



Solutions

Insurance Service 1.0 Pilot

Integration Guide

This document distills important information needed by new members of the Rüşchlikon community when they first connect to SWIFTNet. It describes the development effort required to add SWIFT capability to an existing ACORD Messaging Service (AMS) Gateway.

16 June 2009

Table of Contents

Table of Figures	3
Table of XML Examples	4
Preface	5
1 Background	6
2 Phase 1a Overview	9
3 SWIFT SOAP Header Formats	11
3.1 SOAP Headers Sent from AMS Gateway to SWIFT Gateway	11
3.2 SOAP Headers Sent from SWIFT Gateway to AMS Gateway	14
4 SWIFT Gateway SOAP Headers for Routing and Addressing	17
4.1 Namespace Declaration.....	17
4.2 Identifying the Local SWIFT Gateway Configuration	17
4.3 Addressing a Remote Counterparty on SWIFTNet.....	17
5 SWIFT Gateway SOAP Headers for SWIFTNet Security	19
5.1 Message Authorisation	20
5.2 Message Signing.....	20
6 WS-Security SOAP Header for Local Authentication Protocol (LAU)	21
6.1 Adding LAU to Client Requests and Server Responses.....	21
6.2 LAU Check on Server Requests and Client Responses.....	24
7 Exception Handling	26
8 Other SOAP Header and Message Requirements	27
8.1 Non-SWIFT SOAP Headers	27
8.2 SOAP Version	27
8.3 XML Encoding.....	27
8.4 MIME Message Structure	27
9 Other General Considerations	28
9.1 Size Limits.....	28
9.2 Response Times	28
9.3 Simultaneous Connections	28
10 Phase 1a Test Requirements	29
10.1 Initial Message Syntax Tests	29
10.2 Detailed Connectivity, Protocol, and Interoperability Tests	29
11 Sample Messages	30
12 Support	46
13 References	48
Legal Notices	49

Table of Figures

Figure 1: Current ACORD systems use a point-to-point connectivity architecture over the Internet.....	6
Figure 2: The final Rüsclikon architecture leverages SWIFTNet as the common communication, security, and service delivery platform	7
Figure 3: Final Solution Architecture (Network View): Shows future work beyond scope of Phase 1a.	8
Figure 4: Success in Phase 1a requires installation, configuration, and development effort.....	9
Figure 5: Phase 1a Target Architecture. Single Gateway access on SWIFTNet replaces point-to-point connectivity over the Internet. SWIFT security used consistently to replace multi-bilateral SSL.....	10
Figure 6: The Web Services Host Adapter on the SWIFT Gateway consolidates multiple SOAP messages into a single, synchronous request-reply interaction	11
Figure 7: SWIFTNet multi-hop security architecture	19
Figure 8: Configuring the bilateral key in the SWIFT Gateway	22
Figure 9: WSHA Events must be enabled when debugging.....	46

Table of XML Examples

XML Example 1: Sample Client Request Message from Client AMS to Client SWIFT Gateway	12
XML Example 2: Sample Server Response Message from Server AMS to Server SWIFT Gateway	14
XML Example 3: Sample Server Request Message from Server SWIFT Gateway to Server AMS	15
XML Example 4: Sample Client Response Message from Client SWIFT Gateway to Client AMS. Note that the SWIFT SOAP Header has the same format as the Server Request.	16
XML Example 5: The application identifier contains the message partner name	17
XML Example 6: The WSHA Request Header contains information to identify and locate the sender and receiver on SWIFTNet. It also describes the contents of the message.	18
XML Example 7: AuthorisationContext is used to identify a valid user certificate to SWIFT Gateway	20
XML Example 8: WSHA Security is used to identify a valid signer certificate to SWIFT Gateway	20
XML Example 9: Sample WS Security Header added by an AMS Gateway to carry WSHA-specific Local Authentication information to the local SWIFT Gateway.	23
XML Example 10: Sample WS Security Header added by the SWIFT Gateway to carry WSHA-specific Local Authentication information to the target AMS Gateway.	25
XML Example 11: Sample Client Request Message containing Technical Account PostRs sent from Client (AON) AMS to Client (AON) SWIFT Gateway	33
XML Example 12: Sample Server Request Message sent from Server (Swiss Re) SWIFT Gateway to Server (Swiss Re) AMS Gateway. Contains original Technical Account PostRs from XML Example 11. Note that this shows an example of the MIME Headers added by the SWIFT Gateway.....	36
XML Example 13: Sample Server Response Message returned by Server (Swiss Re) AMS Gateway to Server (Swiss Re) SWIFT Gateway	37
XML Example 14: Sample Client Response Message returned by Client (AON) SWIFT Gateway to Client (AON) AMS Gateway. The body contains the PostRs from XML Example 13	38
XML Example 15: Sample of a Client SOAP Fault Response generated by the local SWIFT Gateway. This example indicates a problem with the format of the SOAP Header. Note that the SAGHeader is incomplete because the information was not included in the original message.....	40
XML Example 16: Sample of a Client SOAP Fault Response generated by SWIFTNet. This example indicates a problem with the certificate that was used to sign the message.....	42
XML Example 17: Sample of a Server SOAP Fault Response generated by the Server (Swiss Re) AMS Gateway.....	43
XML Example 18: Sample of a Client Fault Message returned by Client (AON) SWIFT Gateway to Client (AON) AMS Gateway. Body contains the SOAP Fault Response sent by the Server (Swiss Re) AMS Gateway (XML Example 17). Note that this shows an example of the HTTP Header added by the SWIFT Gateway.....	45

Preface

Purpose of this document

This document distils important information needed by new members of the Rüşchlikon community when they first connect to SWIFTNet. It describes the development effort required to add SWIFT capability to an existing ACORD Messaging Service (AMS) Gateway.

While this guide does not claim to be a comprehensive A-to-Z of ACORD messaging over SWIFTNet, great efforts have been made to understand and manage the effort required to succeed with your integration project.

Intended audience

This document is for new members of the Rüşchlikon community that want to connect to SWIFTNet.

Significant changes

This is the second full public draft, published in June 2009. It contains significant updates from the first public draft that was published in February 2009. The updates capture all the lessons learned in the pilot to date. This version also includes new and more comprehensive message samples. A final version of this document is scheduled for publication after the pilot phase is completed.

1 Background

The initial implementation phase (Phase 1) of the Rüşchlikon Initiative to bring ACORD messaging over SWIFTNet is implemented as three sequential stages. In this document, these are referred to Phase 1a, Phase 1b, and Phase 1c¹. All stages of Phase 1 have some impact on AMS providers, but 1a requires the most significant effort. In this phase, we establish connectivity with SWIFTNet and implement SWIFTNet protocol and security support.

To provide some context to the document, we outline the situation today, and then describe the situation at the end of Phase 1c. The rest of this document focuses only on the software development aspects of Phase 1a for AMS providers. Additional documents are being prepared to describe installation and configuration related to Phase 1a, and all aspects of Phases 1b and 1c.

Current ACORD systems use a point-to-point connectivity architecture over the Internet. Connectivity and Security is negotiated and implemented on a per-counterparty basis, as shown in Figure 1.

Figure 1: Current ACORD systems use a point-to-point connectivity architecture over the Internet

¹ It should be noted that this first pilot implementation phase actually corresponds to Phase 2-3 as discussed in the [PDD-IWG].

The Rüschtikon Initiative improves this by leveraging the SWIFT network infrastructure to deliver a scalable one-to-many hub topology over SWIFT's secure global network SWIFTNet. (See Figure 2)

SWIFTNet includes common security models, addressing schemes, and provides a platform for delivery of valuable shared central services. SWIFT also brings decades of experience as a disciplined standards body to the ACORD solution, to support greater process harmonisation.

Figure 2: The final Rüschtikon architecture leverages SWIFTNet as the common communication, security, and service delivery platform

The final architecture shown in Figure 2 is achieved through three phases. Phases 1b and 1c are introduced in Figure 3, but are otherwise beyond the scope of this document:

- Phase 1b introduces a central proxy server infrastructure to simplify routing and configuration.
- Phase 1c delivers a central infrastructure for value-added service delivery, and the first such service, Message Validation.

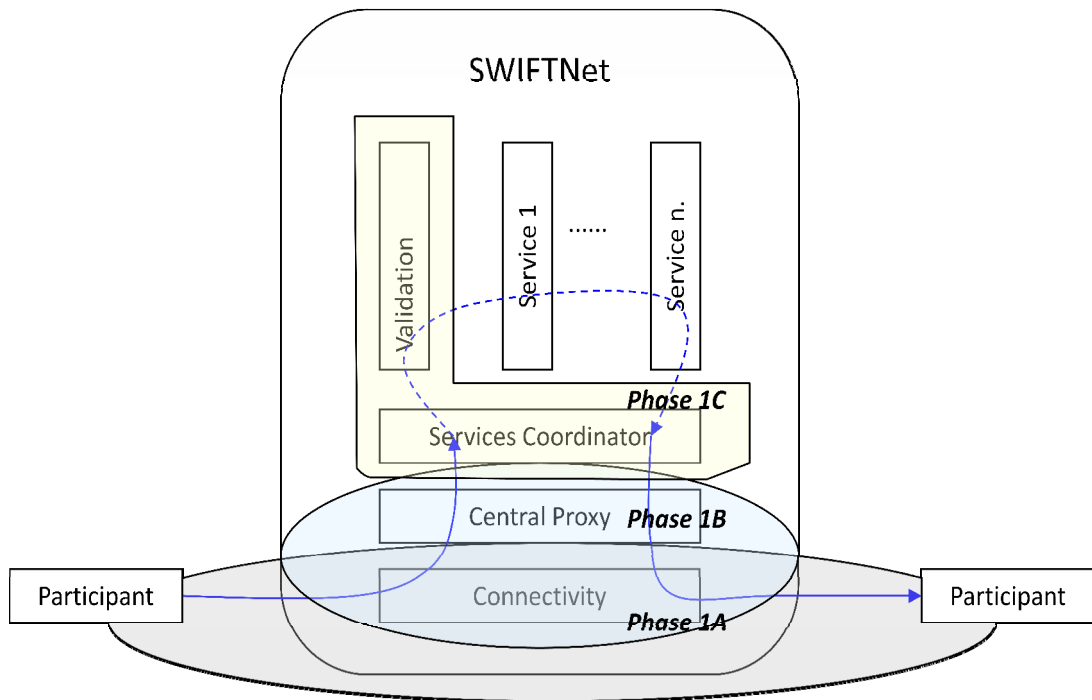


Figure 3: Final Solution Architecture (Network View): Shows future work beyond scope of Phase 1a.

2 Phase 1a Overview

As shown in Figure 4, Phase 1a of the Rüşchlikon migration requires effort in two areas:

- 1) A SWIFT Gateway must be installed and configured.
- 2) AMS providers must implement the SWIFT protocols in the AMS gateway.

Figure 4: Success in Phase 1a requires installation, configuration, and development effort

This document covers only area 2 (the development effort for AMS implementers) including:

- **SWIFT SOAP Headers:** Used to address other participants and to secure messages on SWIFTNet.
- **SWIFT Local Authentication:** Used to secure traffic between AMS and the local SWIFT Gateway.
- **SWIFT Exception Handling:** Introducing the format of the SOAP 1.1 Fault elements.

After successful completion of Phase 1a, all participants can participate in ACORD transactions over SWIFTNet according to the architecture shown in Figure 5. Subsequent phases deliver further efficiencies as were seen in Figure 2 and Figure 3.

Figure 5: Phase 1a Target Architecture. Single Gateway access on SWIFTNet replaces point-to-point connectivity over the Internet. SWIFT security used consistently to replace multi-bilateral SSL.

3 SWIFT SOAP Header Formats

As shown in Figure 6, the SWIFT Gateway hosts a Web service intermediary node called the Web Services Host Adapter (WSHA). WSHA intermediates all Web service requests sent on SWIFTNet, so any SOAP request or response (including Faults) addressed to a SWIFTNet counterparty must be directed through the local WSHA endpoint.

Additionally, all requests and responses (including Faults) sent to WSHA must contain additional SWIFT-defined (but Web services-compliant) SOAP Headers. This section summarises these as relevant to Phase 1a of the Rüşchlikon project.



Figure 6: The Web Services Host Adapter on the SWIFT Gateway consolidates multiple SOAP messages into a single, synchronous request-reply interaction

3.1 SOAP Headers Sent from AMS Gateway to SWIFT Gateway

When an AMS Gateway sends anything to the SWIFT Gateway, it must add additional SOAP-compliant headers addressed to the SOAP Actor called "urn:swift:sag".

An AMS Gateway can send two types of messages to a SWIFT Gateway, Client Requests, and Server Responses. Server Responses include SOAP Faults. Each message type must include a small SOAP header to carry important SWIFT-specific security and routing information.

Client Requests are shown in red in Figure 6, and their structure is introduced in section 3.1.1.1. Server Responses are shown in blue and their structure is introduced in section 3.1.2. Section 4 details the contents of these headers.

3.1.1 SOAP Headers Added to a Client Request

A Client Request is the original outgoing request message sent from the Client AMS Gateway to the Local SWIFT Gateway².

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wssse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
```

² In a Rüşchlikon context, Client Requests may contain for example, a PostRq or PingRq.

```

200401-wss-wssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>

    <!-- SWIFT Gateway Header - See "SWIFT Gateway SOAP Headers for
Routing and Addressing" -->
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
      wsu:Id="SAGHeader"
      SOAP-ENV:actor="urn:swift:sag"
      SOAP-ENV:mustUnderstand="1">
      <wsha:ApplicationId>...</wsha:ApplicationId>
      <wsha:RequestHeader>
        <wsha:Requestor>...</wsha:Requestor>
        <wsha:Responder>...</wsha:Responder>
        <wsha:Service>...</wsha:Service>
        <wsha:RequestType>...</wsha:RequestType>
      </wsha:RequestHeader>
      <wsha:AuthorisationContext>
        <wsha:UserDN>...</wsha:UserDN>
      </wsha:AuthorisationContext>
      <wsha:Security>
        <wsha:SignDN>...</wsha:SignDN>
      </wsha:Security>
    </wsha:SAGHeader>

    <!-- LAU Header
- See "WS-Security SOAP Header for Local Authentication Protocol
(LAU)" -->
    <wsse:Security wsu:Id="LAUHeader"
      SOAP-ENV:actor="urn:swift:sag"
      SOAP-ENV:mustUnderstand="1">
      ... LAU Signature details here ...
    </wsse:Security>

  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SOAPBody">...</SOAP-
ENV:Body>
  <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SOAPBody">
    <ac:PingRq xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Ping"
      xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1"
      xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1
AcordMsgSvc_v-1-5-0.xsd" Version="1.5.0">
      ...
    </ac:PingRq>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XML Example 1: Sample Client Request Message from Client AMS to Client SWIFT Gateway

3.1.2 SOAP Headers Added to a Server Response

A Server Response is any synchronous response or SOAP Fault sent by the server AMS Gateway in reply to a request from the local SWIFT Gateway³.

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wss="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>
    <!-- SWIFT Gateway Header - See "SWIFT Gateway SOAP Headers for
Routing and Addressing" -->
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
      wsu:Id="SAGHeader"
      SOAP-ENV:actor="urn:swift:sag"
      SOAP-ENV:mustUnderstand="1">
      <wsha:ApplicationId>...</wsha:ApplicationId>
      <wsha:AuthorisationContext>
        <wsha:UserDN>...</wsha:UserDN>
      </wsha:AuthorisationContext>
      <wsha:Security>
        <wsha:SignDN>...</wsha:SignDN>
      </wsha:Security>
    </wsha:SAGHeader>

    <!-- LAU Header
- See "WS-Security SOAP Header for Local Authentication Protocol
(LAU)" -->
    <wsse:Security wsu:Id="LAUHeader"
      SOAP-ENV:actor="urn:swift:sag"
      SOAP-ENV:mustUnderstand="1">
      ... LAU Signature details here ...
    </wsse:Security>

  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SOAPBody">
    <ac:PingRs xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Ping"
      xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1"
      xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1
AcordMsgSvc_v-1-5-0.xsd" Version="1.5.0">
      ...
    </ac:PingRs>

```

³ In a Rüsçhlikon context, Server Responses may contain for example, a PostRs, PingRs, or SOAP Fault.

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML Example 2: Sample Server Response Message from Server AMS to Server SWIFT Gateway

3.2 SOAP Headers Sent from SWIFT Gateway to AMS Gateway

When the SWIFT Gateway sends anything to an AMS Gateway, it adds additional SOAP-compliant headers addressed to the SOAP actors called "urn:swift:sag:ws.handler.sag", and "urn:swift:sag:ws.handler.lau". These headers contain information that allows the AMS Gateway to validate the sender of the message and ensure that it has not been locally tampered. The AMS Gateway must act appropriately on all instructions addressed to these SOAP Actors.

Two kinds of messages are sent from a SWIFT Gateway to an AMS Gateway, Server Requests, and Client Responses. Client Responses include SOAP Faults. The additional SWIFT-specific SOAP headers have identical structure in each case⁴, but they are shown separately for completeness.

Server Requests are shown in green in Figure 6, and their structure is introduced in section 3.2.1. Client Responses are shown in orange and their structure is introduced in section 3.2.2. Section 4 details the contents of these headers.

3.2.1 SOAP Headers Received with a Server Request

A Server Request is the message sent from the Server SWIFT Gateway to the Server AMS Gateway. It is the counterpart of the Client Request sent by the client AMS Gateway, but most of the SWIFT network-specific header information has been removed⁵.

The SOAP Body contains the original Client Request SOAP Body. Any additional non-SWIFT SOAP headers added on the client side are also present in their original form.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wss="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>
    <!-- SWIFT Gateway Header - See "SWIFT Gateway SOAP Headers for
Routing and Addressing" -->
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
```

⁴ Although the contents will obviously be different.

⁵ This information was only required to secure and route the message across SWIFTNet and has no meaning to the ultimate recipient.

```

wsu:Id="SAGHeader"
SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
SOAP-ENV:mustUnderstand="1" >
<wsha:ApplicationId>...</wsha:ApplicationId>
<wsha:Security>
  <wsha:SignDN>...</wsha:SignDN>
</wsha:Security>
</wsha:SAGHeader>

<!-- LAU Header
- See "WS-Security SOAP Header for Local Authentication Protocol
(LAU)" -->
<wsse:Security wsu:Id="WSHA124...99859270"
SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
SOAP-ENV:mustUnderstand="1">
... LAU Signature details here ...
</wsse:Security>

</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SOAPBody">
  <ac:PingRq xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Ping"
xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1"
xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1
AcordMsgSvc_v-1-5-0.xsd" Version="1.5.0">
    ...
  </ac:PingRq>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XML Example 3: Sample Server Request Message from Server SWIFT Gateway to Server AMS

3.2.2 SOAP Headers Received with a Client Response

A Client Response is the message sent from the Client SWIFT Gateway to the Client AMS Gateway in response to the original Client Request.

It is the counterpart of the Server Response and the SOAP Body contains the original SOAP Body sent in the Server Response.

Any additional non-SWIFT SOAP headers added on the server side are also present in their original form.

A Client Response may also contain a SOAP Fault generated by the local client SWIFT Gateway⁶ or by SWIFTNet.

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"

```

⁶ NOTE: If the Fault is raised because the SOAP message was not parseable, or the expected SAGHeader was missing, then the Fault does not contain any SWIFT-specific SOAP Headers.

```

        xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
    <SOAP-ENV:Header>
        <!-- SWIFT Gateway Header - See "SWIFT Gateway SOAP Headers for
Routing and Addressing" -->
        <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
wsu:Id="SAGHeader"
SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
SOAP-ENV:mustUnderstand="1" >
            <wsha:ApplicationId>...</wsha:ApplicationId>
            <wsha:Security>
                <wsha:SignDN>...</wsha:SignDN>
            </wsha:Security>
        </wsha:SAGHeader>
        <!-- LAU Header
- See "WS-Security SOAP Header for Local Authentication Protocol
(LAU)" -->
        <wsse:Security wsu:Id="WSHA124...99859270"
SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
SOAP-ENV:mustUnderstand="1">
            ... LAU Signature details here ...
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SOAPBody">
        <ac:PingRs xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Ping"
xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1"
xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1
AcordMsgSvc_v-1-5-0.xsd" Version="1.5.0">
            ...
        </ac:PingRs>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

*XML Example 4: Sample Client Response Message from Client SWIFT Gateway to Client AMS.
Note that the SWIFT SOAP Header has the same format as the Server Request.*

4 SWIFT Gateway SOAP Headers for Routing and Addressing

This section provides details of all SWIFT-specific SOAP header elements that are relevant to the Rüsclikon initiative, and provides examples of AMS-specific usages.

All elements are described, but as seen in section 3, they may appear in different combinations in the different types of messages.

4.1 Namespace Declaration

All SWIFT-specific SOAP-header elements are defined in the following namespace:

```
xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
```

4.2 Identifying the Local SWIFT Gateway Configuration

The first new element of the `SAGHeader` sent to WSHA is the name of a pre-configured *Message Partner* that represents this AMS Application. A Message Partner is a SWIFT Interface concept that identifies the SWIFT Gateway configuration specific to a particular customer and application.

```
<wsha:ApplicationId>MP_AMS</wsha:ApplicationId>
```

XML Example 5: The application identifier contains the message partner name

This element is simply a string that matches the name of a valid WSHA message partner configured in the SWIFT Gateway. It must be configurable at the AMS Gateway, and should be included in debug tracing on received messages because it allows the recipient to uniquely identify the SWIFT Gateway configuration.

If the AMS is used to channel traffic on behalf of more than one ACORD Sender, then it must be able to address multiple message partners⁷.

When preparing a Server Response, the `ApplicationId` should be copied from the Server Request and echoed back to the local SWIFT Gateway.

4.3 Addressing a Remote Counterparty on SWIFTNet

```
<wsha:RequestHeader>
  <wsha:Requestor>cn=ams,o=aonbgb21,o=swift</wsha:Requestor>
```

⁷ The LAU key described in section 6 is defined at the message partner level in the SWIFT Gateway. One message partner per sender ensures both access control and accountability to each.

```

<wsha:Responder>cn=ams,o=swrechcz,o=swift</wsha:Responder>
<wsha:Service>swift.reinsurance!p</wsha:Service>
<wsha:RequestType>any</wsha:RequestType>
</wsha:RequestHeader>

```

XML Example 6: The WSHA Request Header contains information to identify and locate the sender and receiver on SWIFTNet. It also describes the contents of the message.

The next new element is the `RequestHeader`. This is added to a Client Request and contains information to describe the sender (Requestor), receiver (Responder), the SWIFTNet Service, and details of the message contents, as described in the following three sections.

4.3.1 Requestor and Responder

SWIFT uses Distinguished Names (DN) to identify all entities on SWIFTNet. These are based on the X.500 standard and are formatted as shown in Table 1.

Definition	Syntax in Backus-Naur Form
string matches (alphanum space any-of "_,=%")+ length ∈ { 1 .. 100 }	Sw-DN ::= S Node S (',' S Node S)* S ::= space* Node ::= Name S '=' S Name Name ::= Id '%' digit Id ::= letter ('_'? alphanum)*

Table 1: Describing a SWIFTNet Distinguished Name

An AMS must be able to map from ACORD message senders and receivers to SWIFTNet Requestors and Responders. Requestors and Responders must therefore be configurable, and linked to ACORD PartyIds through a mapping table.

Note An AMS system may represent multiple Requestors if deployed in a Service Bureau.

4.3.2 Service Name

Communication on SWIFTNet is achieved through SWIFTNet Services. A service name is a string of maximum 30 characters, including period [.] and exclamation point [!].

At the time of writing, the Rüşchlikon community has two services defined as follows:

- Pilot service: `swift.reinsurance!p`
- Live service: `swift.reinsurance`

Further service names may be added in the future, so they must be configurable at the AMS. An AMS must be able to communicate to multiple services, including at least a pilot and a live service.

4.3.3 Request Type

SWIFTNet can use this to act on the contents of a message, to perform content-based routing, etc. In general, a fixed subset of request types is typically agreed per service and the request header of all SOAP messages must contain a valid request type for the service.

At the time of writing, however, the Rüşchlikon community does not need request-specific features, so the service type is defaulted to the “wildcard” value “any”

The request types used may change in future, so must be configurable at the AMS. The AMS should be prepared to accept any `RequestType` containing a maximum of 30 lowercase alphanumeric characters (a to z and 0 to 9). This includes the periods [.] used to separate the segments (for example, “`acord.account.2005.002`”).

5 SWIFT Gateway SOAP Headers for SWIFTNet Security

SWIFTNet uses a multi-hop mechanism to provide end-to-end security for your messages.

Figure 7: SWIFTNet multi-hop security architecture.

This document discusses three components of SWIFT's end-to-end security mechanism: *Authorisation, Message Signing, and Local Authentication*⁸.

For the hops between the SWIFT Gateways, authorisation and message signing with SWIFTNet PKI is used to guarantee integrity and identity. That is, SWIFT absolutely guarantees that:

- Any message that arrives across SWIFTNet at a SWIFT Gateway is exactly as the sender SWIFT Gateway sent it.
- The sender SWIFT Gateway was allowed to send that message to the receiver, according to the rules provisioned on SWIFTNet.

The remainder of this section details how to achieve *Authorisation* and *Message Signing* via the SWIFT Certification Authority using SWIFT-specific SOAP Headers.

Local Authentication is achieved via an additional WS-Security compliant header as described in section 6.

⁸ These are in addition to security provided by dedicated SWIFT VPN hardware and SSL on all local connections between the AMS and SWIFT Gateways.

5.1 Message Authorisation

```
<wsha:AuthorisationContext>
  <wsha:UserDN>cn=authcert,o=aonbgb21,o=swift</wsha:UserDN>
</wsha:AuthorisationContext>
```

XML Example 7: AuthorisationContext is used to identify a valid user certificate to SWIFT Gateway

5.1.1 User DN

The `UserDN` is a SWIFT-compliant distinguished name (see Table 1) that identifies a valid certificate to the SWIFT Gateway. The SWIFT Gateway checks that this certificate is authorised to send ACORD messages to the Rüşchlikon service and *Closed User Group*⁹ according to rules defined centrally on SWIFTNet.

`UserDN` is added by the AMS Gateway to Client Requests and Server Responses to identify a valid certificate stored on the sender's SWIFT Gateway¹⁰. It comes from AMS Gateway configuration.

5.2 Message Signing

```
<wsha:Security>
  <wsha:SignDN>cn=signcert,o=aonbgb21,o=swift</wsha:SignDN>
</wsha:Security>
```

XML Example 8: WSHA Security is used to identify a valid signer certificate to SWIFT Gateway

5.2.1 Sign DN

The `SignDN` is a SWIFT-compliant distinguished name (see Table 1) that identifies a valid signer certificate to the SWIFT Gateway. The SWIFT Gateway uses this certificate to sign the message before sending it to the receiver.

`SignDN` is added by the AMS Gateway to Client Requests and Server Responses to identify a valid certificate stored on the sender's SWIFT Gateway¹¹. It comes from AMS Gateway configuration, and is typically the same certificate used for the `User DN`¹².

In Server Requests and Client Responses, the SWIFT Gateway passes the `SignDN` to the AMS Gateway to allow the AMS to do additional business sanity checking. The receiving AMS Gateway is expected to validate the `SignDN` by looking it up and checking it against any ACORD PartyIds in the business payload.

⁹ Also known as a CUG, a Closed User Group is a subset of related customers with access to certain SWIFT services and products in a defined context.

¹⁰ The SWIFT Gateway typically includes a dedicated Hardware Security Module (HSM) to store all certificates.

¹¹ The SWIFT Gateway typically includes a dedicated Hardware Security Module (HSM) to store all certificates.

¹² They may be different under certain security configurations that are beyond the scope of Phase 1a

6 WS-Security SOAP Header for Local Authentication Protocol (LAU)

As shown in Figure 7 on page 19, SWIFT's Local Authentication Protocol works together with SWIFTNet security to replace ACORD's current end-to-end SSL-based authentication.

LAU addresses the challenge of extending SWIFTNet's integrity and identity guarantees to the additional hops inside the SWIFT Gateway, to the AMS. This is where *Local Authentication* comes in. It protects the local leg between the AMS and the SWIFT Gateway.

SWIFT's LAU implementation is designed with the following considerations in mind:

- It is trivial for end users.
- A single, consistent implementation removes the need to negotiate multiple bilateral agreements.
- It leaves the most flexibility in future to add central value-added services that may inspect and enrich the message en-route.

Messages between an AMS Gateway and the SWIFT Gateway must be authenticated to guarantee end-to-end:

- **Integrity:** Ensures that a message is not tampered between AMS and the SWIFT Gateway.
- **Identity:** The SWIFT gateway and network provide detailed access controls, and Local Authentication allows identity to be propagated with each message. The SWIFT Gateway authenticates this identity to enable specific access rights for the message. If multiple users share an AMS through a service bureau, then the identity must accurately represent the originator of the message.

Note Although LAU is strictly optional from the perspective of SWIFTNet, the solution proposed for Rüşchlikon is for all AMS implementations to use SWIFT Local Authentication in a consistent and disciplined fashion¹³.

6.1 Adding LAU to Client Requests and Server Responses

Local authentication of messages (and SOAP Faults) is achieved as follows:

- The AMS Gateway and the SWIFT Gateway share a copy of a 32-byte shared local authentication secret (a bilateral key).

Notes on bilateral key (BK):

1. *The BK is configured in the message partner profile of the SWIFT Gateway (see Figure 8).*
 2. *For additional security, SWIFT manages the BK as two 16-byte parts (left and right).*
 3. *Each BK part must contain at least one uppercase, one lowercase, and one numeric character.*
- The 32-byte bilateral key must be securely managed on the AMS Gateway.

Note: SWIFT suggests that the AMS follows the left-right paradigm for consistency for the user.

¹³ SWIFT certification of AMS implementations may also be mandated, depending on user feedback.

- A local message authentication code (LMAC) is used to authenticate each message. The LMAC is a signature computed from the message content and the AMS Gateway's bilateral key.
- The LMAC is 256 bits, and is computed using HMAC-SHA256 [RFC-HMAC] [FIPS-SHA], according to [WS-BASIC-SEC].
- The LMAC is propagated between AMS and the SWIFT Gateway in a single WS-Security [WS-SECURITY] standard header as in the sample XML Example 9.
- Signatures are calculated on the digests of the [XML-E-C14N] canonicalised form of the SOAP body and header blocks as described in [WS-SECURITY].
- SOAP attachments, if present, are considered part of the business message. Each header block and each attachment must be individually signed and referenced in the Local Authentication WS Security block, according to Attachment Complete Signature Transform (ACST) [W3C-SWA].

Note There is a known limitation in SWIFT's current support of ACST. The SWIFT Gateway currently¹⁴ only accounts for the "text/plain" Content-Type. MIME parts of other types are considered as "binary", and so are left as is. Thus, "text/xml" MIME parts are NOT canonicalised with C14N as per the standard. To avoid future backwards compatibility issues when SWIFT adds support for "text/xml" Content-Types, AMS Gateways must set the attachment Content-Type to "application/octet-stream" instead of "text/xml" (see XML Example 11). To ensure that this important type information is not lost, the FileFormatCd element has been made mandatory in the Rüşchlikon AMS Subset defined by ACORD. This must be populated with the correct "text/xml" value.

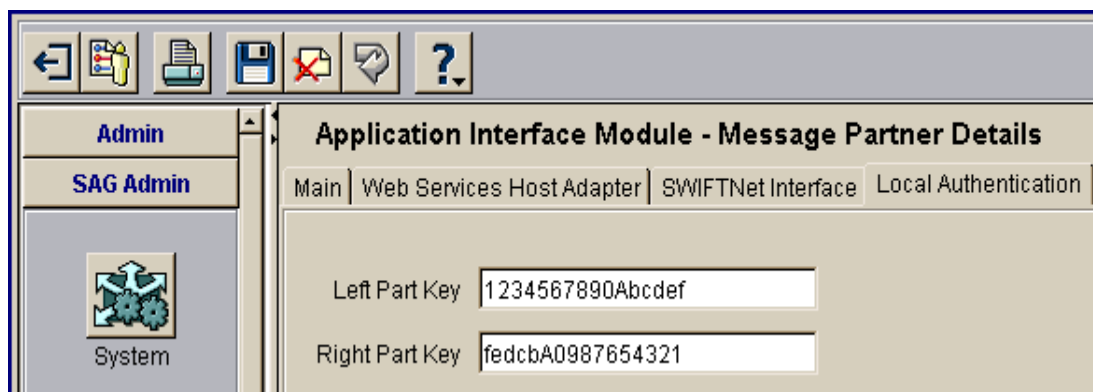


Figure 8: Configuring the bilateral key in the SWIFT Gateway

```
<wsse:Security xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"
  SOAP-ENV:actor="urn:swift:sag"
  SOAP-ENV:mustUnderstand="1">
  <ds:Signature>
```

¹⁴ Please be aware that this known limitation may be fixed in the future.

```

    <ds:SignedInfo>
      <ds:CanonicalizationMethod
ds:Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <c14n:InclusiveNamespaces PrefixList="" />
      </ds:CanonicalizationMethod>
      <ds:SignatureMethod
ds:Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256" />
        <ds:Reference URI="#SAGHeader">
          <ds:Transforms>
            <ds:Transform
ds:Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <c14n:InclusiveNamespaces c14n:PrefixList="" />
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
ds:Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <ds:DigestValue> ... </ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#SOAPBody">
            <ds:Transforms>
              <ds:Transform
ds:Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <c14n:InclusiveNamespaces c14n:PrefixList="" />
              </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod
ds:Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
              <ds:DigestValue> ... </ds:DigestValue>
            </ds:Reference>
            <ds:Reference URI="cid:Attachment1">
              <ds:Transforms>
                <ds:Transform Algorithm="http://docs.oasis-
open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Complete-Signature-
Transform" />
              </ds:Transforms>
              <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                <ds:DigestValue>j6lvEPO...tMueVu8nk=</ds:DigestValue>
              </ds:Reference>
            </ds:Reference>
          </ds:SignatureMethod>
        <ds:SignatureValue> ... </ds:SignatureValue>
      </ds:SignedInfo>
    </wsse:Security>

```

XML Example 9: Sample WS Security Header added by an AMS Gateway to carry WSHA-specific Local Authentication information to the local SWIFT Gateway.

Note XPath filters are supported, but XML Example 9 shows the simpler XPointer shorthand to reference a `wsu:Id` in each signed element.

6.1.1 SWIFT Gateway Response to an LAU Failure

If the client SWIFT Gateway detects an LAU problem on a Client Request, then it locally raises and returns a SOAP Fault that indicates an LAU problem on the client. The problem is also logged on the SWIFT Gateway Event Journal, where an SNMP alert may also be raised.

If the server SWIFT Gateway detects an LAU problem on a Server Response, then it discards the response. Instead, it returns a SOAP Fault to the client, to indicate an LAU Failure on the server. The event is also logged on the server where an SNMP alert may be raised.

The server Fault is propagated to the client AMS which may retry. Since such an exception usually requires manual investigation and intervention on the server, the client AMS may also raise a helpful alert to guide the user to contact the receiver gateway administrator.

6.2 LAU Check on Server Requests and Client Responses

When a SWIFT Gateway sends a Server Request or a Client Response to an AMS Gateway, it adds a WS-Security-compliant LAU header. The signature is calculated using the bilateral key stored on the SWIFT Gateway and is addressed to the SOAP actor named "urn:swift:sag:ws.handler.lau" as shown in XML Example 10.

To ensure that the message has not been tampered in transit, the receiving AMS Gateway must recalculate the LAU with its copy of the bilateral key.

```
<wsse:Security SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
      wsu:Id="WSHA1242750899859270">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
        <cl4n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
      </ds:CanonicalizationMethod>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
      <ds:Reference URI="#WSHA1242...99859269">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
            <cl4n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmldsig#sha256" />
        <ds:DigestValue>...</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#SOAPBody">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
            <cl4n:InclusiveNamespaces
```

```

xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
  </ds:Transform>
</ds:Transforms>
  <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
  <ds:DigestValue>...</ds:DigestValue>
</ds:Reference>
  <ds:Reference URI="cid:AcordBusinessMessage">
    <ds:Transforms>
      <ds:Transform Algorithm="http://docs.oasis-open.org/wss/oasis-
wss-SwAProfile-1.1#Attachment-Complete-Signature-Transform" />
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <ds:DigestValue>...</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
  <ds:SignatureValue>...</ds:SignatureValue>
</ds:Signature>
</wsse:Security>

```

XML Example 10: Sample WS Security Header added by the SWIFT Gateway to carry WSHA-specific Local Authentication information to the target AMS Gateway.

6.2.1 AMS Gateway Response to an LAU Failure

If a Client AMS detects an LAU problem on a Client Response, then the Client Response must be discarded and a serious alert raised. The AMS gateway may retry the message, but since an LAU problem typically requires manual investigation and intervention, processing may also be halted until this has occurred.

If a Server AMS detects an LAU problem on a Server Request, then the Server Response must be discarded and a serious alert must be raised. A SOAP Fault that indicates the LAU failure can be returned to the Server SWIFT Gateway. This allows the SWIFT Gateway to log the failure and raise an SNMP alert if configured to do so.

If the SOAP Fault is signed, then it is propagated to the AMS Client, so the message can guide the receiver to a remote problem on the receiver's gateway.

7 Exception Handling

Any errors detected by SWIFT Gateway or Network are mapped to standard SOAP 1.1 faults like those shown in XML Example 15, XML Example 16, and XML Example 17. In Phase 1a, an AMS Gateway must expect to receive SOAP Faults from the local SWIFT Gateway, the remote SWIFT Gateway, and the remote server application.

SOAP faults generated internally by the SWIFT Gateway and SWIFTNet contain specific additional data in the details element to help the AMS report and act appropriately.

SOAP Faults that originate at the ACORD Server or AMS server must include the server authoriser DN in the SAGHeader block. This allows SWIFTNet to transfer the original fault to the client, otherwise it is logged locally, and a generic SOAP Fault is transferred to the client.

Later Phases 1b and 1c enrich the Fault handling to provide additional value and machine-readable error information from central services. This is expected to be the main development effort for later phases, which otherwise focus on delivery and testing of central services.

Table 2 shows the high-level format of a SWIFT SOAP Fault. For complete details of expected values, see [WSHA-DG] and [SAG-DG].

Type	Definition	Description
fault-text	<pre>string matches (Text Summary) with Summary ::= Severity " - " Code " - " Text Severity ::= "Transient" "Logic" "Fatal" "Warning" Code ::= (Word ":")? Word "." Word "." Word Word ::= alphanum+ Text ::= pr-ascii*</pre>	<p>The fault text can be anything.</p> <p>When the SWIFT Gateway raises the fault the text field contains the summary of the status.</p>
fault-urn	<pre>string matches (pr-ascii* "urn:swift:sag")</pre>	<p>The fault URN can be anything.</p> <p>When the SWIFT Gateway raises the fault, it contains the URN of WSHA.</p>
fault-detail	<pre>{ ANY wsha:ExcStatus }</pre>	<p>The fault detail can be anything in XML.</p> <p>When the SWIFT Gateway raises the fault the text field contains a complete WSHA status represented in XML.</p>

Table 2: Format of SWIFT SOAP Fault contents

8 Other SOAP Header and Message Requirements

8.1 Non-SWIFT SOAP Headers

WSHA does not limit how SOAP header blocks are used between Web service applications. This ensures that applications are generally interoperable over HTTPS and SWIFTNet.

WSHA is compliant with the standards listed in section 13 however, so all SOAP headers must also be fully compliant to ensure expected behaviour.

8.2 SOAP Version

WSHA supports only SOAP 1.1.

8.3 XML Encoding

WSHA supports only UTF-8 for the XML character encoding.

8.4 MIME Message Structure

WSHA requires that the SOAP envelope *must be the first part* of the MIME message.

Section 11 includes an example of a MIME message and header that is generated by the SWIFT Gateway.

9 Other General Considerations

9.1 Size Limits

SWIFT defines the following size limits:

- WSHA supports SOAP messages (body and headers) with a size up to 2,000,000 octets (2Mb).
- WSHA supports SOAP messages and attachments with a size up to 50,000,000 octets (50Mb).

Note: These limits are defined on the sizes seen at WSHA. There is no direct relation between these sizes and the sizes seen at the application. The actual size depends on several factors, including:

- *Applications, intermediate SOAP nodes, and HTTP may use different character encodings. Each character encoding may change the message size.*
- *Intermediate SOAP nodes, XML utilities, and WSHA may use XML serialisation and de-serialisation methods that may affect message size.*

9.2 Response Times

SWIFT has a maximum response time of 5 minutes for SOAP requests. That is, WSHA expects to start receiving a SOAP response from the server side within 5 minutes. The SOAP response itself may take longer than 5 minutes to complete. This may be the case, for example, if the response is very large. If the response is not initiated within 5 minutes, then the local WSHA sends a SOAP fault to the client application.

This is expected to be sufficient for ACORD applications that typically deal with 60-second timeouts. The AMS gateway must manage SWIFT timeouts independently of business-level timeouts, however.

9.3 Simultaneous Connections

WSHA can support up to 500 simultaneous HTTPS connections to WS applications.

This is expected to be sufficient for the Rüşchlikon project.

10 Phase 1a Test Requirements

SWIFT supports two levels of tests to validate that participants have successfully completed phase 1a.

10.1 Initial Message Syntax Tests

To help AMS providers validate that they have successfully implemented the recommendations in this document, SWIFT supports initial syntax tests. These confirm that an AMS produces correctly formatted messages that the SWIFT Gateway and Network can accept and process.

Please find details in section 10.

Note It is not possible to test connectivity, protocol, and interoperability in this manner, so this is addressed in the next phase of tests.

10.2 Detailed Connectivity, Protocol, and Interoperability Tests

When an AMS is able to produce SWIFT formatted messages, more detailed tests can be scheduled. These tests require a connection to a live SWIFT Gateway, so can either be done onsite in the SWIFT labs, or onsite with a registered participant's gateway.

In addition to detailed SWIFTNet interoperability tests, SWIFT can help facilitate interoperability tests with counterparties and between AMS providers at this time.

To support different kinds of tests, SWIFT has prepared several test environments to analyse your messages, and to ensure that the SWIFT Gateway and network can process them.

To participate, please contact SWIFT Support as described in section 12.

11 Sample Messages

This section shows sample messages based on an imaginary test environment at AON Benfield sending a technical account to an imaginary test environment at Swiss Re. Swiss Re responds with a PostRs. Examples of the kinds of Faults that may be generated are also shown.

The sample messages are based on these imaginary environment configurations:

Sample Client Configuration (AON Benfield)

Element	Value
ApplicationId	wsha_client
Requestor	cn=ams-test,o=aonbgb21,o=swift
Responder	cn=ams-test,o=swrechcz,o=swift
Service	swift.reinsurance!p
RequestType	Any
UserDN	cn=ams-test,o=aonbgb21,o=swift
SignDN	cn=ams-test,o=aonbgb21,o=swift
LAU Key Left Part	1234567890abcdef
LAU Key Right Part	fedcbA0987654321

Table 3: Sample configuration that represents an imaginary test environment at AON

Sample Server Configuration (Swiss Re)

Element	Value
ApplicationId	wsha_server
UserDN	cn=ams-test,o=swrechcz,o=swift
SignDN	cn=ams-test,o=swrechcz,o=swift
LAU Key Left Part	1234567890abcdef
LAU Key Right Part	fedcbA0987654321

Table 4: Sample configuration that represents an imaginary test environment at Swiss Re

```

-----=_MIME_Part_
Content-Type: text/xml; charset=utf-8
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
      wsu:Id="SAGHeader"
      SOAP-ENV:actor="urn:swift:sag"
      SOAP-ENV:mustUnderstand="1">

```

```

<wsha:ApplicationId>wsha_client</wsha:ApplicationId>
<wsha:RequestHeader>
  <wsha:Requestor>cn=ams-test,o=aonbgb21,o=swift</wsha:Requestor>
  <wsha:Responder>cn=ams-test,o=swrechcz,o=swift</wsha:Responder>
  <wsha:Service>swift.reinsurance!p</wsha:Service>
  <wsha:RequestType>any</wsha:RequestType>
</wsha:RequestHeader>
<wsha:AuthorisationContext>
  <wsha:UserDN>cn=ams-test,o=aonbgb21,o=swift</wsha:UserDN>
</wsha:AuthorisationContext>
<wsha:Security>
  <wsha:SignDN>cn=ams-test,o=aonbgb21,o=swift</wsha:SignDN>
</wsha:Security>
</wsha:SAGHeader>
<wsse:Security SOAP-ENV:actor="urn:swift:sag"
  SOAP-ENV:mustUnderstand="1"
  xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
ds:Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
        <c14n:InclusiveNamespaces PrefixList="" />
      </ds:CanonicalizationMethod>
      <ds:SignatureMethod
ds:Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256" />
      <ds:Reference URI="#SAGHeader">
        <ds:Transforms>
          <ds:Transform ds:Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
            <c14n:InclusiveNamespaces c14n:PrefixList="" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
ds:Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>NpBCKVg...VNjB01h8=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#SOAPBody">
        <ds:Transforms>
          <ds:Transform ds:Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
            <c14n:InclusiveNamespaces c14n:PrefixList="" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
ds:Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>f8gqLdp91UD...vbbitw0kDz=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="cid:Attachment1">

```

```

        <ds:Transforms>
            <ds:Transform Algorithm="http://docs.oasis-
open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Complete-Signature-
Transform"/>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>j6lwx3rvEPO...tMup4NbeVu8nk=</ds:DigestValue>
        </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>ZTfEWe...axIh8=</ds:SignatureValue>
        </ds:Signature>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body wsu:Id="SOAPBody"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
    <ac:PostRq xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Inbox"
        xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1.4.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1.4.0 AcordMsgSvc_v-1-4-0.xsd">
        <ac:Sender>
            <ac:PartyId>urn:duns:211637244</ac:PartyId>
            <ac:PartyRoleCd>Broker</ac:PartyRoleCd>
        </ac:Sender>
        <ac:Receiver>
            <ac:PartyId>urn:duns:093841880</ac:PartyId>
            <ac:PartyRoleCd>Reinsurer</ac:PartyRoleCd>
        </ac:Receiver>
        <ac:Application>
            <ac:ApplicationCd>Jv-Ins-Reinsurance</ac:ApplicationCd>
            <ac:SchemaVersion>http://www.ACORD.org/standards/Jv-Ins-
Reinsurance/2005-2</ac:SchemaVersion>
        </ac:Application>
        <ac:TimeStamp>2009-02-02T11:42:06.446+00:00</ac:TimeStamp>
        <ac:MsgItem>
            <ac:MsgId>6e5cdb62-de43-11dd-8800-001a4bf2803f</ac:MsgId>
            <ac:MsgTypeCd>TechAccount</ac:MsgTypeCd>
        </ac:MsgItem>
        <ac:SecurityProfileCd>Basic</ac:SecurityProfileCd>
        <ac:WorkFolder>
            <ac:MsgFile>
                <ac:FileId>cid:Attachment1</ac:FileId>
                <ac:FileFormatCd>text/xml</ac:FileFormatCd>
            </ac:MsgFile>
        </ac:WorkFolder>
    </ac:PostRq>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

-----=_MIME_Part_
Content-Type: application/octet-stream
Content-Transfer-Encoding: 8bit
Content-Id: <Attachment1>
Content-Length: 4175
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Jv-Ins-Reinsurance xsi:schemaLocation="http://www.ACORD.org/standards/Jv-
Ins-Reinsurance/2005-2 xml-schema/Jv-Ins-Reinsurance-2005-2.xsd"
    Version="2005-2"
    xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1.4.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.ACORD.org/standards/Jv-Ins-Reinsurance/2005-2">
  <TechAccount Sender="broker" Receiver="reinsurer">
    ...
  </TechAccount>
</Jv-Ins-Reinsurance>
-----=_MIME_Part_

```

XML Example 11: Sample Client Request Message containing Technical Account PostRs sent from Client (AON) AMS to Client (AON) SWIFT Gateway

```

POST /SwiftConnector/SagEndPoint.aspx HTTP/1.1
Accept-Encoding: gzip
Content-Type: multipart/related;type="text/xml";boundary="-----
=_Part_33_7643448.1242750899859"
SOAPAction: ""
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.5.0_08
Host: 172.25.87.201:444
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Length: 9635

-----=_Part_33_7643448.1242750899859
Content-Type: text/xml;charset=utf-8
Content-Transfer-Encoding: 8bit
Content-ID: <ROOTPART>

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
        SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
        wsu:Id="WSHA1242750899859269">

```

```

<wsha:ApplicationId>wsha_server</wsha:ApplicationId>
<wsha:Security>
  <wsha:SignDN>cn=ams-test,o=aonbgb21,o=swift</wsha:SignDN>
</wsha:Security>
</wsha:SAGHeader>
<wsse:Security SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
wsu:Id="WSHA1242750899859270">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
      </ds:CanonicalizationMethod>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
      <ds:Reference URI="#WSHA1242750899859269">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
            <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>2TCm4u...bcrAIlfgljxWoQ=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#SOAPBody">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
            <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>RLYoCLHUZ...52fnOIK5H6N4=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="cid:AcordBusinessMessage">
        <ds:Transforms>
          <ds:Transform Algorithm="http://docs.oasis-
open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Complete-Signature-
Transform" />
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>R8D/DQm...z4yn/Op83Q=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>ZUb9fH8l...n/2v0Ytro=</ds:SignatureValue>
  </ds:Signature>
</wsse:Security>

```

```

    </ds:Signature>
  </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body wsu:Id="SOAPBody"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
  <ac:PostRq xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Inbox"
    xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1.3.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1.3.0 AcordMsgSvc_v-1-3-0.xsd">
    <ac:Sender>
      <ac:PartyId>urn:duns:293452348</ac:PartyId>
      <ac:PartyRoleCd>Broker</ac:PartyRoleCd>
    </ac:Sender>
    <ac:Receiver>
      <ac:PartyId>urn:duns:399317270</ac:PartyId>
      <ac:PartyRoleCd>Reinsurer</ac:PartyRoleCd>
    </ac:Receiver>
    <ac:Application>
      <ac:ApplicationCd>Jv-Ins-Reinsurance</ac:ApplicationCd>
      <ac:SchemaVersion>http://www.ACORD.org/standards/Jv-Ins-
Reinsurance/2005-1</ac:SchemaVersion>
    </ac:Application>
    <ac:TimeStamp>2008-04-29T12:28:46.611+01:00</ac:TimeStamp>
    <ac:MsgItem>
      <ac:MsgId>c164cc72-15dd-11dd-af35-0015600481de</ac:MsgId>
      <ac:MsgTypeCd>TechAccount</ac:MsgTypeCd>
    </ac:MsgItem>
    <ac:WorkFolder>
      <ac:MsgFile>
        <ac:FileId>cid:8f392eb0-ec98-4f33-9378-f0a1e1e843f1</ac:FileId>
        <ac:FileFormatCd>text/xml</ac:FileFormatCd>
      </ac:MsgFile>
    </ac:WorkFolder>
  </ac:PostRq>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
-----_Part_33_7643448.1242750899859
Content-Type: application/octet-stream
Content-Transfer-Encoding: 8bit
Content-ID: <AcordBusinessMessage>

<Jv-Ins-Reinsurance:Jv-Ins-Reinsurance xmlns:Jv-Ins-
Reinsurance="http://www.ACORD.org/standards/Jv-Ins-Reinsurance/2005-1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  Version="2005-1" xsi:schemaLocation="">
  <Jv-Ins-Reinsurance:TechAccount Receiver="reinsurer" Sender="broker">
    ...
  </Jv-Ins-Reinsurance:TechAccount>

```

```
</Jv-Ins-Reinsurance:Jv-Ins-Reinsurance>
-----_Part_33_7643448.1242750899859-
```

XML Example 12: Sample Server Request Message sent from Server (Swiss Re) SWIFT Gateway to Server (Swiss Re) AMS Gateway. Contains original Technical Account PostRs from XML Example 11. Note that this shows an example of the MIME Headers added by the SWIFT Gateway.

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wsssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wsssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>
    <wsha:SAGHeader SOAP-ENV:actor="urn:swift:sag"
      SOAP-ENV:mustUnderstand="1"
      wsu:Id="SAGHeader"
      xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0">
      <wsha:ApplicationId>wsha_server</wsha:ApplicationId>
      <wsha:AuthorisationContext>
        <wsha:UserDN>cn=ams-test,o=swrechcz,o=swift</wsha:UserDN>
      </wsha:AuthorisationContext>
      <wsha:Security>
        <wsha:SignDN>cn=ams-test,o=swrechcz,o=swift</wsha:SignDN>
      </wsha:Security>
    </wsha:SAGHeader>
    <wsse:Security SOAP-ENV:actor="urn:swift:sag" SOAP-
ENV:mustUnderstand="1">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <c14n:InclusiveNamespaces PrefixList=""
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
          <ds:Reference URI="#SOAPBody">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                <c14n:InclusiveNamespaces PrefixList=""
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
              </ds:Transform>
            </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmllenc#sha256" />
          <ds:DigestValue>YTXZ2Z21ZxU...k97wqrPEEUjY=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#SAGHeader">
          <ds:Transforms>
```

```

        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
            <c14n:InclusiveNamespaces PrefixList=""
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
    <ds:DigestValue>w6vgW6FcLAO...5io6f9bbEtFcgTpA=
    </ds:DigestValue>
    </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>71HUA9xWYSj7...zlc7HWodQUZz11+WsQ=
    </ds:SignatureValue>
    </ds:Signature>
</wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body wsu:Id="SOAPBody">
    <ac:PostRs xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Inbox"
xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1.4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1.4.0 AcordMsgSvc_v-1-4-0.xsd">
        ...
    </ac:PostRs>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XML Example 13: Sample Server Response Message returned by Server (Swiss Re) AMS Gateway to Server (Swiss Re) SWIFT Gateway

```

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wssse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
    <SOAP-ENV:Header>
        <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
wsu:Id="WSHA1242756192095315">
            <wsha:ApplicationId>wsha_client</wsha:ApplicationId>
            <wsha:Security>
                <wsha:SignDN>cn=ams-test,o=swrechcz,o=swift</wsha:SignDN>
            </wsha:Security>
        </wsha:SAGHeader>
        <wsse:Security SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
wsu:Id="WSHA1242756192095316">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>

```

```

      <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList=""/>
        </ds:CanonicalizationMethod>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256"/>
        <ds:Reference URI="#WSHA1242756192095315">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
              <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList=""/>
              </ds:Transform>
            </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <ds:DigestValue>deci9YxpZg...GMRlKJ3BZLbTYs=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#SOAPBody">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList=""/>
                </ds:Transform>
              </ds:Transforms>
            <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
              <ds:DigestValue>YTXZ2Z2lZ...97wqrPEEUUjY=</ds:DigestValue>
            </ds:Reference>
          </ds:SignedInfo>
          <ds:SignatureValue>vf2odEl...NsWbsd8=</ds:SignatureValue>
        </ds:Signature>
      </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="SOAPBody">
      <ac:PostRs xmlns="http://www.ACORD.org/Standards/AcordMsgSvc/Inbox"
xmlns:ac="http://www.ACORD.org/Standards/AcordMsgSvc/1.4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ACORD.org/Standards/
AcordMsgSvc/1.4.0 AcordMsgSvc_v-1-4-0.xsd">
        ...
      </ac:PostRs>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

```

XML Example 14: Sample Client Response Message returned by Client (AON) SWIFT Gateway to Client (AON) AMS Gateway. The body contains the PostRs from XML Example 13

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
      SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd"
      wsu:Id="WSHA1242756192095315">
      <wsha:ApplicationId>wsha_client</wsha:ApplicationId>
    </wsha:SAGHeader>
    <wsse:Security SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
      wsu:Id="WSHA1242756192095316">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <cl4n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
          <ds:Reference URI="#WSHA1242756192095315">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                <cl4n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
              </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <ds:DigestValue>dec19YxpZ...GMR1KJ3BZLbTYs=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#SOAPBody">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                <cl4n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
              </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
            <ds:DigestValue>YTXZ2Z2...7wqrPEEUUjY=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>vf2odEl5lD...HV9TxpNsWbsd8=</ds:SignatureValue>
      </ds:Signature>
    </wsse:Security>
  </SOAP-ENV:Header>

```

```

<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  wsu:Id="SOAPBody">
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Client</faultcode>
    <faultstring>
      Transient - Sag:APL-WSHA.001.001 - Invalid format
    </faultstring>
    <faultactor>urn:swift:sag</faultactor>
    <detail>
      <wsha:ExcStatus xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0">
        <wsha:Code>Sag:APL-WSHA.001.001</wsha:Code>
        <wsha:Severity>Transient</wsha:Severity>
        <wsha:Text>Invalid format</wsha:Text>
        <wsha:Action>
          Correct the format or syntax of the message
        </wsha:Action>
        <wsha>Data>
          Incomplete SAGHeader element: missing Security/SignDN element
        </wsha>Data>
      </wsha:ExcStatus>
    </detail>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XML Example 15: Sample of a Client SOAP Fault Response generated by the local SWIFT Gateway. This example indicates a problem with the format of the SOAP Header. Note that the SAGHeader is incomplete because the information was not included in the original message.

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
      SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="WSHA1242756192095315">
      <wsha:ApplicationId>wsha_client</wsha:ApplicationId>
      <wsha:Security>
        <wsha:SignDN>cn=ams-test,o=aonbgb21,o=swift</wsha:SignDN>
      </wsha:Security>
    </wsha:SAGHeader>
    <wsse:Security SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
      wsu:Id="WSHA1242756192095316">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <c14n:InclusiveNamespaces

```

```

xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
  </ds:CanonicalizationMethod>
  <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
  <ds:Reference URI="#WSHA1242756192095315">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
        <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
        </ds:Transform>
      </ds:Transforms>
      <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
      <ds:DigestValue>dec19YxpZ...GMRIKJ3BZLbTYs=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#SOAPBody">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
          <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
        <ds:DigestValue>YTXZ2Z2...7wqrPEEUUjY=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>vf2odEl5lD...HV9TxpNsWbsd8=</ds:SignatureValue>
  </ds:Signature>
</wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"
  wsu:Id="SOAPBody">
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>
      Transient - Sag:APL-WSHA.003.001 - Failure at SNL level
    </faultstring>
    <faultactor>urn:swift:sag</faultactor>
    <detail>
      <wsha:ExcStatus xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0">
        <wsha:Code>Sag:APL-WSHA.003.001</wsha:Code>
        <wsha:Severity>Transient</wsha:Severity>
        <wsha:Text>Failure at SNL level</wsha:Text>
        <wsha:Action>
          See details for cause and corrective action
        </wsha:Action>
        <wsha:Data>

```

```

        Status StatusAttributes
        Severity = Logic
        Code = Sw.Gbl.HSMContextLost
        Parameter = DN=cn=partner5,ou=cs-be,o=swhqbebb,o=swift
        ...
    </wsha:Data>
</wsha:ExcStatus>
</detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XML Example 16: Sample of a Client SOAP Fault Response generated by SWIFTNet. This example indicates a problem with the certificate that was used to sign the message.

```

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
  <SOAP-ENV:Header>
    <wsha:SAGHeader SOAP-ENV:actor="urn:swift:sag"
        SOAP-ENV:mustUnderstand="1"
        wsu:Id="SAGHeader"
        xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0">
      <wsha:ApplicationId>wsha_server</wsha:ApplicationId>
      <wsha:AuthorisationContext>
        <wsha:UserDN>cn=swrechzh-ams-ti,o=swrechzh,o=swift</wsha:UserDN>
      </wsha:AuthorisationContext>
      <wsha:Security>
        <wsha:SignDN>cn=swrechzh-ams-ti,o=swrechzh,o=swift</wsha:SignDN>
      </wsha:Security>
    </wsha:SAGHeader>
    <wss:Security SOAP-ENV:actor="urn:swift:sag" SOAP-
ENV:mustUnderstand="1">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <c14n:InclusiveNamespaces PrefixList=""
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256"/>
          <ds:Reference URI="#SOAPBody">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                <c14n:InclusiveNamespaces PrefixList=""
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />

```

```

        </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>YTXZ2Z...EEUUjY=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#SAGHeader">
    <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
            <c14n:InclusiveNamespaces PrefixList=""
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>w6vgW6FcL...EtFcgTpA=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>71HUA9xW...odQUZz1l+WsQ=</ds:SignatureValue>
</ds:Signature>
</wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="SOAPBody">
    <SOAP-ENV:Fault xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
        ...
    </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

XML Example 17: Sample of a Server SOAP Fault Response generated by the Server (Swiss Re) AMS Gateway.

```

HTTP/1.1 500 Internal Server Error
Date: Tue, 19 May 2009 18:02:21 GMT
Server: Oracle Containers for J2EE
Content-Length: 2569
Connection: Keep-Alive
Keep-Alive: timeout=15, max=100
Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">

```

```

<SOAP-ENV:Header>
  <wsha:SAGHeader xmlns:wsha="urn:swift:sag:xsd:wsha.header.1.0"
    SOAP-ENV:actor="urn:swift:sag:ws.handler.sag"
    wsu:Id="WSHA1242756192095315">
    <wsha:ApplicationId>wsha_client</wsha:ApplicationId>
    <wsha:Security>
      <wsha:SignDN>cn=ams-test,o=swrehcZ,o=swift</wsha:SignDN>
    </wsha:Security>
  </wsha:SAGHeader>
  <wsse:Security SOAP-ENV:actor="urn:swift:sag:ws.handler.lau"
    wsu:Id="WSHA1242756192095316">
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
        </ds:CanonicalizationMethod>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#hmac-sha256" />
        <ds:Reference URI="#WSHA1242756192095315">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
              <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <ds:DigestValue>decI9YxpZ...GMRlKJ3BZLbTYs=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="#SOAPBody">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
              <c14n:InclusiveNamespaces
xmlns:c14n="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="" />
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
          <ds:DigestValue>YTXZ2Z2...7wqrPEEUUjY=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>vf2odE15lD...HV9TxpNsWbsd8=</ds:SignatureValue>
    </ds:Signature>
  </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd"

```

```
wsu:Id="SOAPBody">
  <SOAP-ENV:Fault xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    ...
    ...
    ...
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

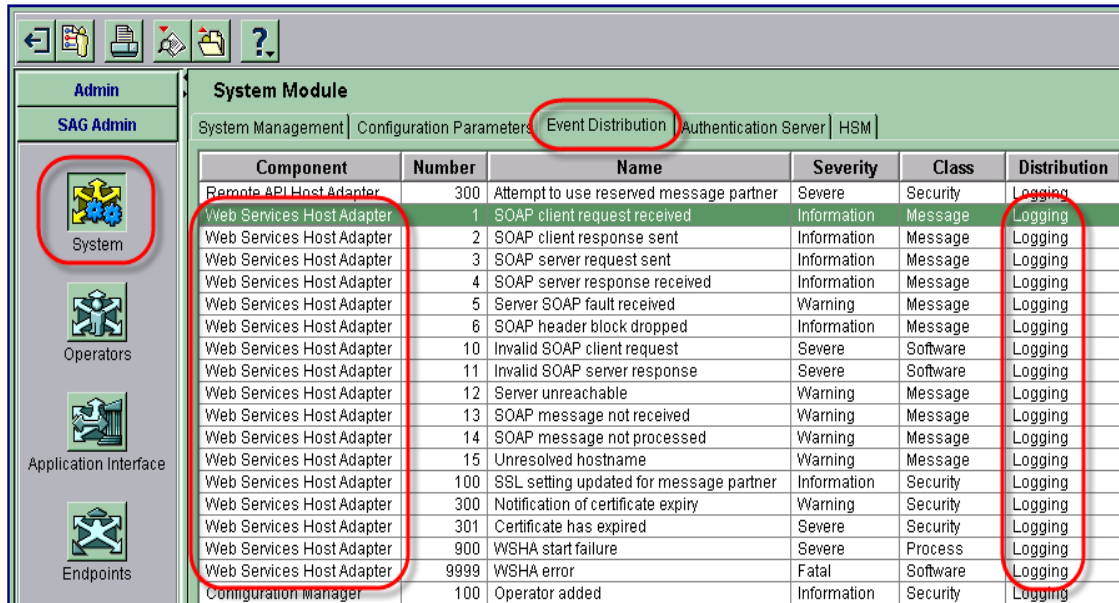
XML Example 18: Sample of a Client Fault Message returned by Client (AON) SWIFT Gateway to Client (AON) AMS Gateway. Body contains the SOAP Fault Response sent by the Server (Swiss Re) AMS Gateway (XML Example 17). Note that this shows an example of the HTTP Header added by the SWIFT Gateway.

12 Support

The SWIFT Integration Services team provides dedicated, expert integration and testing support throughout the Rüşchlikon pilot. To ensure that SWIFT can respond to all support queries in a timely and professional manner however, all communications must come through support@swift.com.

When sending an e-mail to SWIFT Support, please ensure that your company name and “Rüşchlikon” is in the subject line.

If the question is related to a suspected problem with the SWIFT Gateway, then follow these steps to ensure that SWIFT has all the information required to analyse:



Component	Number	Name	Severity	Class	Distribution
Remote API Host Adapter	300	Attempt to use reserved message partner	Severe	Security	Logging
Web Services Host Adapter	1	SOAP client request received	Information	Message	Logging
Web Services Host Adapter	2	SOAP client response sent	Information	Message	Logging
Web Services Host Adapter	3	SOAP server request sent	Information	Message	Logging
Web Services Host Adapter	4	SOAP server response received	Information	Message	Logging
Web Services Host Adapter	5	Server SOAP fault received	Warning	Message	Logging
Web Services Host Adapter	6	SOAP header block dropped	Information	Message	Logging
Web Services Host Adapter	10	Invalid SOAP client request	Severe	Software	Logging
Web Services Host Adapter	11	Invalid SOAP server response	Severe	Software	Logging
Web Services Host Adapter	12	Server unreachable	Warning	Message	Logging
Web Services Host Adapter	13	SOAP message not received	Warning	Message	Logging
Web Services Host Adapter	14	SOAP message not processed	Warning	Message	Logging
Web Services Host Adapter	15	Unresolved hostname	Warning	Message	Logging
Web Services Host Adapter	100	SSL setting updated for message partner	Information	Security	Logging
Web Services Host Adapter	300	Notification of certificate expiry	Warning	Security	Logging
Web Services Host Adapter	301	Certificate has expired	Severe	Security	Logging
Web Services Host Adapter	900	WSHA start failure	Severe	Process	Logging
Web Services Host Adapter	9999	WSHA error	Fatal	Software	Logging
Configuration Manager	100	Operator added	Information	Security	Logging

Figure 9: WSHA Events must be enabled when debugging

1. First, ensure that all WSHA events are enabled (System -> Event Distribution as shown in the screenshot in Figure 9).
2. Run (or ask your Service Bureau to run) the following commands (included with the Gateway install):

```
sag_supportinfo -config
```

```
sag_supportinfo -log -details [-startdate <date>] [-starttime <time>]
[-stopdate <date>] [-stoptime <time>]
```

The date format must be: YYYYMMDD

The time format must be: HH:MM:SS

If no `-startdate` is specified, then the default is to take the events of the last 24 hours.

Each of these commands describes where they have written their output.

The file named **sag_event_log_<timestamp>.zip** contains a complete dump of the information in the SWIFT Gateway Event Journal. Analysing this may help you to identify the problem without SWIFT support.

3. If you are unable to identify the problem yourself, then send the details to support@swift.com. Please ensure that you also send any exceptions received client and server, and a complete description of the problem.

4. When you receive a Case ID from SWIFT Support, follow the directions to logon and upload the outputs from step 2 as attachments to the case. If the resulting archives are too large, then you can split them with the following command:
`sag_supportinfo -split <file_pathname>`

13 References

The reader should be familiar with the following documents and Web services standards:

- [PDD-IWG] - *IWG (Industry Working Group) Electronic Industry Pilot for Reinsurance "Rüschlikon Initiative" Project Definition Document*
- [SAG-DG] - *Alliance Gateway Developer Guide*
- [WSHA-DG] - *Alliance Gateway Web Services Host Adapter Developer Guide*
- [SOAP-11] - *W3C Note - Simple Object Access Protocol (SOAP) Version 1.1*
- [WS-SECURITY] - *OASIS - SOAP Message Security 1.0*
- [WS-BASIC-SEC] - *WS-I - Basic Security Profile 1.0*
- [W3C-SWA] - *W3C Note - SOAP Messages with Attachments (SwA)*
- [WSS-SWA] - *OASIS - Web Services Security - SOAP Messages with Attachments - Profile 1.1*
- [XML-NS] - *W3C Recommendation - Namespaces in XML*
- [XML-E-C14N] - *W3C Recommendation - Exclusive XML-Canonicalisation version 1.0*
- [XML-SIGN] - *W3C Recommendation - XML-Signature Syntax and Processing*
- [RFC-HMAC] - *RFC 2104. HMAC: Keyed-Hashing for Message Authentication. February 1997*
- [FIPS-SHA] - *Federal Information Processing Standards Publications (FIPS PUB) 180-1. SHA: Secure Hash Algorithm*

Legal Notices

Copyright

SWIFT © 2009. All rights reserved.

You may copy this publication within your organisation. Any such copy must include these legal notices.

Confidentiality

This publication may contain SWIFT or third-party confidential information. Do not disclose this publication outside your organisation without the prior written consent of SWIFT.

Disclaimer

SWIFT supplies this publication for information purposes only. The information in this publication may change from time to time. You must always refer to the latest available version.

Translations

The English version of SWIFT documentation is the only official version.

Trademarks

SWIFT is the trade name of S.W.I.F.T. SCRL. The following are registered trademarks of SWIFT: SWIFT, the SWIFT logo, Sibos, SWIFTNet, SWIFTReady, and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.