen kann – gibt es nur die Transport-Ebene, wobei der Grund (unbekannte fachliche Verarbeitungsinstanz oder fachliche Verarbeitungsinstanz nicht verfügbar) im Transportheader in den responsedetails im Tag <report> hinterlegt ist.
Das Element transportbody wird aus syntaktischen Gründen (Pflichtelement) als leeres Element geliefert.

Betriebsfall
Im ungünstigen Fall hat die fachliche Verarbeitungsinstanz die Daten der ursprünglichen Lieferung(en) noch nicht verarbeitet, so dass kein Ergebnisprotokoll vorliegt. Diese Auskunft „keine Daten vorhanden" der fachlichen Verarbeitungsinstanz wird im Transportbody zurückgemeldet.

Anmerkung: eXTra bietet keine Mittel an, um die Bandbreite an unterschiedlichen Situationen differenziert darzustellen im Sinne von „es liegen Rückmeldungen zu allen Lieferungen im genannten Zeitraum vor", oder „es liegen nur Rückmeldungen zu einigen (aber nicht allen) Lieferungen vor". Die Überwachung der Vollständigkeit ist nicht Aufgabe der Transportebene, diese Überwachung muss ggfls. auf der Packet-Ebene des Senders der ursprünglichen Lieferungen geleistet werden.

10. Die Elemente des Headers

10.1. Testindicator

<Testindicator>
Festlegung der Bedeutung auf den einzelnen Ebenen

Festlegung: es gibt 4 Ausprägungen:
=none: kein Testfall, sondern Produktivfall
=receive: Test bis zur Annahme durch die jeweiligen Instanz
=accept: Test bis zum Ende der Verarbeitung durch die jeweiligen Instanz, d.h. ggfls. mit der Entschlüsselung und Dekomprimierung der Daten, sowie Verifikation der Signaturen und Erzeugung des Strukturelements Logging
=process: Test der jeweiligen Instanz und Weitergabe an die nächste Ebene

Regeln:

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

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

Erläuterungen:

Transportebene:

receive: Test des Transports bis zur Annahme der Daten beim physikalischen Empfänger

accept: Test des Transports zum physikalischen Empfänger bis zum Ende der Verarbeitung auf Transportebene, d.h. ohne Weitergabe zum log. Empfänger

=process: Test der Transportebene + Weitergabe an den log. Empfänger

Packet-Ebene:

receive: Test des Transports bis zur Annahme der Daten beim logischen Empfänger

accept: Test des Transports zum logischen Empfänger bis zum Ende der Verarbeitung auf Packet-Ebene, d.h. ohne Weitergabe zur Message-Ebene

=process: Test der Transport- und Packet-Ebene + Weitergabe an die Message-Ebene

Message-Ebene:

receive: Test des Transports bis zur Annahme der Daten durch den fachlichen Empfänger

accept: Test des Transports zum fachlichen Empfänger bis zum Ende der Verarbeitung auf Message-Ebene, d.h. ohne Weitergabe zur Verarbeitungsinstanz des Fachverfahrens

=process: Test der Transport-, Packet- und Message-Ebene + Weitergabe an die Verarbeitungsinstanz des Fachverfahrens

Beispiele im Zusammenspiel mit 3 Ebenen:

Beispiel 1: Test nur mit der Transportebene:

Transport-Header, Testindicator = receive oder accept, Testfall verbleibt in der Transport-Ebene

Packet-Header, Testindicator irrelevant, da der Testfall nicht an die Packet-Ebene weitergereicht wird

Message-Header, Testindicator irrelevant, da die Transport-Ebene den Testfall nicht an die unteren Ebenen weiterreicht

Beispiel 2: Test nur mit der Transport- und Packet-Ebene:

Transport-Header, Testindicator = process, Testfall wird weitergereicht

Packet-Header, Testindicator =receive oder accept, Testfall verbleibt in der

Packet--Ebene

Message-Header, Testindicator irrelevant, da die Packet-Ebene den Testfall nicht an die Message-Ebene weiterreicht

Beispiel 3: Test bis zur Annahme der Daten bei der Verarbeitungsinstanz des Fachverfahrens:

Transport-Header, Testindicator = process, Testfall wird weitergereicht

Packet-Header, Testindicator = process, Testfall wird weitergereicht

Message-Header, Testindicator = receive oder accept, Testfall verbleibt in der Message-Ebene

Beispiel 4: Test mit Verarbeitung des Testfalls durch die Verarbeitungsinstanz des Fachverfahrens:

Transport-Header, Testindicator = process, Testfall wird weitergereicht

Packet-Header, Testindicator = process, Testfall wird weitergereicht

Message-Header, Testindicator = process, Testfall wird an das Fachverfahren weitergereicht und dort verarbeitet

Beispiel 5: Test mit Verarbeitung des Testfalls durch die Verarbeitungsinstanz des Fachverfahrens unter Verwendung des Produktivsystems auf der Transport- und Packet-Ebene:

Transport-Header, Testindicator = none, wird auf der Transport-Ebene als Produktivfall behandelt und zur nächsten Ebene weitergereicht

Packet-Header, Testindicator = none, wird auch auf der Packet-Ebene als Produktivfall behandelt und zur nächsten Ebene weitergereicht

Message-Header, Testindicator = process, Testfall wird an das Fachverfahren weitergereicht und dort verarbeitet

<Testindicator> optionales oder Pflicht-Element

Festlegung:

a) Testindicator ist optional;
   Zweck: ein Produktiv-Fall (=Regelfall) muss nicht mit Testindicator = none deklariert werden
b) Testindicator wird als optionales Element deklariert, das aber (als bisher einzige Ausnahme) nicht profilierbar ist
   Zweck: die DÜ-Verfahren werden angehalten grundsätzlich für Testmöglichkeiten zu sorgen

10.2. Zusammenwirken von scenario, procedure und datatyp

Allgemeines

<scenario>, <procedure>, <datatyp>

Frage: sind die Tags auf jeder Ebene Pflicht oder optional?

Tags sind optional, können aber im Rahmen der Profilierung Pflichtangabe werden (z.B. procedure im TransportHeader optional, im PacketHeader

jedoch Pflicht) oder auch je nach Ebene ganz entfallen.

Empfehlungen

Benötigt ein DÜ-Verfahren nur die Transportebene, so sollten diese drei Elemente als Pflichtelemente definiert werden. Damit werden implizite Festlegungen vermieden und das DÜ-Verfahren kann im Zuge von Erweiterungen z.B. mehrere Fachverfahren (<procedures>) und/oder Datentypen ohne Anpassungsmaßnahmen unterstützen

Unterstützt ein DÜ-Verfahren neben der Transportebene auch die Packet- oder Message-Ebene, so sollten auf der Transport-Ebene <scenario> als Pflichtelement, dagegen <procedure> und <datatyp> gar nicht oder nur als optionale Elemente definiert werden. Damit wird klar zum Ausdruck gebracht, dass <procedure> und <datatyp> Begriffe sind, die nicht zur Transportebene gehören.

Querbezug des eXTra-Kommunikationsszenarios <scenario> zur der unterlagerten DFÜ-Ebene mit einem spezifischen DFÜ-Protokoll z.B. e-mail oder http(s)

Was gibt SOAP hierzu vor (scenario und deren Ausprägungen ist von SOAP entlehnt)? Ist bei SOAP ein Querbezug zum darunterliegenden DFÜ-Protokoll vorhanden? Gibt das darunterliegende DFÜ-Protokoll die möglichen Ausprägungen von <scenario> vor? Ist also z.B. bei e-mail ein „request-with-acknowledgement" überhaupt zulässig?

Antwort: SOAP hat keinen Querbezug zum DFÜ-Protokoll, sondern betrachtet dies abstrakt. Bei SOAP gibt es keine zeitlichen Vorschriften, wann ein acknowledgement, bzw. eine response zu erfolgen hat. Bei e-mail wäre also ein Vorgang „request-with-acknowledgement" zulässig, das dann in 2 DFÜ-Aktionen = 1 request e-mail durch den Sender und 1 acknowledgement e-mail durch den Empfänger aufzuspalten wäre.

Vorschlag:

Konformität zu SOAP, d.h. kein Querbezug zum verwendeten DFÜ-Protokoll

Festlegung:
Bei Verwendung von Punkt-zu-Punkt Verbindungen, z.B. http(s) soll am Abschluss des Sendeprozesses der physikalische Empfänger auf Transport-Ebene ein "acknowledgement" noch in der gleichen Anschaltung geben.

10.3. Application

<application> <certificate>
Klarstellung:
bei manchen DÜ-Verfahren ist dies die von der annehmenden Stelle vergebene Registration (Zulassung) des auf Client-Seite eingesetzten Produktes
Deshalb wird das Tag unbenannt in: <application> <registrationID>

10.4. Report

<Report>

Wie kann dem Sender die korrekte Bearbeitung bzw. Fehler in den Signatures, bzw. Plug-in mitgeteilt werden?

Antwort: im Header der jeweiligen Ebene in den Responsedetails im Element <Report>

Im Report muss eindeutig erkennbar sein, von welcher Instanz ein Eintrag stammt, z.B. von der Instanz Entschlüsselung, Signatures, Plug-in etc.

Festlegung über geeignete Namenskonventionen müssen noch getroffen werden