

Richtlinien Leistungsstandard: Anforderungen Transmitter

Richtlinien Leistungsstandard-CH (KLE)

Swissdec, 6002 Luzern

www.swissdec.ch

Richtlinien für Leistungsstandard-CH (KLE)
Anforderungen Transmitter

Die Richtlinien für Lohndatenübermittlung wurden in Zusammenarbeit mit folgenden Beteiligten erarbeitet:

- Suva
- Schweizerischer Versicherungsverband

Herausgeber

Swissdec
Postfach 4358
Fluhmattstrasse 1
6002 Luzern
www.swissdec.ch

Inhaltsverzeichnis

1. Einleitung	6
1.1 Ablauf einer Übermittlung.....	6
1.2 Institution und Domäne	7
2. Übersicht Use Cases Transmitter	7
2.1 Übersichtsdiagramme zu den Use Cases	8
2.2 Erläuterungen zu den Use Cases	10
2.3 Tests	10
2.4 Summary Use Cases	11
2.4.1 UC001 Ereignisdeklaration senden.....	11
2.4.2 UC002 Ereignissynchronisation senden	11
2.4.3 UC003 Prozesskontrolle durchführen	11
2.4.4 UC004 Stories melden und quittieren	11
2.4.5 UC005 Stories abholen und verarbeiten	11
2.4.6 UC006 Ereignis schliessen	11
2.4.7 UC007 Datenflusskontrolle durchführen	11
2.4.8 UC008 Testdaten kennzeichnen.....	11
2.4.9 UC009 Security anwenden	11
2.4.10 UC010 Erreichbarkeit prüfen.....	12
2.4.11 UC011 Interoperabilität prüfen	12
2.4.12 UC012 Transmitter konfigurieren	12
2.4.13 UC013 Supportinformationen pflegen.....	12
2.5 Use Cases und zugehörige Operationen	12
3. Use Case 001: Ereignisdeklaration senden.....	13
3.1 Spezielle Anforderungen.....	14
3.1.1 Archiv-Files erstellen.....	14
3.1.2 Adressierung.....	14
4. Use Case 002: Ereignis synchronisieren.....	15
5. Use Case 003: Prozesskontrolle durchführen	16
6. Use Case 004: Stories melden und quittieren.....	18
7. Use Case 005: Stories abholen und verarbeiten.....	19
8. Use Case 006: Ereignis schliessen.....	20
9. Use Case 007: Datenflusskontrolle durchführen.....	21
10. Use Case 008: Testdaten kennzeichnen.....	21
11. Use Case 009: Security anwenden.....	21
12. Use Case 010 Erreichbarkeit prüfen (PIV)	22
13. Use Case 011: Interoperabilität prüfen	23
13.1 Spezielle Anforderungen.....	24
13.1.1 Vorbedingungen.....	24
13.1.2 Nachbedingungen.....	25
14. Use Case 012: Transmitter konfigurieren.....	26
15. UC013 Supportinformationen anzeigen und senden	26
15.1 Zusätzliche Informationen	27
16. Anhang	27
16.1 Referenzen	27

Abbildungsverzeichnis

Abbildung 1: Declare und Synchronize im Leistungsstandard, BPMN Diagramm und Schnittstellen	7
Abbildung 2: Ereignisdeklaration senden	8
Abbildung 3: Ereignissynchronisation senden	8
Abbildung 4: Allgemeine Use Cases	9
Abbildung 5: Beispiel für das Ausfüllen des Job-Elements	14
Abbildung 6: AwaitStory mit Frist	16
Abbildung 7: Available - beim Endempfänger abzuholende Ereignisse	17
Abbildung 8: SuccessType für Quellensteuer	19
Abbildung 9: Use Case 010 Erreichbarkeit prüfen	22
Abbildung 10: Use Case11: Interoperabilität prüfen	23
Abbildung 11: Notifications	27

Tabellenverzeichnis

Tabelle 1: Verbindlichkeit von Anforderungen	5
Tabelle 2: Use Cases und Operationen	12
Tabelle 3: Use Case 001 LM übermitteln	13
Tabelle 4: Use Case 004: Stories melden und quittieren	18
Tabelle 5: Use Case Beschreibung Stories abholen und verarbeiten	19
Tabelle 6: Use Case 006 Ereignis schliessen	20
Tabelle 7: Use Case 10 Erreichbarkeit prüfen	22
Tabelle 8: Use Case Beschreibung Interoperabilität prüfen	23
Tabelle 9: Vorbedingungen (Transmitter)	24
Tabelle 10: Auswertung und Antwort Distributor	25
Tabelle 11: Auswertung Transmitter	25

Übersicht der Änderungen Version 20180901

Richtlinien zur Ereignisdatenübermittlung - Anforderungen für Transmitter, Version 1.0, Ausgabe 20180901 vom 26.09.2018.

Kapitel	Änderung
	Dokument bereinigt und komplettiert.

Konventionen in diesem Dokument

Folgende Schriftarten werden in diesem Dokument verwendet:

Text	Dokumentation
Text	Code
<Text>	XML-Element
[TEXT]	Referenz auf ein anderes Dokument

Die Verbindlichkeit von Anforderungen ist wie folgt definiert:

Verbindlichkeit	Wort
Pflicht	<i>muss</i>
Wunsch	<i>soll (sollte)</i>
Absicht	<i>wird</i>
Vorschlag	<i>kann</i>

Tabelle 1: Verbindlichkeit von Anforderungen

Achtung:

Für das konzeptionelle Verständnis genügen oft ältere Schemabilder, d. h. **verbindlich sind immer nur die offiziellen¹ XML-Files**.

Spezielle Ausdrücke sind im Glossar von (RLID, 2018) erklärt.

¹ www.swissdec.ch

1. Einleitung

Dieses Dokument enthält funktionale, technische und zusätzliche Anforderungen an Transmitter, welche im Rahmen des Leistungsstandard-CH eingesetzt werden. Ein Transmitter wird dazu verwendet, Ereignismeldungen aus einer Unternehmensbuchhaltung an ein oder mehrere Endreceiver elektronisch zu versenden.

1.1 Ablauf einer Übermittlung

Der Prozess startet, indem ein Mitarbeiter eines Unternehmens infolge Krankheit arbeitsunfähig wird, einen Unfall oder einen Rückfall erleidet. Sobald der Mitarbeiter das Unternehmen darüber informiert, wird im Rahmen des Absenzenmanagements unter Berücksichtigung der vertraglichen Regelungen entschieden, ob eine Meldung an den Versicherer notwendig ist. Die zuständige Person löst die Erfassung der Ereignismeldung direkt im Swissdec-zertifizierten ERP-System aus und übermittelt diese an den/die involvierten Versicherer. Sobald der Versicherer die Ereignismeldung empfängt, wird eine Referenz (*InsuranceCaseID*) vergeben und dem Unternehmen zurückgemeldet. Diese dient fortan der Identifikation des Ereignisfalles.

Wird dem Versicherer von Dritten ein Poststück zugestellt und ist kein entsprechendes Ereignis in der Datenbank vorhanden, informiert der Versicherer das Unternehmen, das wiederum im Rahmen des Absenzenmanagements die Anmeldung beim Versicherer prüft.

Ist die Ereignismeldung beim Versicherer angekommen, legt er ein Dossier an oder reaktiviert ein bereits vorhandenes Dossier (Rückfall). Muss der Versicherer zur Stellungnahme weitere Abklärungen tätigen, erteilen das Unternehmen, der Mitarbeiter und allenfalls Dritte (z.B. Arzt) die geforderten Angaben. Sind sämtliche Informationen vorhanden und die Unterlagen vollständig, fällt der Versicherer den Leistungsentscheid und teilt diesen dem Unternehmen und dem Mitarbeiter mit. Besteht ein Leistungsanspruch, beginnt auf Seiten des Versicherers die Fallführung und die Leistungserbringung. Auf Seiten des Unternehmens werden die erhaltenen Taggelder verarbeitet und ereignisspezifische Informationen mit dem Versicherer und Mitarbeiter ausgetauscht. Dabei kann es sich zum Versicherer hin beispielsweise um die Meldung einer Veränderung der Arbeitsunfähigkeit oder um die Klärung von offenen Fragen handeln und zum Mitarbeiter hin um arbeitsplatzbezogene Informationen. Der Mitarbeiter wiederum informiert das Unternehmen und den Versicherer über seine Genesung, Arbeitsunfähigkeit etc.

Der Prozess des Leistungsstandard-CH endet, wenn

- der Mitarbeiter arbeitsfähig oder ausgetreten ist
- der Mitarbeiter verstirbt
- das Unternehmen alle Taggelder erhalten hat
- der Versicherer alle Taggelder bezahlt hat
- die erste Rente ausgerichtet wird
- das Ereignis vom Versicherer abgelehnt wird

Für den obengenannten Spezialfall, dass der Versicherer von Dritten (z.B. Arzt) ein Poststück erhält und kein entsprechendes Ereignis vom Unternehmen gemeldet wurde, gilt folgendes Vorgehen: Der Versicherer muss sich (z. B. telefonisch) an das Unternehmen wenden, damit eine Ereignismeldung erstellt wird. Kommunizieren das Unternehmen und der Versicherer bereits mit Leistungsstandard, kann über diesen Kanal eine Ereignismeldung mittels einer Nachricht verlangt werden.

Für eine detaillierte Beschreibung des Ablaufs wird auf die Richtlinien zum Leistungsstandard (RLID, 2018) verwiesen.

1.2 Institution und Domäne

Wir unterscheiden in diesem Dokument zwischen den Begriffen Domäne und Institution.

Domäne: Organisation, welcher Daten übermittelt werden. Domänen, die der Leistungsstandard-CH unterstützt sind UVG, UVGZ, KU und KTG.

Institution: Empfänger, welcher Daten erhält. Hier handelt es sich um Versicherungen, welche den jeweiligen Domänen angehören.

Eine Firma kann innerhalb einer Domäne mehrere Institutionen kontaktieren. Eine Institution kann mehrere Domänen unterstützen.

2. Übersicht Use Cases Transmitter

Die Ereignisdatenübermittlung erfolgt getrennt in Deklaration und wiederkehrender Synchronisation.

1. Übermittlung der initialen Ereignisdaten (Registrierung)
2. Ergänzung des deklarierten Ereignisses durch Daten in Form sogenannter Stories und gleichzeitig empfangen von Ergebnissen, geändertem Status etc. (Synchronize)

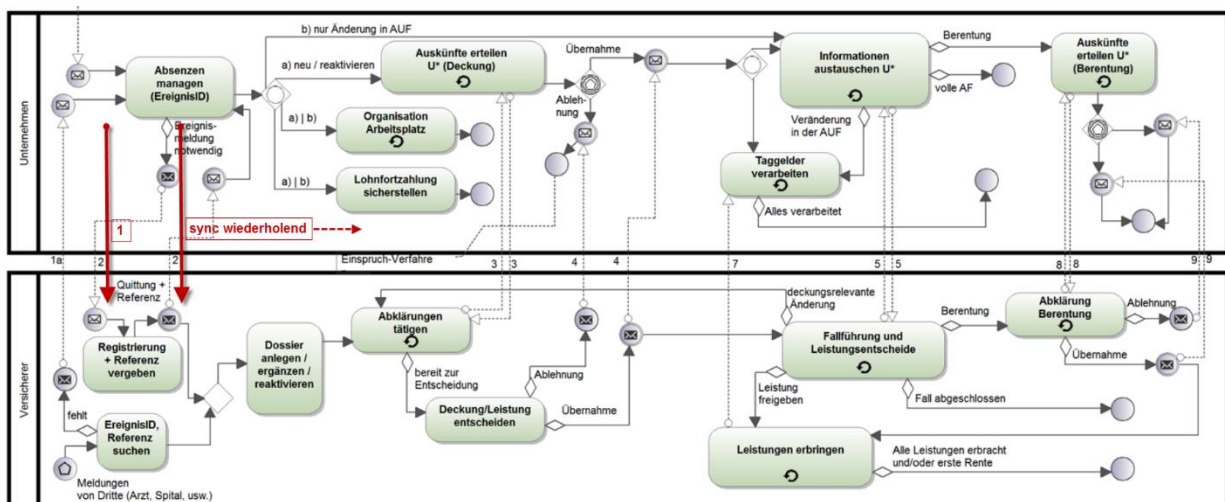


Abbildung 1: Declare und Synchronize im Leistungsstandard, BPMN Diagramm und Schnittstellen

2.1 Übersichtsdiagramme zu den Use Cases

In einem ersten Schritt wird das Ereignis registriert:

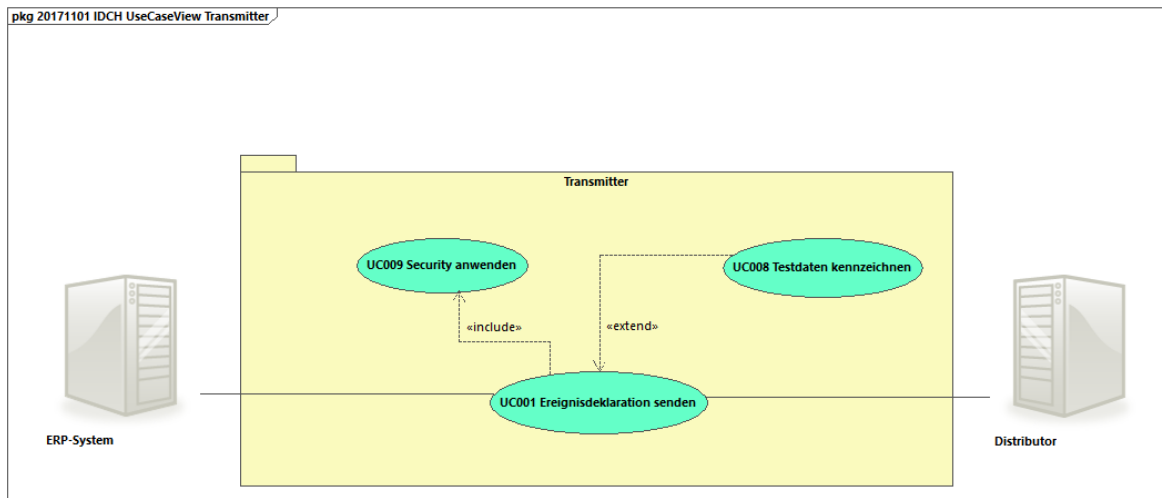


Abbildung 2: Ereignisdeklaration senden

Anschliessend werden diverse Synchronisierungen des Ereignisses durchgeführt: Stories werden gesendet und empfangen, wobei wichtig ist, dass in jedem Fall Prozesskontrolle und Datenflusskontrolle beachtet werden (Siehe die entsprechenden Kapitel).



Abbildung 3: Ereignissynchronisation senden

Des Weiteren gibt es Use Cases für die Installation und Konfiguration des Systems, sowie den Umgang mit Supportinformationen.

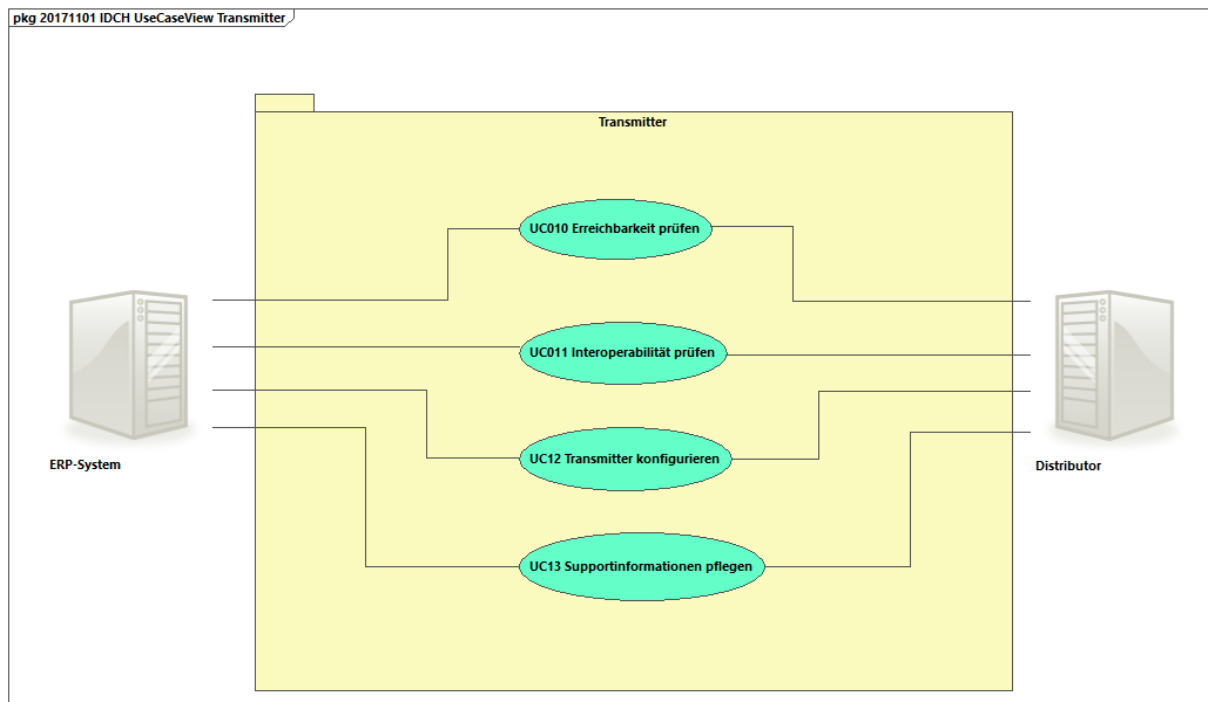


Abbildung 4: Allgemeine Use Cases

2.2 Erläuterungen zu den Use Cases

Die als Use Cases abgebildeten Anforderungen beziehen sich auf den technischen Teil eines Systems aus ERP-System und Transmitter, welcher die elektronische Aufbereitung und Übermittlung von Ereignismeldungen übernimmt.

Fachliche Anforderungen, welche sich auf den spezifischen Inhalt von Ereignismeldungen, Stories etc. beziehen, sind nicht Teil dieser Spezifikation.

Ein ERP-System mit Transmitter *muss* für die Zertifizierung immer die folgenden Systemanforderungen erfüllen:

- UC001 Ereignisdeklaration senden
- UC002 Ereignissynchronisation senden
- UC003 Prozesskontrolle durchführen
- UC004 Stories melden und quittieren
- UC005 Stories und Quittungen abholen und verarbeiten
- UC006 Ereignis schliessen
- UC007 Datenflusskontrolle durchführen
- UC008 Testdaten kennzeichnen
- UC009 Security anwenden
- UC010 Erreichbarkeit prüfen
- UC011 Interoperabilität prüfen
- UC012 Transmitter konfigurieren
- UC013 Supportinformationen pflegen

Wie die Interaktion zwischen Benutzer und System gestaltet wird, liegt in der Entscheidung der Systemhersteller und wird in dieser Spezifikation nicht beschrieben.

2.3 Tests

Die Tests der Zertifizierung beziehen sich auf die Use Cases. Zusammen mit den Anforderungen tragen sie zum Gesamtverständnis des zu bauenden Systems bei. Die Tests werden mit Vorteil bereits während der Entwicklung vom Hersteller einbezogen (Test Driven Development).

2.4 Summary Use Cases

2.4.1 UC001 Ereignisdeklaration senden

Ein neues Ereignis (Unfallmeldung, etc.) wird via Distributor an einen oder mehrere Endreceiver gesendet und die Antwort ausgewertet. Die Antwort vom Distributor wird gesichert, vgl. Kap. 3 "Use Case 001: ".

Das Ereignis besitzt nun eine CompanyCaseID (Ereignis-Identifikation des Unternehmens), eine InsuranceCaseID (Ereignis-Identifikation des Endempfängers) und eine IncidentCaseID (gemeinsame Ereignis-Identifikation für beide beteiligten Systeme).

2.4.2 UC002 Ereignissynchronisation senden

Nach dem Registrieren einer Meldung (UC001) kann das gemeldete Ereignis nun synchronisiert werden. Dies beinhaltet folgende Varianten: Stories melden und quittieren (UC004), Stories abholen und verarbeiten (UC005), sowie Ereignis schliessen (UC006).

Grundsätzlich geht es darum, dass beide Systeme über dieselben Informationen zum Fall verfügen. Aus diesem Grund *muss* stets die Prozesskontrolle durchgeführt werden, um zu prüfen, ob der Endempfänger noch Informationen erwartet. (UC003). Zu beachten ist auch, dass es mit Hilfe der Datenflusskontrolle (UC007) möglich ist, die Datenmenge anzupassen, um die synchrone Verbindung nicht zu überlasten.

2.4.3 UC003 Prozesskontrolle durchführen

Der Endempfänger benötigt zum Verarbeiten eines Ereignisses verschiedene Informationen, die er beim ERP-System anfordern kann. Dazu verwendet er das Element `<AwaitStory>` im Bereich der Prozesskontrolle (`<ProcessCtrl>`). Das ERP-System muss stets präsent haben, welche Stories vom Endempfänger noch erwartet werden, auch wenn diese noch nicht zum Versenden bereit sind.

2.4.4 UC004 Stories melden und quittieren

Beim Synchronisieren eines Ereignisses können vom ERP-System neue Stories gemeldet werden. Hier handelt es sich um Informationen, die der Endreceiver benötigt, und die das ERP-System ihm zur Verfügung stellen kann. Bereits vom Endreceiver erhaltene Stories *müssen* ausserdem quittiert werden.

Nach dem Melden der Stories muss die Antwort des Receivers daraufhin überprüft werden, ob weitere Stories zur Abholung bereit sind (Datenflusskontrolle, UC007). Ebenfalls muss geprüft werden, ob der Endempfänger noch die Lieferung von Stories erwartet (Prozesskontrolle, UC006).

2.4.5 UC005 Stories abholen und verarbeiten

Nach dem Melden des Ereignisses kann das ERP-System durch Synchronisierung des Ereignisses feststellen, ob der Endempfänger über Stories verfügt, die er dem ERP-System zu melden hat. Ist dies der Fall, können die Stories vom ERP-System abgeholt und verarbeitet werden. Dabei ist auf die Datenflusskontrolle zu achten (UC 007). Ebenfalls kann der Endempfänger dem ERP-System mitteilen, welche Stories er noch erwartet (Prozesskontrolle, UC006).

2.4.6 UC006 Ereignis schliessen

Ist ein Ereignis in den Augen des ERP-Benutzers abgeschlossen, kann er den Endreceiver darüber informieren, dass das Ereignis geschlossen werden kann. Dieser entscheidet, ob die Schliessung auch von seiner Seite aus möglich ist, und schliesst in diesem Fall das Ereignis ab.

2.4.7 UC007 Datenflusskontrolle durchführen

Sind mehrere Ereignisse bei einer Institution gemeldet, und es werden grosse Datenmengen verschoben, so kann das ERP-System die Menge an zu übermittelnden Daten steuern, um eine Überlastung des synchronen Systems zu verhindern und die Gefahr eines Timeouts zu verringern. Auch der Endempfänger kann mit der Datenflusskontrolle die Datenmenge kontrollieren, die er zu empfangen und zu versenden in der Lage ist.

2.4.8 UC008 Testdaten kennzeichnen

Eine beliebige Meldung kann als Testfall gekennzeichnet werden. Sie wird somit über das produktive System versendet, vom Endreceiver jedoch nicht produktiv verarbeitet.

2.4.9 UC009 Security anwenden

Jede übermittelte Meldung muss doppelt signiert und verschlüsselt sein.

2.4.10 UC010 Erreichbarkeit prüfen

Eine spezielle Meldung wird via Internet an den Distributor gesendet um zu prüfen, ob dieser erreichbar ist.

2.4.11 UC011 Interoperabilität prüfen

Eine spezielle Meldung wird an den Distributor gesendet, um die Interoperabilität (z.B. Encoding, Marshalling, Zeitangaben etc.) zwischen Transmitter und Distributor zu prüfen.

2.4.12 UC012 Transmitter konfigurieren

Um korrekt an die Endempfänger übermitteln zu können, ist es wichtig, dass das ERP-System stets die aktuellen Versicherungsprofile der Endempfänger eingepflegt hat. Auch muss der Digitalisierungs-Scope des Endempfängers bekannt sein.

2.4.13 UC013 Supportinformationen pflegen

Sämtliche Supportinformationen (Notifications, Faults) müssen dem Endbenutzer klar verständlich dargestellt werden. Der Benutzer muss wissen, woher die Meldung kommt, und wie er darauf zu reagieren hat.

2.5 Use Cases und zugehörige Operationen

Das zugrundeliegende Modell ist ein Client – Server System mit dem Transmitter als Client. Verwendet werden die XML-Standards WSDL und XML-Schema. Die nachfolgenden Operationen und Elemente befinden sich im zugehörigen WSDL-File (WSDLID, 2018) und im beschreibenden Schema (XSDID, 2018). Verfahren und Protokoll sind in (RLID, 2018) erläutert.

Use Case	Operation / Element
UC001 Ereignisdeklaration senden	<ul style="list-style-type: none">▪ DeclareIncident▪ DeclareIncidentResponse▪ IncidentDeclarationFault
UC002 Ereignissynchronisation senden	<ul style="list-style-type: none">▪ SynchronizeIncident▪ SynchronizeIncidentResponse▪ IncidentDeclarationFault
UC009 Erreichbarkeit prüfen	<ul style="list-style-type: none">▪ Ping▪ PingResponse
UC010 Interoperabilität prüfen	<ul style="list-style-type: none">▪ CheckInteroperability▪ CheckInteroperabilityResponse

Tabelle 2: Use Cases und Operationen

3. Use Case 001: Ereignisdeklaration senden

Use Case Diagramm: siehe Abbildung 2: Ereignisdeklaration senden auf Seite 8.

Kurzbeschreibung	Eine elektronische Ereignismeldung <i>muss</i> an einen oder mehrere Endempfänger versendet werden. Die Rückantwort der Endempfänger wird ausgewertet und abgelegt. Ein Archiv-File der gesendeten Meldung wird ebenfalls gesichert.
Akteure	ERP-System, Distributor, Endreceiver
Auslöser	Ein Angestellter des Unternehmens ist von einem Ereignis betroffen, welches versicherungstechnisch relevant ist.
Vorbedingungen	Das ERP-System ist in der Lage, elektronische Ereignismeldungen zu versenden und zu empfangen.
Nachbedingungen	<ul style="list-style-type: none"> Die Ereignismeldung wurde vom Endreceiver empfangen und durch eine Rückantwort quittiert. Bei einem Fehlschlag: <ul style="list-style-type: none"> Fehlermeldung
Included Use Cases	UC009 Security anwenden
Standardablauf	<p>vgl. Abbildung 1: auf Seite 7.</p> <ol style="list-style-type: none"> Das ERP-System übergibt dem Transmitter die Ereignisdaten mit den Empfängeradressen. Der Transmitter bereitet die Meldung als SOAP-Request mit zugehöriger Adressierung (Job) auf. Die Meldung wird nach Spezifikation doppelt signiert und verschlüsselt. (UC009) Der Transmitter sendet die aufbereitete und signierte Meldung über SSL, an den Distributor. Der Distributor prüft die Ereignismeldung auf Validität und Plausibilität. Der Distributor bereitet eine oder mehrere Meldungen für die gewählten Endempfänger auf und sendet diese an den/die Endreceiver. Der Endreceiver prüft die Ereignismeldung und startet die Verarbeitung des Jobs. Der Transmitter wertet die Rückantwort des Distributors ab. Das Ergebnis der Job-Verarbeitung wird auf Transmitterseite aufbereitet und angezeigt.
Alternative Abläufe	<p>{UC008} Daten als Testdaten versenden</p> <p>{nach Schritt 1}</p> <ol style="list-style-type: none"> b) Die Meldung wird als Testmeldung gekennzeichnet. (Ein Element TestCase wird in die Meldung eingefügt). <p>{weiter mit Schritt 2}</p> <p>ACHTUNG: Wird ein Ereignis als Testfall gemeldet, müssen auch alle weiteren Synchronisierungen als Testfall markiert sein (UC002, UC008)</p>
Fehlerliste	<p>Fachliche Fehler:</p> <ul style="list-style-type: none"> die Meldung verstösst gegen die Plausibilisierungsregeln <p>Technische Fehler:</p> <ul style="list-style-type: none"> Fehler beim Signieren oder Verschlüsseln der Endreceiver ist nicht erreichbar die vom ERP-System aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben)

Tabelle 3: Use Case 001 LM übermitteln

3.1 Spezielle Anforderungen

3.1.1 Archiv-Files erstellen

Mit dieser Anforderung wird sichergestellt, dass eine Kopie jeder gesendeten und empfangenen Meldung gesichert wird. Die Daten müssen zu einem SOAP-Request aufbereitet und als XML-Instanzdokument abgelegt werden. Archivdateien müssen signiert sein, dürfen aber nicht verschlüsselt sein.

3.1.2 Adressierung

Vor dem Senden eines XML-Dokuments muss im IncidentDeclarationContainer angegeben werden, welche Institution die Datei auf welchem Übermittlungsweg erhalten sollen. Dazu werden die Institutionen unter dem Element `<Job>` aufgelistet. Das Element `<ProcessByDistributor>` bestimmt, ob die betreffende Institution vom Distributor Daten erhalten soll oder nicht.

```
<Job xmlns="http://www.swissdec.ch/schema/id/20171101/IncidentDeclarationContainer">
  <Addressees>
    <UVG-LAA institutionIDRef="#UVG-LAA">
      <ProcessByDistributor>true</ProcessByDistributor>
    </UVG-LAA>
    <UVGZ-LAAC institutionIDRef="#UVGZ-LAAC">
      <ProcessByDistributor>true</ProcessByDistributor>
    </UVGZ-LAAC>
  </Addressees>
</Job>
```

Abbildung 5: Beispiel für das Ausfüllen des Job-Elements

Eine Institution erhält Daten vom Distributor, wenn sie

- Im `<Job>` gelistet ist
- `<ProcessByDistributor>` auf `'true'` steht

Ist die Institution nicht im Job gelistet, wird sie vom Distributor komplett ignoriert und alle auf diese Institution bezogenen Daten werden kommentarlos verworfen.

Steht die Institution unter `<ProcessByDistributor>` auf `'false'`, wird sie ebenfalls vom Distributor ignoriert, dieser sendet jedoch den Status "ignored" zurück.

4. Use Case 002: Ereignis synchronisieren

Nach jeder Ereignisdeklaration ist mit dem Aufruf `<SynchronizeIncident>` das Ereignis zu synchronisieren. Dies *kann*, muss aber nicht, folgende Elemente beinhalten:

- UC003 Stories melden
- UC004 Stories abholen
- UC005 Stories quittieren
- UC007 Ereignis schliessen

Wie es der Name sagt, werden zwischen Transmitter und Endempfänger Informationen ausgetauscht mit dem Ziel, die Beteiligten auf denselben Wissensstand zu bringen. Die Informationen werden also synchronisiert. Bei der Übermittlung von Informationen spricht man von sogenannten Stories, die zwischen den Beteiligten ausgetauscht werden.

Was die Synchronisierung im Einzelnen für ein Ziel hat, hängt von den gesendeten oder angeforderten Stories ab.

Kurzbeschreibung	Das Ereignis wird synchronisiert.
Akteure	ERP-System, Distributor, Endreceiver
Auslöser	Der Akteur will mit dem Empfänger Stories austauschen.
Vorbedingungen	<ul style="list-style-type: none"> ▪ Die Ereignismeldung wurde erfolgreich an den Endempfänger gesendet ▪ CompanyCaseID, InsuranceCaseID und IncidentCaseID sind bekannt
Nachbedingungen	Das Ereignis wurde zwischen Absender und Empfänger synchronisiert. Falls die Synchronisierung nicht abgeschlossen ist (der Endempfänger hat noch weitere Stories zu liefern), ist der Absender darüber in Kenntnis gesetzt worden.
Included Use Cases	UC003 Prozesskontrolle durchführen UC009 Security anwenden
Standardablauf	<ol style="list-style-type: none"> 1. Der Akteur wählt die Stories aus, die er übermitteln will und sendet diese dem Endempfänger. 2. Dieser wertet die Stories aus und meldet zurück, was bei ihm noch an offenen Stories vorhanden ist und/oder was der Akteur von ihm verlangt hat. 3. Der Akteur analysiert die erhaltene Antwort, überprüft die empfangenen Stories und pflegt alle relevanten Daten ins ERP-System ein.
Fehlerliste	<p>Fachliche Fehler:</p> <ul style="list-style-type: none"> ▪ die Meldung verstösst gegen die Plausibilisierungsregeln <p>Technische Fehler:</p> <ul style="list-style-type: none"> ▪ Fehler beim Signieren oder Verschlüsseln ▪ der Endreceiver ist nicht erreichbar ▪ die vom ERP-System aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben)

5. Use Case 003: Prozesskontrolle durchführen

Bei jeder Synchronisierung *muss* das ERP-System den laufenden Prozess kontrollieren, d.h. überprüfen, was auf beiden Seiten noch offen ist:

- Welche Stories werden vom Endempfänger noch erwartet (AwaitStory)
- Hat der Endempfänger noch Stories, die er dem ERP-System liefern muss (Available)
- Sind weitere Änderungen notwendig?

Das ERP-System *muss* dem Benutzer aufzeigen, welche Stories er dem Endempfänger liefern muss und bis wann, falls unter der entsprechenden Story eine Frist gesetzt wurde.

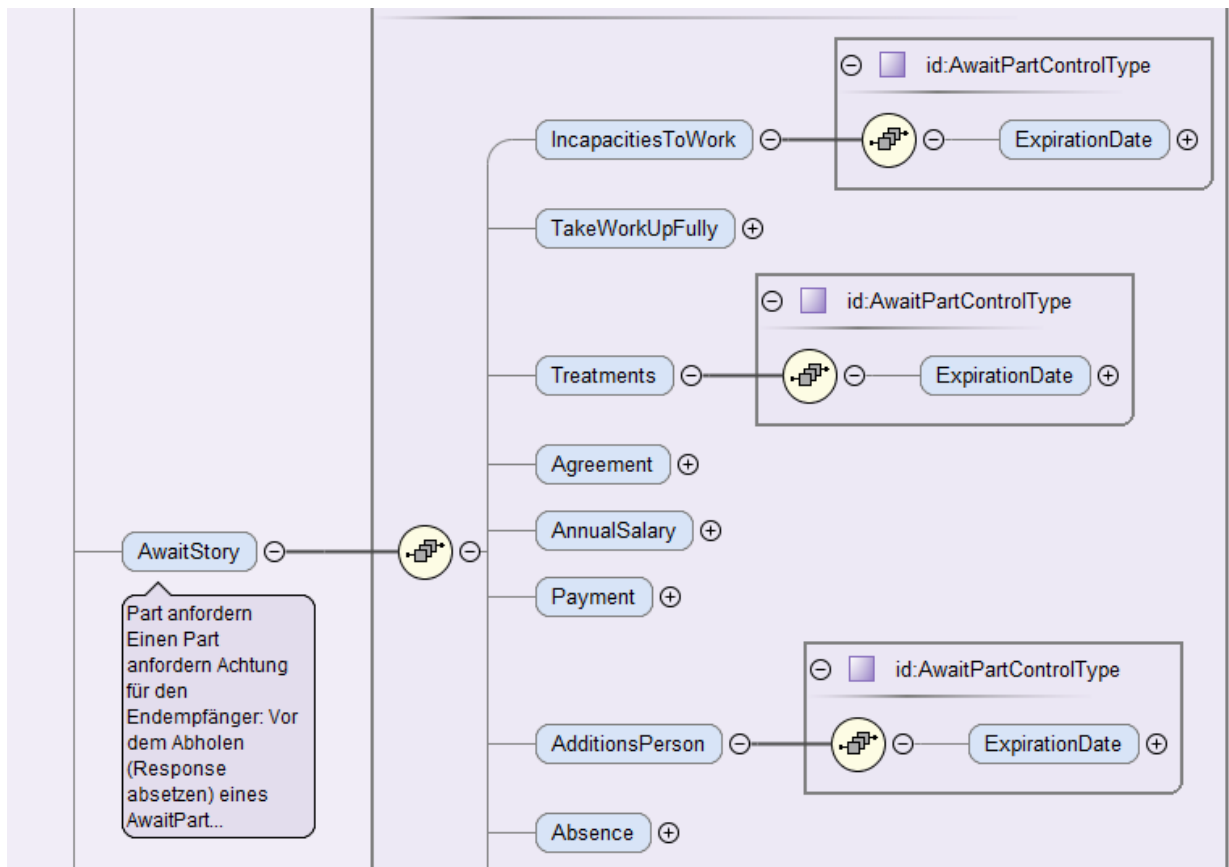


Abbildung 6: AwaitStory mit Frist

Ebenfalls **muss** dem Benutzer klar aufgezeigt werden, für welches Ereignis noch Stories abgeholt werden müssen.

Weitere Einzelheiten zur Prozesskontrolle können den Richtlinien für den Leistungsstandard (RLID, 2018) entnommen werden.

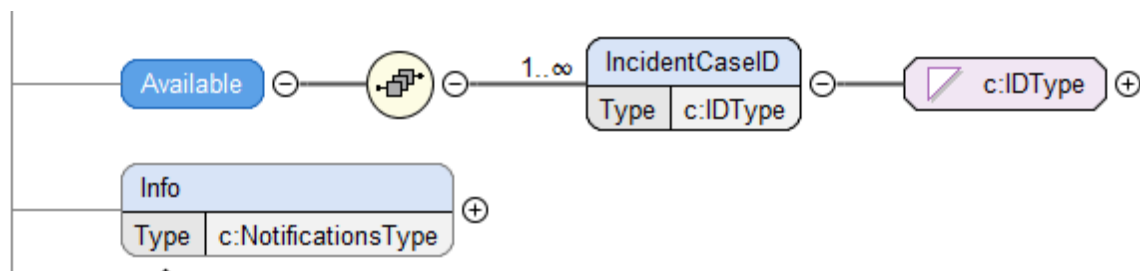


Abbildung 7: Available - beim Endempfänger abzuholende Ereignisse

6. Use Case 004: Stories melden und quittieren

Kurzbeschreibung	Das ERP-System meldet eine oder mehrere Stories an den Endempfänger
Akteure	ERP-System, Distributor, Endempfänger
Auslöser	Der Akteur hat Informationen, die er an den Endempfänger weiterleiten will.
Vorbedingungen	<ul style="list-style-type: none"> Das Ereignis wurde dem Endempfänger erfolgreich gemeldet DeclarationID, Reference und IncidentID sind bekannt und korrekt gesetzt
Nachbedingungen	<ul style="list-style-type: none"> Der Endempfänger hat die Stories empfangen
Included UCs	UC002, UC003
Standardablauf	<ol style="list-style-type: none"> Im ERP-System wird ausgewählt, welche Stories gemeldet werden sollen. Ausserdem werden alle vom Endempfänger erhaltenen Stories quittiert. (RLID, 2018) Es wird ein SynchronizeIncidentRequest aufbereitet, welcher die ausgewählten Informationen beinhaltet. Das Ereignis wird synchronisiert (UC002)
Alternative Abläufe	<p>Keine Stories zu quittieren</p> <ol style="list-style-type: none"> Es werden nur neue Stories gemeldet, da keine erhaltenen Stories zu quittieren sind. Dann weiter mit 2. <p>Keine Stories zu melden</p> <ol style="list-style-type: none"> Das ERP-System quittiert nur erhaltene Stories, hat dabei aber keine neuen Stories zu melden. Der Request enthält nur einen aktualisierten IncidentContext aber keine weiteren Stories. Dann weiter mit 2.
Fehlerliste	<ol style="list-style-type: none"> Es gibt Probleme bei der Übermittlung. Die Übermittlung <i>kann</i> wiederholt werden. Es gibt fachliche Probleme bei der Auswahl der Stories. (RLID, 2018)

Tabelle 4: Use Case 004: Stories melden und quittieren

7. Use Case 005: Stories abholen und verarbeiten

Kurzbeschreibung	Das ERP-System holt Stories ab, die beim Endempfänger bereit liegen und verarbeitet sie im System.
Akteure	ERP-System, Distributor, Endempfänger
Auslöser	Dem Akteur stehen Informationen zur Verfügung, die er beim Endempfänger abholen muss.
Vorbedingungen	<ul style="list-style-type: none"> Das Ereignis wurde dem Endempfänger erfolgreich gemeldet DeclarationID, Reference und IncidentID sind bekannt und korrekt gesetzt Der Endempfänger hat dem Akteur mitgeteilt, dass Stories zur Abholung bereit liegen (Available)
Nachbedingungen	<ul style="list-style-type: none"> Die vom Endempfänger bereitgestellten Stories wurden abgeholt Die erhaltenen Informationen wurden ins System eingepflegt
Included UCs	UC002, UC003
Standardablauf	<ol style="list-style-type: none"> Der Akteur synchronisiert das Ereignis, von dem der Endempfänger mitgeteilt hat, dass noch Stories abzuholen sind. (UC002) Der Akteur liest aus der <code>SynchronizeIncidentResponse</code> die erhaltenen Stories aus und pflegt sie ins ERP-System ein. (RLID, 2018)
Fehlerliste	<p>fachliche Fehler:</p> <ul style="list-style-type: none"> Es werden keine Stories zurückgegeben. Die Stories können nicht verarbeitet werden. <p>Siehe Richtlinien für den Leistungsstandard (RLID, 2018)</p> <p>Technische Fehler:</p> <ul style="list-style-type: none"> Es gibt Probleme bei der Übermittlung. Die Übermittlung kann wiederholt werden.

Tabelle 5: Use Case Beschreibung Stories abholen und verarbeiten

8. Use Case 006: Ereignis schliessen

Kurzbeschreibung	Ist ein Ereignis seitens ERP-System abgeschlossen, kann der Benutzer des ERP-Systems die betroffenen Endempfänger darüber informieren, dass das Ereignis geschlossen werden kann. Die Endempfänger entscheiden, ob das Ereignis auch für sie abgeschlossen ist und schliessen es, falls dies der Fall ist.
Akteure	ERP-System, Distributor, Endempfänger
Auslöser	Seitens ERP ist ein Ereignis abgeschlossen.
Vorbedingungen	<ul style="list-style-type: none"> Es besteht ein Ereignis, das ERP-System und Endempfänger betrifft. Der Endempfänger hat keine Stories mehr zu liefern (<Available>) Der Endempfänger erwartet keine Stories mehr (<AwaitStory>)
Nachbedingungen	<ul style="list-style-type: none"> Das Ereignis wurde von beiden Seiten (ERP-System und Endempfänger) geschlossen. Das Ereignis wurde vom ERP-System archiviert.
Included UseCases	UC002, UC003
Standardablauf	<ol style="list-style-type: none"> Das ERP-System synchronisiert das zu schliessende Ereignis (UC002), wobei <IncidentForCompanyClosed> in der Prozesskontrolle ausgewählt ist. Dabei kann angegeben werden, ob das Ereignis normal beendet oder abgebrochen wird. Das ERP-System bearbeitet die Antwort vom Endempfänger gemäss dem normalen Ablauf (UC005). Das ERP-System archiviert das Ereignis.
Alternative Abläufe	<p>Der Endempfänger verlangt weitere Stories {nach Schritt 2}</p> <p>3 b) Das ERP-System lässt das Ereignis offen und bearbeitet es mit weiteren Synchronisierungen, bis es erneut geschlossen werden kann. (UC002)</p>
Fehlerliste	<p>Technische Fehler gemäss UC002.</p> <p>Fachliche Fehler gemäss den Richtlinien zum Leistungsstandard (RLID, 2018)</p>

Tabelle 6: Use Case 006 Ereignis schliessen

9. Use Case 007: Datenflusskontrolle durchführen

Die Datenflusskontrolle soll eine Überlastung der betroffenen Systeme einer Übermittlung verhindern.

Der Endempfänger kann von sich aus beeinflussen, wie viele Informationen er innerhalb einer Response senden will, um die Auslastung seines Systems zu optimieren. Um die Infrastruktur eines schwächeren ERP-Systems allerdings nicht zu überlasten, sind diesem die Möglichkeiten geboten, den Umfang einer Response einzuschränken.

Der Endempfänger teilt dem ERP-System jeweils mit, zu welchen Ereignissen er noch Informationen zu liefern hat (<Available>). Das ERP-System ist jedoch nicht verpflichtet, diese alle innerhalb des nächsten Requests abzurufen. Vielmehr besteht die Möglichkeit, den Umfang des Datenflusses zu kontrollieren, indem nur eines oder wenige Ereignisse zur Synchronisation ausgewählt werden. Der Endempfänger reagiert darauf, indem er die Antworten zu diesen Ereignissen liefert und für jene, die nicht abgefragt wurden, weiterhin mit <Available> reagiert.

Der Transmitter *muss* dem Benutzer die Möglichkeit geben, aus der Liste der verfügbaren Ereignisse jene auszuwählen, welche bei der nächsten Synchronisation abgefragt werden sollen.

10. Use Case 008: Testdaten kennzeichnen

Bei der Registrierung eines Ereignisses ist es möglich, dieses als Testfall zu kennzeichnen. Dies geschieht, indem das Element <TestCase> an entsprechender Stelle (gemäss Schema) in die XML-Instanz eingefügt wird. Das Ereignis wird vom Distributor normal verarbeitet, vom Endempfänger aber als Testfall behandelt.

Jegliche weiteren Aufrufe in Bezug auf dieses Ereignis *müssen* ebenfalls als Testfall markiert sein (*SynchronizeIncident*)

Es darf keine Mischformen in der Übermittlung geben: Was als Testfall beginnt, muss als Testfall beendet werden. Was produktiv registriert wurde, darf nicht als Testfall synchronisiert werden.

Wichtig ist hierbei, dass sich die Testfall-Markierung nicht auf eine Übermittlung, sondern auf ein einzelnes Ereignis bezieht. Es ist also möglich und wahrscheinlich, dass in einer Synchronisierung gleichzeitig produktive und Testfall-Ereignisse übermittelt werden können.

11. Use Case 009: Security anwenden

Ausser dem Erreichbarkeitstest *muss* jede Übermittlung doppelt signiert und verschlüsselt werden. Einzelheiten dazu finden sich in den Dokumenten zur Sicherheit auf Transmitterseite (SECTID, 2018), sowie in den zusätzlichen Informationen zur doppelten Signierung (SUA).

12. Use Case 010 Erreichbarkeit prüfen (PIV)

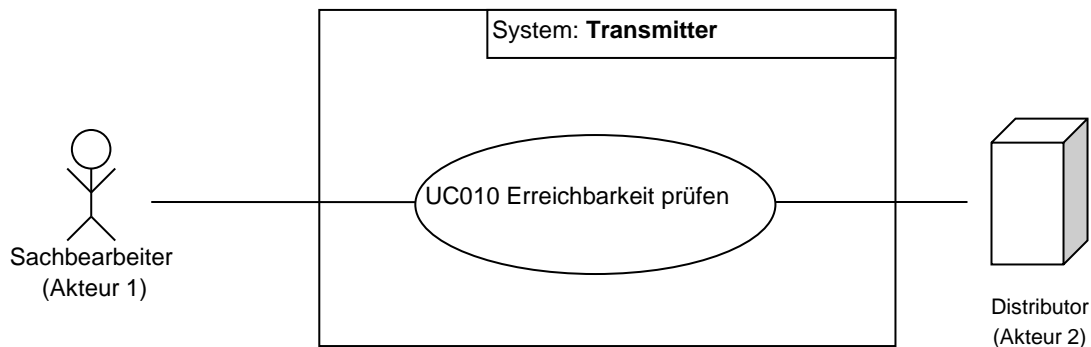


Abbildung 9: Use Case 010 Erreichbarkeit prüfen

Kurzbeschreibung	Die Erreichbarkeit des Distributors <i>muss</i> geprüft werden. Dazu wird eine einfache Anfrage (WSDLID, 2018) an den Distributor geschickt. Die Rückantwort des Distributors bestätigt die Erreichbarkeit.
Akteure	Akteur 1: Sachbearbeiter, Akteur 2: Distributor
Auslöser	Die Erreichbarkeit des Distributors soll geprüft werden.
Vorbedingungen	Keine
Nachbedingungen	<ul style="list-style-type: none"> Die Rückantwort des Distributors enthält einen Timestamp mit der Systemzeit des Distributors (WSDLID, 2018). Bei einem Fehlschlag: <ul style="list-style-type: none"> Distributor nicht erreichbar: Fehlermeldung Inhalt verschieden (WSDLID, 2018) (ACKNSwissdec, 2018): Fehlermeldung
Included Use Cases	-
Standardablauf	<ol style="list-style-type: none"> Der Akteur löst die Überprüfung aus. Der Transmitter sendet eine einfache Serveranfrage (Ping) an die Zieladresse des Distributors Der Transmitter wertet die Rückantwort des Distributors aus
Alternative Abläufe	Distributor nicht erreichbar {nach Schritt 1} <ol style="list-style-type: none"> Eine Fehlermeldung wird angezeigt. {Ende}
Fehlerliste	technische Fehler: <ul style="list-style-type: none"> der Distributor ist nicht erreichbar der Distributor sendet eine falsche Antwort

Tabelle 7: Use Case 10 Erreichbarkeit prüfen

Mit dem Ping-Aufruf wird die Systemzeit übermittelt, so dass es möglich ist, die Zeiten von Distributor und Absender zu vergleichen. Damit lassen sich Timestamp-Probleme aufdecken.
Dieser UseCase dient der Qualitätssicherung bei der Installation und der Entwicklung.

13. Use Case 011: Interoperabilität prüfen

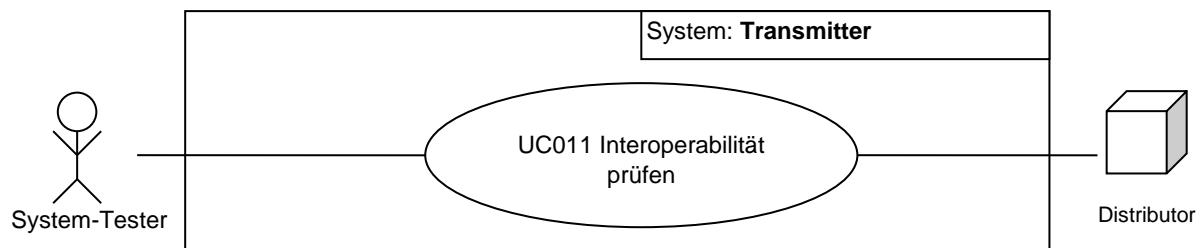


Abbildung 10: Use Case11: Interoperabilität prüfen

Kurzbeschreibung	Damit die Interoperabilität zwischen einem Transmitter und dem Distributor überprüft werden kann, <i>muss</i> der Transmitter einen «CheckInteroperabilityRequest» (WSDLID, 2018) absetzen können.
Akteure	System-Tester, Distributor
Auslöser	Installation soll getestet werden.
Vorbedingungen	Keine
Nachbedingungen	Die Übermittlung war erfolgreich, die Resultate entsprechen den Erwartungen.
Included Use Cases	-
Standardablauf	<ol style="list-style-type: none"> 1. Der Akteur startet die Interoperabilitätsprüfung und gibt Werte für Operand 2 ein. 2. Der Akteur löst das Senden der Daten aus. 3. Der Transmitter bereitet die Serveranfrage auf. 4. Die Meldung wird mit dem privaten Schlüssel/Zertifikat des Herstellers und mit der Unternehmensidentifikation nach Spezifikation (SECTID, 2018) signiert. 5. Der Transmitter sendet die Serveranfrage ssl-verschlüsselt an den Distributor. 6. Der Distributor bearbeitet die gesendeten Daten (Transformation Umlautstring, Berechnung «FirstOperand +- SecondOperand») und schickt die Antwort an den Transmitter. 7. Der Transmitter wertet die Antwort des Distributors aus. 8. Der Transmitter zeigt die Antwort des Distributors an.
Fehlerliste	<p>Fachliche Fehler:</p> <ul style="list-style-type: none"> ▪ Interoperabilität ist nicht gegeben <p>Technische Fehler:</p> <ul style="list-style-type: none"> ▪ Fehler beim Signieren ▪ Fehler beim ver/ -entschlüsseln ▪ der Distributor ist nicht erreichbar

Tabelle 8: Use Case Beschreibung Interoperabilität prüfen

13.1 Spezielle Anforderungen

Der Interoperabilitätstest wird zu Entwicklungszwecken und bei der Installation verwendet, um die Interoperabilität zwischen einem Transmitter und dem Distributor zu gewährleisten.

Die grössten zu erwartenden Schwierigkeiten liegen dabei in den Bereichen Codierung von Zeichenketten (Encoding) und Interpretation von Fließkommazahlen.

Ausserdem erlaubt der Interoperabilitätstest einen einfachen und schnellen Security-Check. Beide Systeme (Transmitter und Distributor) müssen dabei bestimmte Auswertungen vornehmen, um bei einem eventuellen Fehler auf den Verursacher schliessen zu können.

Die Parameter in den folgenden Tabellen sind in (WSDLID, 2018) ersichtlich.

13.1.1 Vorbedingungen

Der Transmitter sendet folgende Daten:

Parametername	Wert	Bemerkungen
UmlautString	ÄËÖÜÄËÖÜÄËÖÜÄËÖÜ	fester Wert
FirstOperand	999000000000.00	fester Wert, 999 milliards
SecondOperand	keine Vorgabe	beliebige Fließkommazahl
SystemDateTime	Datum und Zeit des Transmitters	Systemdatum und –zeit

Tabelle 9: Vorbedingungen (Transmitter)

13.1.2 Nachbedingungen

Auswertung und Antwort des Distributors:

Parametername	Auswertung / Berechnung	Bemerkungen
UmlautStringsCorrect	$\text{UmlautString}_{\text{TRANS}} = \text{ÄÖÜÄÉÓÚÀÈÒÙÂÊÔÛ}$	Rückgabe: true / false
FirstOperandIsCorrect	$\text{FirstOperand}_{\text{TRANS}} = 999000000000.00$	Rückgabe: true / false
UmlautString	äëöüáéóúàèòùâêôû	Rückgabe: $\text{UmlautString}_{\text{DISTR1}}$ Gross- zu Kleinbuchstaben.
AdditionResult	$\text{AdditionResult}_{\text{DISTR1}} = \text{FirstOperand}_{\text{TRANS}} + \text{SecondOperand}_{\text{TRANS}}$	Rückgabe: berechneter Wert $\text{AdditionResult}_{\text{DISTR1}}$
SubstractionResult	$\text{SubstractionResult}_{\text{DISTR1}} = \text{FirstOperand}_{\text{TRANS}} - \text{SecondOperand}_{\text{TRANS}}$	Rückgabe: berechneter Wert $\text{SubstractionResult}_{\text{DISTR1}}$
SystemDateTime	Datum und Zeit des Distributors	Rückgabe: Systemdatum und -zeit

Tabelle 10: Auswertung und Antwort Distributor

Auswertung des Transmitters:

Parametername	Auswertung / Berechnung	Bemerkungen
UmlautStringsCorrect	$\text{UmlautStringsCorrect} = \text{true}$	muss true sein
FirstOperandIsCorrect	$\text{FirstOperandIsCorrect} = \text{true}$	muss true sein
UmlautString	$\text{UmlautString}_{\text{DISTR1}} = \text{äëöüáéóúàèòùâêôû}$	muss äëöüáéóúàèòùâêôû sein
AdditionResult	$\text{FirstOperand}_{\text{TRANS}} + \text{SecondOperand}_{\text{TRANS}} = \text{AdditionResult}_{\text{DISTR1}}$	Berechnung und Vergleich, Genauigkeitsgrad 2 Nachkommastellen
SubstractionResult	$\text{FirstOperand}_{\text{TRANS}} - \text{SecondOperand}_{\text{TRANS}} = \text{AdditionResult}_{\text{DISTR1}}$	Berechnung und Vergleich, Genauigkeitsgrad 2 Nachkommastellen
SystemDateTime	$ \text{SystemDateTime}_{\text{DISTR1}} - \text{SystemDateTime}_{\text{TRANS}} < 1\text{h}$	Betrag Zeitdifferenz sollte < 1 Stunde sein

Tabelle 11: Auswertung Transmitter

14. Use Case 012: Transmitter konfigurieren

Das ERP-System ist für die korrekte Adressierung der Endempfänger im Transmitter verantwortlich. Hierbei handelt es sich um die Pflege und regelmässige Aktualisierung der Versicherungsprofile mit gültiger Identifikation der einzelnen Institutionen.

Die jeweils gültigen Profile können entweder direkt beim Versicherer bezogen werden oder stehen auf der Webseite der Swissdec (<http://www.swissdec.ch>) zum Download bereit.

Auf welche Weise und wie regelmässig die Versicherungsprofile aktualisiert werden, liegt in der Verantwortung des ERP-Herstellers oder des Endbenutzers.

15. UC013 Supportinformationen anzeigen und senden

Kurzbeschreibung	Fehler, Warnungen und Informationen gemäss (ACKNSwissdec, 2018) <i>müssen</i> ausgewertet und dem Benutzer angezeigt und/oder dem Endempfänger mitgeteilt werden. IDs <i>müssen</i> verwendet werden.
Akteure	Lohnbuchhaltungsapplikation, Transmitter, Distributor
Auslöser	Eine Meldung oder eine Anfrage wurde via Distributor an einen Endempfänger gesendet. Die Antwort wird via Distributor empfangen.
Vorbedingungen	<ul style="list-style-type: none"> Distributor sendet eine Antwort
Nachbedingungen	<ul style="list-style-type: none"> Fehler, Warnungen und Informationen aus der Antwort werden aufbereitet und dem Benutzer vollständig und in verständlicher Form angezeigt. Für den Endbenutzer nicht relevante Informationen müssen dem technischen Support zur Verfügung stehen (StackTrace, Fault-Detail, etc.) Hinweise an den Endreceiver müssen diesem als Notification gesendet werden. Bei einem Fehlschlag: Distributor nicht erreichbar: Fehlermeldung
Included Use Cases	-
Fehlerliste	<p>technische Fehler:</p> <ul style="list-style-type: none"> Fehler beim Signieren der Distributor ist nicht erreichbar die von der Lohnbuchhaltung aufbereitete Meldung entspricht nicht dem Schema (Validität nicht gegeben) Fehler beim ver/ -entschlüsseln <p>Fachliche Fehler:</p> <ul style="list-style-type: none"> Gemäss (RLID, 2018)

15.1 Zusätzliche Informationen

Gibt es beim Empfang einer Story Probleme, *muss* dies im nächsten Request dem Endempfänger mitgeteilt werden. Dies geschieht unter Verwendung der Notification-Struktur: Eingeteilt in Info, Warning und Error werden jeweils Details zur jeweiligen Story übermittelt.

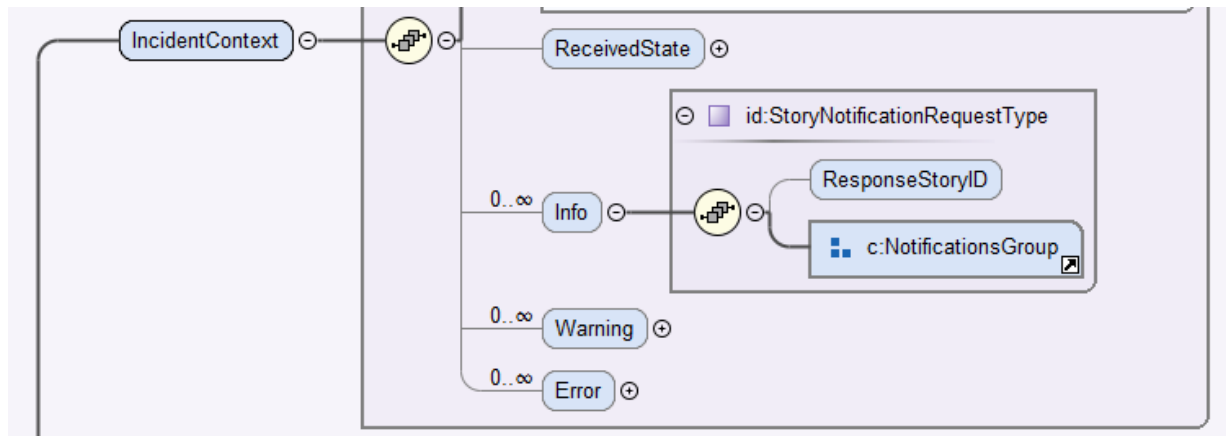


Abbildung 11: Notifications

16. Anhang

16.1 Referenzen

Die folgenden Referenzen können, zum Teil gebündelt als zip-Files, über das Internet bezogen werden. Die darin enthaltenen index.html - Files geben Zugang zu Informationen, der Übersicht und den einzelnen Dokumenten.

<http://www.swissdec.ch/richtlinien.htm> Richtlinien zum Leistungsstandard.

ACKNSwissdec, S. (2018). AcknowledgementNotification. Bern, Schweiz.

OVID, S. (2018). IncidentOverview. Bern, Schweiz.

RLID, S. (2018). Richtlinien für den Leistungsstandard-CH. Bern, Schweiz.

SECTID, S. (2018). ID_SecurityTransmitter. Bern, Schweiz.

WSDLID, S. (2018). IncidentDeclarationService. Bern, Schweiz.

XSDID. (2018). IncidentDeclarationServiceTypes.xsd. Bern, Schweiz.