

12 Reliable Messaging Technology

12.1 Überblick

In den vorherigen Kapiteln haben wir gesehen, wie man eine Nachricht vom Dienstenutzer zum Dienst übertragen kann. Später haben wir diese Übertragung auf dem Transportkanal optimiert. Alle diese Betrachtungen gingen davon aus, dass die Nachrichten sicher beim jeweiligen Empfänger ankommen. Betrachten wir jetzt aber einmal das reale Umfeld, in dem wir uns bewegen. Hier kommt man leicht zu dem Schluss, dass die Annahme der garantierten Übertragung einer Nachricht nicht sonderlich lange Bestand hat.

Unser Herr Kainer nutzt jetzt regelmäßig den ATM oder das Sparkassenportal. Das bedeutet, er kann entweder Geld abheben, eine Überweisung durchführen, Theaterkarten bestellen oder vieles mehr. Der ATM-Dienst wiederum läuft irgendwo auf dieser Welt, möglicherweise auf einem anderen Kontinent. Die Nachrichten müssen also unter Umständen eine gewaltige Entfernung über die verschiedensten Leitungswege zurücklegen. Unsichtbar für Herrn Kainer müssen wir also garantieren, dass trotz der Leitungswege, Zwischenstationen und Umgebungsbedingungen (Wetter, Stromausfälle, mobile Endgeräte, ...) die Nachrichten beim Empfänger ankommen und wir außerdem eine zuverlässige Antwort bekommen, die den Empfang bestätigt.

Die *Reliable Messaging Technology* (WS-RM) [50] stellt uns eine Spezifikation zur Verfügung, um die garantierte Übertragung von Nachrichten zu definieren. *WS-RM Policy* lässt uns wiederum per Policy unsere Wünsche in die Dienstbeschreibung aufnehmen.

Garantierte Nachrichtenübertragung am Beispiel

12.2 Die Technologie

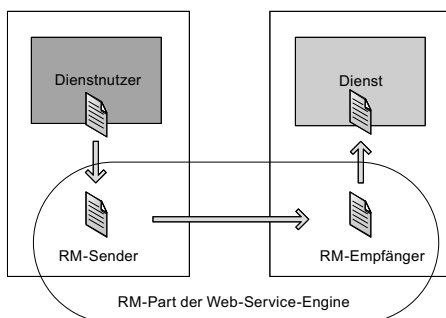
Oft ist es notwendig, dass Nachrichten garantiert beim Empfänger ankommen und dass garantiert eine Antwort zum Sender zurückkommt. Die *Reliable Messaging Technology* spezifiziert einen Mechanismus, der uns genau diese Fähigkeiten zur Verfügung stellt. Primär wird ein Protokoll definiert, um Nachrichten zu identifizieren, den Transport zu

WS-RM SOAP Binding

überwachen und die garantierte Auslieferung zu erzwingen. Das geschieht zwischen genau zwei beteiligten Parteien: dem Sender der Nachricht und dem Empfänger. Obwohl die WS-RM-Spezifikation selbst unabhängig vom verwendeten Nachrichtenformat ist, wird ein SOAP-Binding für die Out-of-the-Box-Interoperabilität als fester Bestandteil in der Spezifikation definiert.

Der Mechanismus der *Reliable Messaging Technology* ist erweiterbar, sodass zum Beispiel auch Sicherheitsaspekte Berücksichtigung finden. Die Spezifikation kann mit *WS-Security* (siehe Kapitel 13) kombiniert werden, um eine sichere, garantierte Nachrichtenübertragung zu ermöglichen. Die Abbildung 12-1 zeigt das Prinzip einer Nachrichtenübertragung mit WS-RM.

Abb. 12-1
WS-RM und Nachrichtenübertragung



WS-RM definiert einige Formen der Übertragung, die von den jeweiligen Web-Service-Engines zu unterstützen sind. Dabei wird jedoch keine Aussage über die zu verwendenden Algorithmen und Technologien gemacht. So ist es denkbar, dass Nachrichten persistent in Datenbanken zwischengespeichert werden. Diese Überlegung führt zu einem interessanten Aspekt der WS-RM-Technologie; es sollte immer beachtet werden, dass zusätzliche Mechanismen für die garantierte Auslieferung von Nachrichten zusätzliche Übertragungszeit bedeuten.

Performance beachten

Die folgenden Ausliefergarantien sind von den Endpunkten zu unterstützen (WS-RM-Spezifikation 1.1):

■ AtLeastOnce

Jede Nachricht ist garantiert einmal (oder mehrmals) auszuliefern, in allen anderen Fällen ist ein Übertragungsfehler zu erzeugen. Das bedeutet, der RM-Sender muss so lange den Sendevorgang wiederholen, bis eine Bestätigung vom RM-Empfänger eingegangen ist. Der RM-Empfänger hat die Aufgabe, die Nachricht garantiert zuzustellen. Er hat jedoch nicht die Aufgabe, doppelte Nachrichten, die während dieses Übertragungsprozesses entstehen können, herauszufiltern.

- **AtMostOnce**
Jede Nachricht ist höchstens einmal auszuliefern. Der RM-Sender kann den Sendevorgang wiederholen, bis eine Bestätigung eintrifft – er muss es aber nicht tun. Der RM-Empfänger muss doppelte Nachrichten aus der Übertragung herausfiltern.
- **ExactlyOnce**
Jede Nachricht ist garantiert einmal auszuliefern, in allen anderen Fällen ist ein Übertragungsfehler zu erzeugen. Das bedeutet, der RM-Sender muss so lange den Sendevorgang wiederholen, bis eine Bestätigung vom RM-Empfänger eingegangen ist. Der RM-Empfänger hat die Aufgabe, die Nachricht garantiert zuzustellen. Er hat zusätzlich die Aufgabe, doppelte Nachrichten herauszufiltern.
- **InOrder**
Die Nachrichten sind garantiert in der gleichen Reihenfolge auszuliefern, wie diese die Quelle verlassen haben. Das bedeutet, der RM-Empfänger muss die Nachrichten eventuell gemäß ihrer ursprünglichen Reihenfolge ordnen.
Diese Auslieferungsgarantie kann in Kombination mit *AtLeastOnce*, *AtMostOnce* und *ExactlyOnce* genutzt werden.

12.3 Policies für das Reliable Messaging

12.3.1 Standard-Policies

Die WS-RM-Policies ermöglichen es uns, alle Informationen zum garantierten Nachrichtenaustausch in der WSDL anzulegen [49]. Die genauen Angaben, welche Policy-Elemente erlaubt sind und wie sie angeordnet werden müssen, sind in der WS-RM-Policy-Spezifikation [49] zu finden. Bitte beachten Sie jedoch, dass es aufgrund der Komplexität des Themas (lokaler Speicher für Nachrichten, Timeout-Handling usw.) vorkommen kann, dass nicht alle Web-Service-Engines alle Policies unterstützen.

Der folgende Quelltext zeigt exemplarisch das Einschalten der garantierten, einmaligen Auslieferung einer Nachricht, wobei die Reihenfolge der einzeln gesendeten Nachrichten zwischen Sender und Empfänger erhalten bleibt.

Listing 12.1

*Garantierte
Auslieferung der
Nachrichten*

```

<wsdl:definitions
  ...
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsp=
    "http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsrmp=
    "http://docs.oasis-open.org/ws-rx/wsrmp/200702"
  xmlns:wssu=
    "http://docs.oasis-open.org/wss/2004/01/
      oasis-200401-wss-wssecurity-utility-1.0.xsd">
  ...
  <wsp:Policy
    wsu:Id="ATMPortBindingPolicy" >
    <wsrmp:RMAssertion>
      <wsp:Policy>
        <wsrmp:DeliveryAssurance>
          <wsp:Policy>
            <wsrmp:ExactlyOnce/>
            <wsrmp:InOrder/>
          </wsp:Policy>
        </wsrmp:DeliveryAssurance>
      </wsp:Policy>
    </wsrmp:RMAssertion>
  </wsp:Policy>
  ...
  <wsdl:binding
    name="ATMServiceSOAP"
    type="tns:ATMServicePort" >
    <wsp:PolicyReference URI="#ATMPortBindingPolicy" />
    ...
  </wsdl:binding>
  ...
</wsdl:definitions>

```

12.3.2 Erweiterungen der Hersteller

Wie schon im letzten Absatz angekündigt, erfordert die Reliable-Messaging-Technologie eine ganze Menge mehr Konfiguration vom Hersteller der Web-Service-Engine. Deshalb kann es vorkommen, dass wir neben den Standard-Policies weitere, herstellerabhängige Informationen vorfinden. Diese werden nicht von allen Web-Service-Engines ausgewertet und sind deshalb auch nicht so problematisch. Zumindest in der Theorie. Wenn es jedoch in diesem Zusammenhang zu Inkompatibilitäten kommt, sollten Sie sich an die Dokumentation halten.

Im folgenden Fragment einer Dienstbeschreibung ist die Erweiterung `RmFlowControl` unter einem herstellerabhängigen Namespace zu finden. Diese Policy-Erweiterung erkennt zum Beispiel der Sun Metro Web Service Stack (ab Version 1.3) und das Microsoft .NET Framework (ab Version 3.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions ...
  xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:net=
    "http://schemas.microsoft.com/net/2005/02/rm/policy">
  ...
  <wsp:Policy ...>
    <wsp:ExactlyOne>
      <wsp>All>
        ...
        <net:RmFlowControl>
          <net:MaxReceiveBufferSize>
            64
          </net:MaxReceiveBufferSize>
        </net:RmFlowControl>
      </wsp>All>
    </wsp:ExactlyOne>
  </wsp:Policy>
  ...
</definitions>
```

Listing 12.2*Policy der Hersteller*

12.4 Garantierte Nachrichtenübertragung und der ATM-Dienst

Im folgenden Beispiel wollen wir eine Operation für unseren Dienst entwickeln, die es ermöglicht, eine fremde Währung zu beziehen. Nehmen wir wieder einmal unser Beispiel, in dem unser Herr Kainer in den Urlaub fahren möchte und noch dazu noch eine Handvoll US-Dollar benötigt. In diesem Zusammenhang muss natürlich das Konto mit entsprechend diesem Betrag belastet werden. Da die Operation kritisch ist, muss sie in jedem Fall erfolgen, sodass die garantierte Auslieferung der Nachrichten zum Tragen kommt. Als Erstes definieren wir die Nachrichten, die zwischen Dienst und Dienstenutzer auszutauschen sind. Da wäre erst einmal die Anfrage vom Dienstenutzer zum Dienst.

Beispiel »rm/ATM«

Listing 12.3*Anfrage an den Dienst*

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:fi-atm="http://www.fi.de/atm/types"
  targetNamespace="http://www.fi.de/atm/types">
  <xsd:element name="ATMForeignCurrencyOrder"
    type="fi-atm:ATMForeignCurrencyOrderType" />
  <xsd:complexType name="ATMForeignCurrencyOrderType">
    <xsd:sequence>
      <xsd:element name="securetoken"
        type="xsd:token" />
      <xsd:element name="Product"
        type="fi-atm:ProductDataType"
        minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ProductDataType">
    <xsd:sequence>
      <xsd:element name="currency"
        type="xsd:string" />
      <xsd:element name="value"
        type="xsd:float" />
      <xsd:element name="amount"
        type="xsd:integer" />
      <xsd:element name="currencyvaluedate"
        type="xsd:date" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

Als Antwort erhalten wir die Information, dass alles in Ordnung gegangen ist, und als Referenz für etwaige Nachverfolgung die Transaktionsnummer.

Listing 12.4*Response*

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:fi-atm="http://www.fi.de/atm/types"
  targetNamespace="http://www.fi.de/atm/types">
  <xsd:element name="ATMOrderConfirmation"
    type="fi-atm:ATMOrderConfirmationType" />
  <xsd:complexType name="ATMOrderConfirmationType">
    <xsd:sequence>
      <xsd:element name="confirmations"
        type="fi-atm:ATMOrderConfirmation"
        minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>

```

```

<xsd:complexType name="ATMOrderConfirmation">
  <xsd:sequence>
    <xsd:element name="transactionid"
      type="xsd:token" />
    <xsd:element name="confirmed"
      type="xsd:boolean" />
    <xsd:element name="charged"
      type="xsd:boolean" />
    . . .
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

Die Dienstbeschreibung des ATM-Dienstes kann jetzt um die Buchungsoption ergänzt werden. Gleichzeitig definieren wir die Policy für eine garantierte, einmalige Nachrichtenübertragung.

```

<definitions
  xmlns:wsp=
    "http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsu=
    "http://docs.oasis-open.org/wss/2004/01/
      oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsam=
    "http://www.w3.org/2007/05/addressing/metadata"
  xmlns:wsm=
    "http://docs.oasis-open.org/ws-rx/wsrmp/200702"
  ...>
  <wsp:UsingPolicy />
  <wsp:Policy wsu:Id="ATMPortBindingPolicy">
    <wsp:ExactlyOne>
      <wsp:All>
        <wsam:Addressing wsp:Optional="false"/>
        <wsm:RMAssertion>
          <wsp:Policy></wsp:Policy>
        </wsm:RMAssertion>
      </wsp:All>
    </wsp:ExactlyOne>
  </wsp:Policy>
  ...
</definitions>

```

Listing 12.5

Policy für garantierte Übertragung

Die Implementierung des Dienstes sowie des Test-Dienstnutzers ist weniger spektakulär und in vollständiger Form in den Beispielen zum Buch auf der Webseite zu finden. Im folgenden Listing sind die wichtigsten Zeilen für die Kommunikation zwischen Dienst und Dienstnutzer skizziert. In unserer Benutzeroberfläche für die Mustersparkasse ist das Beziehen von Devisen, wie in Abbildung 12-2 zu sehen ist, leicht möglich.

Listing 12.6

Dienstnutzer im
Quelltext

```

. . .
// Neuen ATMServicePort generieren.
ATMServicePort port = service.getATMServiceSOAP();

// Neue ATMOrder anlegen.
ATMForeignCurrencyOrderType order =
    new ATMForeignCurrencyOrderType();
order.setProduct( foreignCurrency );
order.setSecuretoken( token );

// Bestellung auslösen ...
ATMOrderConfirmationType atmConfirmation =
    port.bookForeignCurrency( order );

// ... und Antwort abholen.
ATMOrderConfirmation atm =
    atmConfirmation.getConfirmations().get( 0 );

// Beispielhaft die Ticket-Nummer zurückgeben lassen.
String transactionNo = atm.getTransactionid();
. . .

```

Abb. 12-2

Dienstnutzer

The screenshot shows a web form titled 'Mustersparkasse'. It contains the following fields and values:

Währung:	Amerikanischer Dollar
Stück:	1
Wert:	200
Datum:	23.12.2009

Below the fields is a button labeled 'Bestellen'.

Nach der Ausführung der Operation mithilfe des Test-Dienstnutzers können wir uns jetzt die Nachrichten anschauen. Zuerst betrachten wir die gesendete Nachricht, vom Dienstnutzer zum Dienst:

Listing 12.7

Request-Nachricht

```

<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope">
  <S:Header>
    ...

```

```

<ns2:Sequence
  xmlns="http://www.w3.org/2005/08/addressing"
  xmlns:ns2="
    "http://docs.oasis-open.org/ws-rx/wsrn/200702"
  <ns2:Identifizier>
    uuid:904f22a8-f821-4dbc-92b3-aa8cbc4d78d1
  </ns2:Identifizier>
  <ns2:MessageNumber>1</ns2:MessageNumber>
</ns2:Sequence>
</S:Header>
<S:Body>
  ...
</S:Body>
</S:Envelope>

```

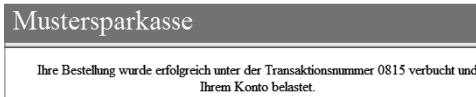


Abb. 12-3
Dienstenutzer, Antwort vom Dienst

Deutlich ist die ID der Nachricht zu sehen: ns2:Identifizier. RM-Quelle und RM-Ziel (siehe Abbildung 12-1) nutzen die ID, um den erfolgreichen Austausch der Nachrichten festzustellen. Das Ergebnis enthält dann, wie erwartet, die Nachrichten-ID:

```

<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope">
  <S:Header>
    ...
  <ns2:Sequence
    xmlns="http://www.w3.org/2005/08/addressing"
    xmlns:ns2="http://docs.oasis-open.org/ws-rx/wsrn/200702" >
    <ns2:Identifizier>uuid:21c099e7-8761-4037-a969-8464a7aacf64</ns2:Identifizier>
    <ns2:MessageNumber>1</ns2:MessageNumber>
  </ns2:Sequence>
  <ns2:AckRequested
    xmlns="http://www.w3.org/2005/08/addressing"
    xmlns:ns2="http://docs.oasis-open.org/ws-rx/wsrn/200702" >
    <ns2:Identifizier>uuid:21c099e7-8761-4037-a969-8464a7aacf64</ns2:Identifizier>
  </ns2:AckRequested>
  <ns2:SequenceAcknowledgement
    xmlns="http://www.w3.org/2005/08/addressing"
    xmlns:ns2="http://docs.oasis-open.org/ws-rx/wsrn/200702" >
    <ns2:Identifizier>uuid:904f22a8-f821-4dbc-92b3-aa8cbc4d78d1</ns2:Identifizier>
    <ns2:AcknowledgementRange Lower="1" Upper="1"/>

```

Listing 12.8
Response-Nachricht

```
</ns2:SequenceAcknowledgement>  
</S:Header>  
<S:Body>  
  ...  
</S:Body>  
</S:Envelope>
```

12.5 Zusammenfassung

In diesem Kapitel haben wir Folgendes gelernt:

- Die garantierte Auslieferung der Nachrichten ist durch die WS-ReliableMessaging-Spezifikation beschrieben.
- Verschiedene Möglichkeiten stehen für das wiederholte Senden von Nachrichten zur Verfügung.
- Die garantierte Reihenfolge der Nachrichten zwischen Sender und Empfänger kann erzwungen werden.
- Verschiedene Hersteller bringen Erweiterungen für die Reliable-Messaging-Policy mit. Diese werden nicht von allen Engines verstanden und sind demzufolge auf diesen Engines zu ignorieren.

12.6 Übungsaufgaben

1. Beschreiben Sie in wenigen Worten, was mit der Reliable-Messaging-Technologie erreicht werden soll.
2. Welche Ausliefergarantien sind in der Spezifikation definiert? Können diese kombiniert werden?
3. Was passiert mit Policies, die nicht durch den Standard abgedeckt sind?