[FORMAT]	EXTERNAL	28-JUL-22	UNCLASSIFIED
FORMAT	AUDIENCE	DATE	CLASSIFICATION

Australian Government
Australian Taxation Office

Created By AUSkey Replacement Project

Machine Authentication Service - Security Token Service (MAS-ST)

Service Definition

Version 1.0

UNCLASSIFIED

Document Owner: DCIS - ATO

Australian Government
Australian Taxation Office

ENDORSEMENT/VERSION CONTROL

CURRENT VERSION NUMBER	1.0	Revision	8	DATE	28/07/2022

<Enduring documents require business endorsement. Non-enduring document require approval. Depending on document type being built with from this template choose the appropriate header above and duplicate in the approval section following.>

APPROVAL			
[approver’s name]	[approver’s position]	[approved/endorsed]	[date]
[approver’s name]	[approver’s position]	[approved/endorsed]	[date]
[approver’s name]	[approver’s position]	[approved/endorsed]	[date]

VERSION CONTROL

Version	Revision date	Author	Summary of Change
0.1	19/10/2019	ATO	Initial Draft
0.2	21/11/2019	ATO	Minor changes and clarifications
0.3	17/12/2019	ATO	Minor changes and clarifications
1.0	28/07/2022	ATO	Minor changes to version number and document name following MAS-CM document release.

Document versioning is described in IT Standard 03

Table of Contents

Table of Figures

1 About This Document

1.1 Purpose

This document is a service definition of the Machine Authentication Service - Security Token service. This is otherwise known as MAS-ST.

As the goal of the MAS-ST service is to provide a SAML implementation that is compatible with the VANguard Security Token Service (STS), the document is based on the VANguard Security Token Service Definition v1.3 document and the ATO acknowledges the assistance provided by the Department of Industry, Innovation and Science (DIIS) in developing the MAS-ST service.

While the goal of the MAS-ST service is to be compatible with the SAML implementation of VANguard STS, there will be differences due to the different underlying technology used to deliver the service. Where any differences arise, these will be highlighted to assist existing customers of the VANguard STS to on board to MAS-ST.

It defines the interactions between an Initiating party and the STS, and the activities required to consume the service.

1.2 Scope

This document does not attempt to describe WS-* or SAML; it only covers the specifics of the implementation of these protocols for STS, and especially how the service treats optional or special parameters. This document should therefore be read in conjunction with the formal specifications of those protocols.

1.3 Intended Audience

This document is intended for use by service providers writing business software that is required to interact with Government agencies over the internet. It is intended for use by both business and technical persons.

Executive readers will gain insight into the service offering by reading the section 'Business Summary'

Business Analysts should read the whole document.

Developers and Testers should read the whole document

1.4 Changes since Last Contract Version

From v1.3 of the VANguard Contract, this contract:

- Supports both the VANguard v1.2 (SHA1) and the VANguard v1.3 (SHA256) services through different endpoints. This document mentions only v1.3 but v1.2 can be substituted. In particular, the ActAs functionality is available in both. Note that the v1.3 service is recommended for improved security.
- Supports M2M MAS Machine credentials issued by the relevant myGovID CA. These Machine Credentials are equivalent to AUSkey Device Credentials.
- The only credential type supported is ABR_Device as described in [Common Elements] below. These are issued by the relevant myGovID CA.
- Does not support Third Party Certificate Authorities.
- Supports AUSkey Device credentials issued by the relevant AUSkey CA.
- WS-Policy negotiation (for claims) is not supported.
- RSTR AttributeStatement Actor claim is not supported
- The majority of namespace aliases are different

1.5 Terminology

Term	Description
SAML	Secure Assertion Markup Language – an OASIS standard
Security Token	This is an industry term for an Assertion about identity that is issued by an Identity Provider. In the context of this contract, the MAS-ST issues a signed SAML Assertion. A serialisation of the claims that are digitally signed by the STS.
Initiating Party	This is the business user or application controlled by the user requesting a security token
Relying Party	This is the application that will consume the security token, for granting or denying access to resources
STS	Security Token Service
RST	Request for Security Token
RSTR	Request for Security Token Response
TSC	Technical Service Contract
SOAP	Simple Object Access Protocol – an internet messaging standard that utilises XML message constructs for exchanging messages in a distributed environment.
WS-Security	How to exchange security tokens or use tokens to protect the confidentiality and integrity of SOAP messages
WS-SecureConversation	How to optimize the use of security tokens for SOAP message security in multiple message exchange (e.g. session) scenarios
WS-Trust	How to request the security tokens needed to satisfy policy requirements and protect SOAP messages in a wide variety of trust relationships

1.6 References and Related Documents

[Common Elements] “Common Elements for VANguard Services” Document Revision V1.41

[WS-Trust] WS-Trust 1.3 OASIS Web Service Secure Exchange (**http://docs.oasis-open.org/ws-sx/wstrust/200512/ws-trust-1.3-os.html**)

[WS-Addressing] W3C Recommendation, "Web Services Addressing (WS-Addressing)", 9 May 2006.**http://www.w3.org/TR/2006/REC-ws-addr-core-20060509**

[WS-Policy] W3C Member Submission, "Web Services Policy 1.2 - Framework", 25 April 2006. (**http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/**)

[WSPolicyAttachment] W3C Member Submission, "Web Services Policy 1.2 - Attachment", 25 April 2006.(**http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/**)

[WS-Security] OASIS Standard, "OASIS Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)", March 2004. (**http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soapmessage-security-1.0.pdf** OASIS Standard, "OASIS Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", February 2006. (**http://www.oasis-open.org/committees/download.php/16790/wssv1.1-spec-os-SOAPMessageSecurity.pdf**)

[XML-C14N] W3C Recommendation, "Canonical XML Version 1.0", 15 March 2001. **http://www.w3.org/TR/2001/REC-xml-c14n-20010315**

[XML-Encrypt] W3C Recommendation, "XML Encryption Syntax and Processing", 10 December 2002. **http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/**

[XML-Signature] W3C Recommendation, "XML-Signature Syntax and Processing", 12 February 2002. **http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/**

2 Business View

2.1 Business Summary

2.1.1 Value to Relying Party

This service will allow an initiating party (IP) to request a security token that can be used to verify identity, with a relying party (RP). The RP can then use this token in order to determine if access to a protected resource should be granted to the IP.

The value of this service is that the RP can be assured that the IP’s credentials have been verified by the STS, and that the credential holder is who they claim to be for a given business transaction. The RP is leveraging myGovID identity infrastructure to allow the IP to gain access to the RP’s services. For example the STS verifies credentials issued by myGovID CA that an initiating party can use to lodge a software transmission via the Standard Business Reporting (SBR) platform. This removes the need for the RP to maintain its own authentication infrastructure.

Successful authentication of the IP will result in a security token being issued in the response. The RP will establish a trust relationship in advance with MAS-ST as a trusted service for issuing security tokens.

The initiating party software is responsible for dealing with SOAP faults that may be returned from the service (for example, missing mandatory elements). The relying party is responsible for validating aspects of the security token presented for a business transaction.

A policy may exist between the relying party and MAS-ST. The STS issues a security token to an IP, for consumption by the relying party (RP).

2.1.2 Broad Functionality

The principal operation of this service in this release is 'Issue'. Requests are formatted in accordance to the WS-Trust specification [WS-Trust]. The RequestSecurityToken (RST) request can only request a single token per request and is required to specify the token type returned, the only options supported are SAML 1.1 and SAML2.0 encoded token.

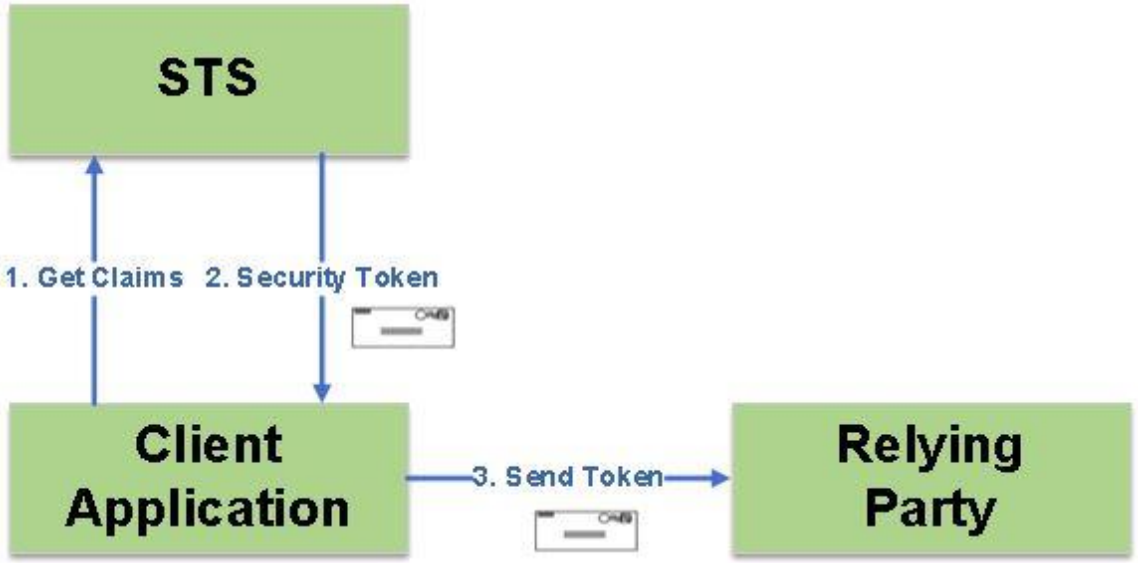


Figure 1: Basic Scenario

Figure 1: Basic Scenario shows a basic scenario of the interaction between the initiating party, relying party and MAS-ST under a typical scenario.

1. The client application initiates a request to MAS-ST requesting the claims required by the relying party. A default set of claims will be returned if claim elements omitted.
2. MAS-ST will validate the request and on success, will issue a security token that includes the claims requested to the initiating party for the relying party.
3. The security token is passed to the relying party from the initiating party, the relying party can choose to allow or deny access based on the validity of the token and their risk assessment of the token presented in order to conduct the business transaction.

Note: The default lifetime of a security token is 30 minutes.

2.1.3 Policy and Risks Security Tokens

MAS-ST will not issue a security token where it does not recognise the relying party, or the value for the claims requested specified as mandatory cannot be determined.

MAS-ST will generate a SOAP fault in response to the requests that cannot be satisfied.

Existing auditing and logging will be used to log these requests for internal use by the ATO.

The relying party is responsible upon receiving the Security Token to satisfy itself of the suitability of the security token in respect to the business transaction.

In summary the relying party is required to:

- Be satisfied that the issued security token was issued by MAS-ST by verifying the digital signature. Implement their own business rules in respect to the claims presented in the token.
- Verify the token lifetime is within acceptable bounds to proceed with transaction.
- Ensure the credential used to obtain the SAML token is acceptable for the transaction.
- Verify the session key contained in the security token for the relying party matches the key used to sign the service request from the IP. This verifies the services request is from the IP. Session keys are used to prove possession of the issued security token. A service request to the relying party should be signed with this session key issued by the STS encoded for both the IP and RP, the relying party should ensure service request received is signed with the session key contained in the MAS-ST issued security token and compare the equality of the session keys to ensure the transmission is from the verified business user and has not been intercepted.
- Ensure the claims presented are sufficient to conduct the business transaction.

2.2 Technical Summary

The Security Token Service will issue a token containing SAML encoded assertions about the initiating party. Business users/applications wishing to access electronic government services can request a security token to gain access to protected resources. MAS-ST will support a subset of requests detailed in WS-Policy v1.3 [WS-Trust].

This contract describes the internet-exposed web service that offers issuance of a security token for use by participating relying parties within the Australian Government.

Service access is governed by the Application (message) layer security. Access is determined by the endpoint address contained in the AppliesTo element of the RST and is restricted to known relying parties. Communication is secured using WS-Security and with confidentiality provided by SSL (configured in accordance with the Australian Government Protective Security Manual). All requests must be signed by Initiating Party and conform to the specification described in this document and the [Common Elements].

All requests are expressed as SOAPv1.2 requests

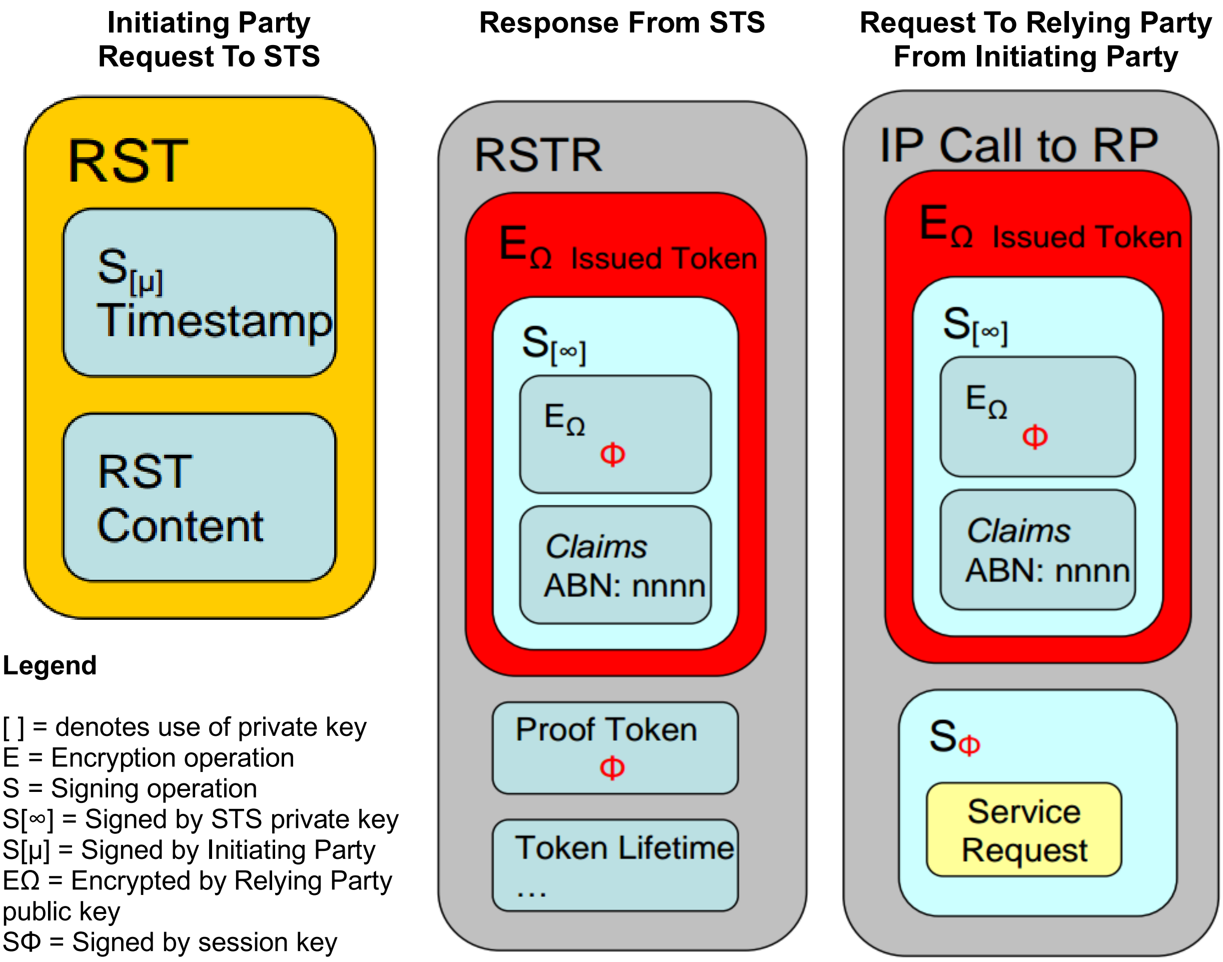
The security token is returned within the SOAP body as specified by the [WS-Trust] specification; the security token will be digitally signed using the MAS-ST Authentication private key.

The STS will validate the incoming RST by verifying the AppliesTo element of the RST is trusted. The STS manages a list of trusted Relying Parties for whom tokens can be issued, along with their public certificates and URI’s.

The STS returns a RSTR message. Contained in the response is a signed assertion termed Issued Token and a Signed Proof Token. The RSTR contains a WS-Security secured assertion for consumption by the relying party, and a proof token for the initiating party.

The following diagram is a visual indicator of the interactions between the three parties, of the encryption and signing that occurs for each interaction.

2.2.1 WS-Trust Message Structure



Legend

[ ] = denotes use of private key
E = Encryption operation
S = Signing operation
S[∞] = Signed by STS private key
S[μ] = Signed by Initiating Party
EΩ = Encrypted by Relying Party public key
SΦ = Signed by session key

Φ = Session key (Symmetric as requested by IP)

Figure 2: Message Structure

2.3 Relying Party Metadata

In order for the STS to issue tokens for a RP the STS requires the public key of the RP. This certificate will be held by the STS. For each endpoint the STS will use the RP certificate to apply crypto algorithms to secure the message exchange. This mechanism is used to prevent man-in-the-middle attack. The URI identifies the RP for which the initiating party is requesting the assertion. The endpoint is used to identify the certificate that will be used to encrypt the issued token contained in the RSTR response from MAS-ST.

2.4 Claims Processing

A RST request from an initiating party can make demands that specific claims be present in the assertion retuned in the security token issued by MAS-ST. This is normally a requirement expressed by a relying party to the initiating party.

MAS-ST allows for three options for evaluating claims data and uses the following rules in processing claims data for an RST request. Claims are categorised into distinct categories outlined below. MAS-ST will determine if a security token can be issued after successful verification of an initiating party's certificate and after evaluating the following rules.

Default Claims

If a request does not specify interest in specific claims and omits the claims element, a default set of claims for the initiating party certificate type presented will be evaluated and returned in the issued security token. The certificate will have a subset of the claims specified in [Common elements] set as the default claims.Please note that the only certificate type supported by MAS-ST is ABR_Device.

Compulsory Claims

A compulsory claim is a claim that MAS-ST will always evaluate and return for each request ignoring RST request setting. If a compulsory claim cannot be evaluated, a security token will not be issued.

Additionally, when the ActAs element is specified in the request, the resulting actor claim in the response also has this behaviour. That is, it will be returned whether requested or not.

Optional Claims

An optional claim is a claim expressed in the request with the 'optional' attribute equal to true. This instructs MAS-ST to process the claim specified as optional and if it cannot evaluated, a security token should be issued. The request will not fail to issue a security token based on the individual claim not being present. The claim will be omitted from assertion returned in the response as a value was undeterminable.

An optional claim with the optional attribute equal to false, instructs MAS-ST that if a claim value cannot be determined a security token should not be issued.

3 Technical View

3.1 Service Endpoints

The external endpoints are:

Environment	Security Token URL
Production (SHA256)	https://softwareauthorisations.ato.gov.au/R3.0/S007v1.3/service.svc
EVTE (SHA256)	https://softwareauthorisations.acc.ato.gov.au/R3.0/S007v1.3/service.svc
Production (SHA1)	https://softwareauthorisations.ato.gov.au/R3.0/S007v1.2/service.svc

EVTE (SHA1) https://softwareauthorisations.acc.ato.gov.au/R3.0/S007v1.2/service.svc

3.1.1 Naming convention of Service Endpoint

https://softwareauthorisations.[<Environment>.]ato.gov.au/R<Intermediary>/<ServiceID>v<ContractNo>/Service.svc

Note: [<Environment>.] is empty for production.

Note: the contract version number forms part of the URL.

3.2 Service Consumption – General rules

3.2.1 Pre Conditions

1. The service request must be well formed and conform to MAS-ST's request schema for this contract.
2. The claims requested should conform to the list defined in the [CommonElements].
3. The relying party must register with MAS-ST before use is authorised to the service.
4. The Initiating party has been issued with a credential compatible with the myGovID verification infrastructure (machine credential).

3.2.2 Post Conditions

1. The service request will be recorded by MAS-ST for audit purposes (who, what, when, outcome).
2. A response will be provided to the Initiating Party indicating success or failure.
3. A successful response will contain an encrypted signed Security Token. Tokens are wrapped in a single SOAP response.
4. The SOAP response is protected using WS-Security.

3.2.3 Namespaces

Namespace	Specification(s)
http://schemas.xmlsoap.org/soap/envelope/	[SOAP]
http://www.w3.org/2003/05/soap-envelope	[SOAP12]
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	[WS-Security]
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	[WS-Security]
http://docs.oasis-open.org/wss/oasis-wss-wsecurity-secext-1.1.xsd	[WS-Security]
http://docs.oasis-open.org/ws-sx/ws-trust/200512	[WS-Trust]
http://www.w3.org/2000/09/xmldsig#	[XML-Signature]
http://www.w3.org/2001/04/xmlenc#	[XML-Encrypt]
http://schemas.xmlsoap.org/ws/2004/09/policy	[WS-Policy]
http://www.w3.org/2005/08/addressing	[WSAddressing]
http://www.w3.org/2001/XMLSchema	[XML-Schema1]

http://schemas.xmlsoap.org/ws/2005/05/identity http://vanguard.business.gov.au/2009/02 http://vanguard.business.gov.au/2016/03	[XML-Schema2] [Claims]

3.2.4 Service Response

The operations of this service will return an encrypted signed security token. The contents of the issued token will only be readable by the relying party.

A response will be formatted in accordance to [WS-Trust] RSTR contained in a SOAP envelope.

The response will hold a status. The status will advise of errors or constraints on token information. The response will be held within a SOAP body.

3.2.5 Service Operations

Summary of Operations supported by the S007 Security Token Service

Operation/s	Returns	Parameters Passed In	Parameters Passed Out
Issue	SOAP envelope whose body holds a Response containing an encoded Security token	A WS-Trust 1.3 RequestSecurityToken: • Token Type • Request Type • AppliesTo • KeyType • KeySize • Claims • Lifetime • ActAs	Issue a WS-Trust 1.3 RequestSecurityTokenResponse: • Token Type RequestType • AppliesTo • KeyType • KeySize • Lifetime • RequestedSecurityToken • RequestedAttachedReference • ProofToken • RequestedUnAttachedReference

3.2.6 Communication mechanism

The service request and response will be transported over the internet using HTTPS.

The request will be a SOAP message.

The service will be exposed as a W3C Web Service as a standard, neutral method of interaction between RP and MAS-ST.

The service will not be exposed in a UDDI or other registry and hence dynamic binding will not be possible.

The response will be signed with a MAS-ST private key.

Communication protocols

The following Web service standards will be used:

- Messages will be contained in a SOAP envelope
- Service messaging will be done using XML
- Web services descriptions will be expressed using WSDL

- An XML Request will be used to carry the request for a security token. The request will be placed in a SOAP body.
- An XML Response will be used to convey the returned security token. The response will be placed in a SOAP body.
- WS-Security standards (WS-Security) and SSL v3.0 / TLS will be used for security

3.2.7 Signing

WS-Security is employed in the SOAP message to protect the message at the application layer.

The returned security token will be signed with the MAS-ST private signing key.

Signing under this contract is summarised in the diagram below:

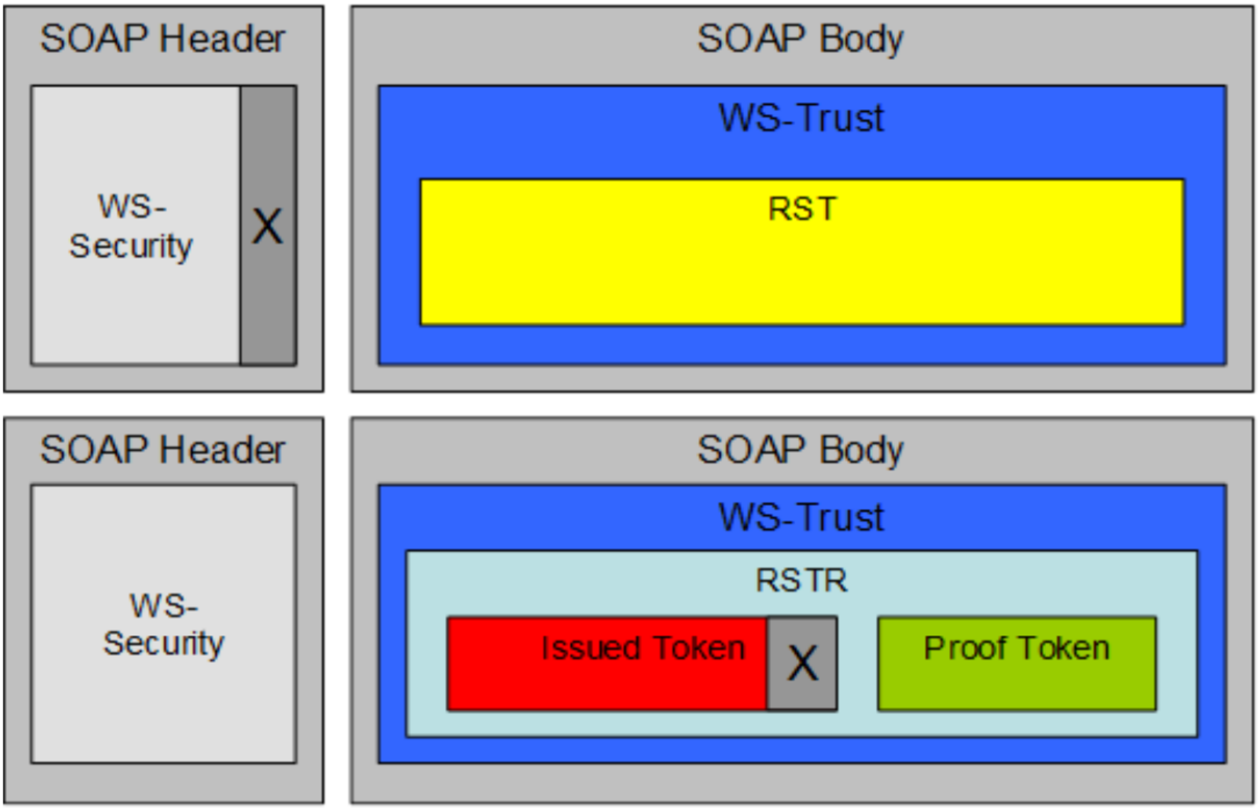


Figure 3: Signing

3.3 STS with delegation: Relationship Token

When a Relationship Token is included in the ActAs element of the request, the response will describe the Second Party defined in the Relationship Token, not the Initiating Party performing the request.

The response will also contain claims about the Initiating Party performing the request. These claims will be included as an encoded AttributeStatement, within the Actor claim as per an STSwD request.

Relationship Tokens can only be included in RSTs signed by a myGovId M2M machine credential.

3.3.1 RST ActAs element

The <v13:RelationshipToken> element is a way of specifying a Relationship which the initiating party wishes to ActAs.

The following section will describes the format of a Relationship Token.

The following table summarises the elements the Relationship Token may contain.

Element Name	Data type	Multiplicity	Mandatory (M) / Optional (O)
/v13:RelationshipToken	Complex	1	M
/v13:RelationshipToken/@wsu:Id	string	-	M
/v13:RelationshipToken/v13:Relationship	Complex	1	M
/v13:RelationshipToken/v13:Relationship/@v13:Type	URI	1	M
/v13:RelationshipToken/v13:Relationship/v13:Attribute	Complex	1	M
/v13:RelationshipToken/v13:Relationship/v13:Attribute@v13:Name	string	-	M
/v13:RelationshipToken/v13:Relationship/v13:Attribute@v13:Value	string	-	M
/v13:RelationshipToken/v13:FirstParty	Complex	1	M
/v13:RelationshipToken/v13:FirstParty/@v13:Scheme	URI	-	M
/v13:RelationshipToken/v13:FirstParty/@v13:Value	string	-	M
/v13:RelationshipToken/v13:SecondParty	Complex	1	M
/v13:RelationshipToken/v13:SecondParty/@v13:Scheme	URI	-	M
/v13:RelationshipToken/v13:SecondParty/@v13:Value	string	-	M

The following xml outlines the basic format of a Relationship Token, each section will be described.

```
<v13:RelationshipToken wsu:Id="..." xmlns:v13="...">
  <v13:Relationship v13:Type="...">
    <v13:Attribute v13:Name="..." v13:Value="..."/>
  </v13:Relationship>
  <v13:FirstParty v13:Scheme="..." v13:Value="..."/>
  <v13:SecondParty v13:Scheme="..." v13:Value="..."/>
</v13:RelationshipToken>
```

3.3.1.1 Element Name: RelationshipToken

Description: This element is used to represent a Relationship between two parties. This element is mandatory.

Element Location: /wst:RequestSecurityToken/wst:ActAs

Element Attributes:

Attribute Name	Type	Description	Mandatory(M)/ Optional (O)
wsu:Id	string	Specifies a globally unique identifier for the token.	M

Example:

```
<v13:RelationshipToken xmlns:v13="http://vanguard.business.gov.au/2016/03" Id="_3e4e371c-
b52c-4e32-9d8f-a06e64a90bd3">
  ...
</v13:RelationshipToken>
```

3.3.1.2 Element Name: Relationship

Description: This element is used to represent the relationship between a First Party and a Second Party. This element is mandatory.

Element Location: /wst:RequestSecurityToken/wst:ActAs/v13:RelationshipToken

Element Attributes:

Attribute Name	Type	Description	Mandatory(M)/ Optional (O)
V13:Type	string	This attribute specifies a string used to uniquely identify the type of relationship, for example “OSPfor”.	M

Accepted Values:

Currently the only accepted value for Type is:

- “OSPfor”

Example: The following example defines a relationship of the type OSPfor.

```
<v13:RelationshipToken xmlns:v13="…" ID="…">
  <v13:Relationship v13:Type="OSPfor">...</v13:Relationship>
</v13:RelationshipToken>
```

3.3.1.3 Element Name: Attribute

Description: This element represents a Relationship Attribute. This element is mandatory.

Element Location: /wst:RequestSecurityToken/wst:ActAs/v13:RelationshipToken/v13:Relationship

Element Attributes:

Attribute Name	Type	Description	Mandatory(M)/ Optional (O)

v13:Name	string	This required attribute specifies the unique name of the Relationship Attribute. For example this might be “SSID” to define the Software Subscription Identifier.	M
v13:Value	string	This required attribute specifies the value of the Relationship Attribute.	M

Accepted Values:

Currently the only accepted value for Name is:

- “SSID”

Example: The following example defines a relationship of the type OSPfor, with the Attribute “SSID” with its value set to “1234567890”.

```
<v13:RelationshipToken xmlns:v13="…" ID="…">
  <v13:Relationship v13:Type="OSPfor">
    <v13:Attribute v13:Name="SSID" v13:Value="1234567890"/>
  </v13:Relationship>
  ...
</v13:RelationshipToken>
```

3.3.1.4 Element Name: FirstParty

Description: This element is used to represent the First Party in this relationship, this must describe the identity that is claimed by the initiating party of the RST. This element is mandatory.

Element Location: /wst:RequestSecurityToken/wst:ActAs/v13:RelationshipToken

Element Attributes:

Attribute Name	Type	Description	Mandatory(M)/ Optional (O)
v13:Scheme	string	This attribute specifies a string which uniquely identifies both the issuer and type of the Identifier presenting the Value attribute.	M
v13:Value	string	This attribute specifies a string which contains the value unique identifier for the First Party.	M

Accepted Values:

Currently the only accepted value for Scheme is:

- “uri://abr.gov.au/ABN”

The ABN in the value attribute will be validated against the initiating party token and must be the same as the ABN embedded within the Machine credential used to sign the RST.

Example: The following example defines a relationship with the First Party identified by their ABN.

```
<v13:RelationshipToken xmlns:v13="…" ID="…">
  ...
  <v13:FirstParty v13:Scheme=" uri://abr.gov.au/ABN " v13:Value="
51824753556"/>
  ...
</v13:RelationshipToken>
```

3.3.1.5 Element Name: SecondParty

Description: This required element represents the Second Party in this relationship; this is the identity which the initiating party wishes to act as. This element is mandatory.

Element Location:
/wst:RequestSecurityToken/wst:ActAs/v13:RelationshipToken/v13:Relationship

Element Attributes:

Attribute Name	Type	Description	Mandatory(M)/ Optional (O)
v13:Scheme	string	This attribute specifies a string which uniquely identifies both the issuer (or related issuers) of the Second Party identifier and its type.	M
v13:Value	string	This attribute specifies a string which contains the value unique identifier for the Second Party.	M

Accepted Values:

Currently the only accepted value for Scheme is:

- “uri://abr.gov.au/ABN”

The ABN in the value attribute will be validated against the initiating party token and must be the same as the ABN embedded within the Machine credential used to sign the RST.

Example: The following example defines a relationship with the Second Party identified by their ABN.

```
<v13:RelationshipToken xmlns:v13="…" ID="…">
  ...
  <v13:SecondParty v13:Scheme="uri://abr.gov.au/ABN" v13:Value="
51824753556"/>
  ...
</v13:RelationshipToken>
```

4 Request and Response protocols

This contract is implemented as a Web Service. As such it receives and returns SOAP messages.

4.1 Service Operation: Issue()

The Issue operation accepts a WS-Trust request for the issuance of a security token for use by the initiating party to access relying party services. The request is signed by the initiating party's private key.

4.1.1 Request Validation

An Issue operation is composed of a number of XML elements with RequestSecurityToken being the top level element.

A security token will be returned if the following validation has been applied successfully:

- The relying party is a known to MAS-ST Identified by the value of AppliesTo endpoint.
- The Initiating Party has submitted a valid request specifying mandatory fields.
- The credential passed to the STS is verifiable by MAS-ST.
- The mandatory claims requested are known to MAS-ST.
- The certificate used in the request has not been revoked by the CA as of the last CRL update.
- The certificate used in the request has not expired.

Under certain conditions a response is not possible due to the following conditions

- The request does not adhere to this Technical Service Contract:

A SOAP fault will be generated under any of the following:

- MAS-ST cannot validate the certificate.
- The relying party cannot be identified.
- The certificate type presents is not supported by MAS-ST.
- MAS-ST cannot make the assertion according to the requested claims.
- The request does not adhere to this Technical Service Contract:
    - The total size of the request exceeds the maximum allowable size (greater than 100KB).
    - The request contains a request for more than one security token.
    - The request is missing a mandatory element.
    - An element within the request contains invalid data.
- A technical error occurred.

The response contains attributes that give a reason code and a reason text.

Some errors will result in a SOAP fault. Under best practice for Internet applications, errors are not explained in the response message but are logged for later use by support staff (see below).

Note: all date or time values are expressed as Universal Coordinated Time (UTC) according to [RFC 3339] with a zero offset. For example, 2006-12-20T13:39:57Z represents 39 minutes and 57 seconds after 1 pm on December 20th, 2006 in UTC.

4.1.2 Request Security Token (RST)

The following section will describe the format of the input message known as the Request Security Token (RST) .

The following table summarises the elements the RST may be contain the data type and whether they are mandatory or optional for each element.

Element Name	Data type	Mandatory (M) / Optional (O)
wst:/RequestSecurityToken	Complex	M
wst:/RequestSecurityToken/wst:TokenType	URI	O
wst:/RequestSecurityToken/wst:RequestType	URI	M
wst:/RequestSecurityToken/wsp:AppliesTo	[WS-Addressing] Endpoint	M
wst:/RequestSecurityToken/wst:KeyType	URI	O
wst:/RequestSecurityToken/wst:KeySize	Integer	O
wst:/RequestSecurityToken/wst:Claims	Complex	O
wst:/RequestSecurityToken/wst:Lifetime	Complex	O
wst:/RequestSecurityToken/wst:Lifetime/wsu:Created	xs:datetime	O
wst:/RequestSecurityToken/wst:Lifetime/wsu:Expires	xs:datetime	O
Wst:/RequestSecurityToken/wst:ActAs	Complex	O

The following xml outlines the basic format of a RST Input request each section will be described

```
<wst:RequestSecurityToken xmlns:wst="...">
  <wst:TokenType>...</wst:TokenType>
  <wst:RequestType>...</wst:RequestType>
  ...
  <wsp:AppliesTo>...</wsp:AppliesTo>
  <wst:Claims Dialect="...">...</wst:Claims>
  <wst:Entropy>
    <wst:BinarySecret>...</wst:BinarySecret>
  </wst:Entropy>
  <wst:Lifetime>
    <wsu:Created>...</wsu:Created>
    <wsu:Expires>...</wsu:Expires>
  </wst:Lifetime>
  <wst:ActAs>...</wst:ActAs>
  ...
</wst:RequestSecurityToken>
```

4.1.2.1 Element Name: RequestSecurityToken

Description: This element is used to request a security token. This element is mandatory

Element Location: SOAP Body

Element Attributes:

Attribute Name	Type	Description	Mandatory(M) / Optional (O)
Context	URI	Specifies an identifier for the request, All Responses to the request will echo the value in the response.	O

Validation Rules:

The context value will be limited to a maximum of 512 characters. A defined safe set of characters will be defined.

No other processing will be done on the content of the context attribute data this will be simply echoed in RSTR.

The RequestSecurityToken element must be signed by the requestor using a token contained/referenced in the request.

Error Conditions: A violation of the validation rule will result in a SOAP fault being generated.

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.

Example: The following example requests the optional attribute context to be echoed in the response.

```
<wst:RequestSecurityToken xmlns:wst="http://docs.oasis-open.org/wssx/ws-trust/200512"
Context="http://RelyingParty.gov.au/RST/UniqueRequestID">
  ...
</wst:RequestSecurityToken>
```

4.1.2.2 Element Name: TokenType

Description: This element is used to request the type of security token to be returned in the response.

Element Location: /wst:RequestSecurityToken/wst:TokenType

Element Attributes: None

Validation Rules:

If specified this value must be either the SAML1.1 or SAML2.0 qualifiers, refer to accepted values for this element. If specified and not matched a soap fault will be generated.

If this element is not specified SAML 2.0 will be the default.

Accepted Values:

To return a SAML 1.1 token

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

To return a SAML 2.0 token

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0

A SOAP fault must be returned if one of these values is not specified

Error Conditions: A SOAP fault will be generated if the URI is not matched to one in the validation rules

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.

Example:

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
  ...
</wst:RequestSecurityToken>
```

4.1.2.3 Element Name: RequestType

Description: This element is used to request a security token.

Element Location: /wst:RequestSecurityToken/wst:RequestType

Element Attributes: None

Validation Rules:

This value must be “http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue“. A soap fault must be returned if this is not the value.

This element is mandatory a soap fault wst:InvalidRequest will be generated if this element is missing.

Error Conditions: A SOAP fault will be generated if the URI is not that which is listed in the validation rules.

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.

Example:

```
<wst:RequestSecurityToken xmlns:wst="...">
```

```
  ...
  <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
  ...
</wst:RequestSecurityToken>
```

4.1.2.4 Element Name: AppliesTo

Description: This mandatory element is used to specify the relying party that the initiating party, requires a security token for. The AppliesTo element is specified as a WS-Addressing Endpoint.

The response session key is encrypted with the public key of the relying party associated with the Endpoint.

Namespace: Endpoint reference as defined in [WS-Addressing].

http://www.w3.org/2005/08/addressing

Element Location: /wst:RequestSecurityToken/wsp:AppliesTo

Element Attributes: None

Validation Rules:

This value will be pattern matched against a list of known relying parties endpoints. The request from initiating parties with an invalid request or missing value will result in a SOAP Fault being generated.

The Endpoint reference must have only one address element.

Error Conditions:

A wst:RequestFailed SOAP fault will be generated if the end point is unknown.

A wst:RequestFailed SOAP fault will be generated if the AppliesTo endpoint is malformed.

A MissingAppliesTo SOAP fault will be generated if the AppliesTo element is not present.

Fault Code	**Fault String**
wst:RequestFailed	The specified request failed
MissingAppliesTo	The AppliesTo mandatory field has not been supplied.

Example: The following example requests the issuance of a security token, specifying the relying party Endpoint that will consume the token.

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
      <Address>http://RelyingParty.gov.au/servicename/ServiceA</Address>
    </EndpointReference>
  </wst:AppliesTo>
  ...
```

```
</wst:RequestSecurityToken>
```

Note: WS-Addressing specification allows an identity element to be specified this may cause an error.

4.1.2.5 Element Name: KeyType

Description: This element is used to specify the type of key required to prove possession, known as the proof key. The proof key is placed in the proof token and the issued token. The initiating party proves possession to the relying party by accessing the proof key and signing a service request with the proof key.

Specifying symmetric key instructs the STS to generate a symmetric key and include it in the issued token and proof token.

Element Location: /wst:RequestSecurityToken/wst:KeyType

Element Attributes: None

Validation Rules:

Refer to accepted values for this element.

Accepted Values:

To use a symmetric key

http://docs.oasis-open.org/ws-sx/ws-trust/200512/SymmetricKey

Error Conditions: A wst:RequestFailed SOAP fault must be returned if one of these values are not specified and the element is present in the request.

Fault Code	Fault String
wst: RequestFailed	The specified request failed

Example: The following example requests specify the use of a symmetric key.

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/SymmetricKey</wst:KeyType>
  ...
</wst:RequestSecurityToken>
```

4.1.2.6 Element Name: KeySize

Description: This element is used to indicate the size of the key required. This value is specified as the number of bits. This value is used to indicate the strength of the security requested.

Element Location: /wst:RequestSecurityToken/wst:KeySize

Element Attributes: None

Validation Rules:

This element must be an integer, if specified. Refer to accepted values for this element. The default size is dependent on the key type chosen. The default key size if not specified is 256 for symmetric key type.

Accepted Values:

An integer value must be specified.

A SOAP fault will be generated if the value is non integer.

Error Conditions: A wst: InvalidRequest SOAP fault must be returned if the value is non integer.

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.

Example: The following example requests a 512-bit key.

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:KeySize>512</wst:KeySize>
  ...
</wst:RequestSecurityToken>
```

4.1.2.7 Element Name: Claims

Description: This element is used to request specific claims returned in the response token

Element Location: /wst:RequestSecurityToken/wst:Claims

Element Attributes:

Attribute Name	Type	Description	Mandatory(M) / Optional (O)
Dialect	URI	Specifies a uri to indicate the syntax of the claims in the containing element.	M

Validation Rules:

If the claims element is specified then the dialect must be specified and this value must correspond to the value published in the [Common Elements]
Dialect=“http://schemas.xmlsoap.org/ws/2005/05/identity”

Common Elements outlines the claim names that must be specified in the request.

At least one child (claim type) element must be present. A wst:InvalidRequest SOAP fault will be generated if no child elements are found.

If the claims element is omitted, a default set of claims are returned based on the initiating party certificate type present in the request.

Accepted Values:

The claims element must be well formed and the child elements must be a claim type element. Refer to common elements document.

A SOAP fault will be returned under the following scenarios :

1. If the claim type is unknown, that is, the claim name specified is not recognised by the service or not compliant with the naming schema specified in [Common Elements] a SOAP fault wst:InvalidRequest will be generated only if the claim is required. Unknown Optional claims will not trigger a SOAP fault.
2. The claim type cannot be compiled for the certificate used to sign the request (IP certificate).

Error Conditions:

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.
wst:BadRequest	The request token is not understood.

Example: The following example requests the australianbusinessnumber to be returned in response. Refer to [Common Elements] for a list of claim URI's

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:Claims Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"
xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity">
    ...
    <i:ClaimType Uri="http://vanguard.business.gov.au/2009/03/australianbusinessnumber"/>
    ...
  </wst:Claims>
  ...
</wst:RequestSecurityToken>
```

4.1.2.8 *Element Name: ClaimType*

Description: This element is used to request specific claims returned in the response token.

Element Location: /wst:RequestSecurityToken/wst:Claims

Element Attributes:

Attribute Name	Type	Description	Mandatory(M) / Optional (O)
URI	URI	Specifies a uri to indicate the syntax of the claims in the containing element.	M
Optional	Boolean	This attribute specifies if the claim is required in the response. If a claim has been marked as mandatory and cannot be fulfilled from the claims store, a SOAP fault must be returned. If this attribute is omitted it defaults to	O

mandatory “false”

Validation Rules:

In the Claim Type element the URI attribute must be specified. This value must correspond to the names published in [Common Elements].

Table 1 in [Common Elements] outlines the claim names format for both saml 1.1 and saml 2.0 that are acceptable in the request.

If a Claim is specified as not optional “optional= false” and a value cannot be evaluated and returned, MAS-ST will not issue a security token.

Accepted Values:

Each ClaimType element must be well formed and specified with the Claims parent element.

A wst:InvalidRequest SOAP fault will be returned if the claim type is not well formed.

A wst:InvalidRequest SOAP fault will be returned if the claim type URI attribute is not specified.

Error Conditions:

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.
wst:BadRequest	The request token is not understood.

Example:

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:Claims xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity/claims">
    <i:ClaimType
Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/australianbusinessnumber"/>
    <i:ClaimType Uri=" http://vanguard.ebusiness.gov.au/2008/06/identity/claims/businessname"
Optional="true"/>
    ...
  </wst:Claims>
  ...
</wst:RequestSecurityToken>
```

4.1.2.9 Element Name: Lifetime

Description: This element is used to request a variation to the validity period of a security token

Element Location: /wst:RequestSecurityToken/wst:Lifetime

Element Attributes: None

Validation Rules:

If specified the value must contain a valid expires child node.

If not specified the system will generate a token valid for thirty minutes.

Accepted Values:

The lifetime element has two child nodes as specified by the [WS-Security] specification. Acceptable child nodes for this element are Created and Expires.

Error Conditions:

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.
wst:BadRequest	The request token is not understood.

Example: The following example requests a token that expires at 4 February 2009

```
<wst:RequestSecurityToken xmlns:wst="...">
...
<wst:Lifetime>
<wsu:Expires>2009-02-04T05:26:29.314Z</wsu:Expires>
</wst:Lifetime>
...
</wst:RequestSecurityToken>
```

4.1.2.10 Element Name: Expires

Description: This element is used to specify the upper bound of the validity time period for the requested security token.

Element Location: /wst:RequestSecurityToken/wst:Lifetime/wsu:Expires

Element Attributes: None

Validation Rules:

The value must be a valid date and time specified in UTC format with a zero offset, if an invalid date is passed a SOAP fault wst:InvalidRequest will be generated. If milliseconds are supplied, the precision must not be greater than 3.

MAS-ST implements a minimum and maximum bound for requests measured using MAS-ST clocks. The difference between the MAS-ST clock and the expires datetime must be less than or equal to the maximum eight hours for the upper bound. The minimum upper bound period is five minutes. eg. A request for a security token to expire within five minutes from the request being validated will result in a SOAP fault being generated.

This element must be specified as a direct child of the Lifetime element to be valid.

Accepted Values:

The lifetime element has two child nodes as specified by the [WS-Security] specification acceptable child nodes for this element are created and expires. The created date in the request will be ignored by the STS and MAS-ST clocks will be used to specify the creation date. MAS-ST does not support post dated requests and as such ignore created date times.

Error Conditions:

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.
wst:BadRequest	The request token is not understood.

Example: The following example shows a caller specifying an expiry time.

```
<wst:RequestSecurityToken xmlns:wst="...">
  ...
  <wst:Lifetime>
    <wsu:Expires>2009-02-04T05:26:29.314Z</wsu:Expires>
  </wst:Lifetime>
  ...
</wst:RequestSecurityToken>
```

4.1.2.11 *Element Name: ActAs*

Description: This element indicates that the requested token is expected to contain information about the identity represented by the content of the element and the token requestor intends to use the returned token to act as this identity.

For further detail of the functionality of the ActAs element and Actor attribute, including examples, see sections 3.3 and 3.4, above.

Element Location: /wst:RequestSecurityToken/wst:ActAs

Element Attributes: None

Validation Rules:

The ActAs element must not be empty.

Accepted Values: See sections 3.3 and 3.4, above.

Error Conditions:

Fault Code	Fault String
wst: InvalidRequest	The request was invalid or malformed.
wst:BadRequest	The request token is not understood.

Example: See sections 3.3 and 3.4, above.

4.1.3 Request Security Token Response (RSTR)

The following table summaries if the elements in the request are mandatory or optional. Each element is described.

Abbreviations:

RSTRC = RequestSecurityTokenResponseCollection

RSTR = RequestedSecurityTokenResponse

Element Name	**Data type**	**Mandatory (M) / Optional (O)**
wst:/RSTRC	Complex	M
wst:/RSTRC/wst:/RSTR/wst:TokenType	URI	M
wst:/RSTRC/wst:/RSTR/wst:RequestType	URI	M
wst:/RSTRC/wst:/RSTR/wsp:AppliesTo	[WS-Addressing] Endpoint	M
wst:/RSTRC/wst:/RSTR/wst:KeyType	URI	M
wst:/RSTRC/wst:/RSTR/wst:KeySize I	nteger	M
wst:/RSTRC/wst:/RSTR/wst:Lifetime	Complex	M
wst:/RSTRC/wst:/RSTR/wst:Lifetime/wsu:Created	xs:datetime	M
wst:/RSTRC/wst:/RSTR/wst:Lifetime/wsu:Expires	xs:datetime	M
wst:/RSTRC/wst:/RSTR/wst:RequestedSecurityToken/EncryptedAssertion	Complex	M
wst:/RSTRC/wst:/RSTR/wst:RequestedAttachedReference	Complex	M
wst:/RSTRC/wst:/RSTR/wst:RequestedProofToken	Complex	M
wst:/RSTRC/wst:/RSTR/wst:RequestedProofToken/wst:BinarySecret	Base64 encoded sequence	M
wst:/RSTRC/wst:/RSTR/wst:RequestedUnAttachedReference	Complex	M

The following xml outlines the basic format of a RSTR Output Response each section will be described below.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="http://docs.oasisopen.org/ws-sx/ws-wst/200512">
  <wst:RequestSecurityTokenResponse>
    <wst:RequestedSecurityToken>
      ...
      <wst:TokenType>...</wst:TokenType>
      <wst:RequestType>...</wst:RequestType>
      <wsp:AppliesTo xmlns:wsp="...">...</wsp:AppliesTo>
      <wst:KeyType>...</wst:KeyType>
      <wst:KeySize>...</wst:KeySize>
      <wst:Lifetime>...</wst:Lifetime>
      <wst:RequestedAttachedReference>...</wst:RequestedAttachedReference>
      <wst:RequestedUnattachedReference>...</wst:RequestedUnattachedReference>
      <wst:RequestedProofToken>...</wst:RequestedProofToken>
      <wst:Entropy>
        <wst:BinarySecret>...</wst:BinarySecret>
      </wst:Entropy>
      ...
    </wst:RequestedSecurityToken>
    ...
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.1 Element Name: RequestSecurityTokenResponseCollection

Description: This element can be used to return one or more security tokens. This element is mandatory for any token response. This element will only contain one security token.

Element Location: SOAP Body

Element Attributes: None

Validation Rules: None

Example: The following example response shows the response collection element contained in the soap body.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-
wst/200512">
    ...
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.2 Element Name: RequestSecurityTokenResponse

Description: This element is used to return the security token. This element is mandatory for any token response

Element Location: wst:/RequestSecurityTokenResponseCollection

Element Attributes:

Attribute Name	Type	Description	Mandatory(M) / Optional (O)
Context	URI	Specifies an identifier for the request, This value is copied from the request into the response Note: This attribute is echoed from the request if specified. This field can be used by initiating party applications to link requests with responses.	O

Validation Rules: None

Example: The following example response echoes the context attribute passed in the RST request.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  ...
  <wst:RequestSecurityTokenResponse
Context="http://RelyingParty.gov.au/RST/UnqiueRequestID">...</wst:RequestSecurityTokenRespo
```

```
nse>
  ...
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.3 Element Name: TokenType

Description: This element is used to identify the format type of security token returned in the response.

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst:TokenType

Element Attributes: None

Validation Rules: None

Returned Values:

The returned value is SAML 1.1 if specified in the request

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

The returned value is SAML 2.0 if specified or not specified in the token request

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0

Example: The following example response is a SAML 2.0 token returned in response.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
    ...
    <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
    ...
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.4 Element Name: AppliesTo

Description: This element is used to indicate the relying party that the initiating party is requesting the security token for. This will be the endpoint address specified in the request.

This element must be in the response and is echoed from the corresponding RST.

Namespace: Uses WS-Addressing to specify an endpoint reference
http://www.w3.org/2005/08/addressing

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wsp:AppliesTo

Element Attributes: None

Validation Rules: Echoed directly from the RST request.

Example: The following example is a response to the issuance of a security token specifying the relying party's endpoint that will consume the token.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse xmlns:wst="...">
    ...
    <wst:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
        <Address>http://RelyingParty.gov.au/servicename/Service.svc</Address>
      </EndpointReference>
    </wst:AppliesTo>
    ...
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.5 Element Name: KeyType

Description: This element is used to indicate keytype used for protecting the proof key in the response. This element must be in the response, and should match the corresponding element (or default value) in the RST

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wsp:
KeyType

Element Attributes: None

Validation Rules: None

Example: The following example is a response to a RST request where the key type was not specified in the request. The default behaviour is to use a symmetric key.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
    ...
    <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/SymmetricKey</wst:KeyType>
    ...
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.6 Element Name: KeySize

Description: This element is used to indicate size of the key used in the response in bits

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wsp: KeySize

Element Attributes: None

Validation Rules: None

Example: The following example response to the issuance of a security token indicating a 512 bit key is being used

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
    ...
    <wst:KeySize>512</wst:KeySize>
    ...
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.7 Element Name: Lifetime

Description: This element is used to indicate the period of validity for the returned security token

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst: Lifetime

Element Attributes: None

Validation Rules: None

Accepted Values:

The lifetime element has two child nodes as specified by the [WS-Security] specification child nodes for this element are created and expires. The values of the elements will be date time in UTC format with a zero offset.

The created element defines when the STS issued the token. The expires element defines the upper bound of the validity of the token. A RP should reject tokens that have expired.

Example: The following example response indicates the lifetime of the security token.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityToken>
    ...
    <wst:Lifetime>
      <wsu:Created>2009-02-04T05:21:29.314Z</wsu:Created>
      <wsu:Expires>2009-02-04T05:26:29.314Z</wsu:Expires>
    </wst:Lifetime>
    ...
  </wst:RequestSecurityToken>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.8 Element Name: RequestedSecurityToken

Description: This element contains the SAML assertion, this element will only contain one security token. The definition of this element is defined in the common elements [Common Elements] document.

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst: RequestedSecurityToken

Element Attributes: None

Validation Rules:

1. It is the responsibility of the relying party to verify the claims in the token are sufficient
2. Verify the attributes of the claimant are proven by the signatures
3. Verify the issuer of the security token is trusted

Example: The following example response to the issuance of a security token, specifies the requested SAML assertion.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
    <wst:RequestedSecurityToken>
      <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
        <Assertion>...</Assertion>
      </EncryptedAssertion>
    </wst:RequestedSecurityToken>
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.9 Element Name: RequestedAttachedReference

Description: This element is used to reference the returned security token in the response message header. This element contains a WS-Security Security token reference and is used to reference the token placed inside the SOAP message header.

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst: RequestedSecurityToken/wst:RequestedAttachedReference

Element Attributes: None

Validation Rules: None

Example: The following example response to the issuance of a security token specifies a key reference is attached in the message and is identified using the key identifier

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
```

```
    <wst:RequestedSecurityToken>
      <wst:RequestedAttachedReference>
        <SecurityTokenReference a:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
token-profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:a="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-
1.1.xsd">
          <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLID">_91b7228b-4fba- 4aac-b476-2079888d3929</KeyIdentifier>
        </SecurityTokenReference>
      </wst:RequestedAttachedReference>
    </wst:RequestedSecurityToken>
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.10 Element Name: RequestedProofToken

Description: This element is used to return the proof of possession token associated with the requested security token. The contents of the element are generated by the STS based on the key type specified in the request.

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst:
RequestedProofToken

Element Attributes: None

Validation Rules: None

Example: The following is an example response to the issuance of a security token specifying the proof token as a result of a symmetric key type request

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
    <wst:RequestedSecurityToken>
      <wst:RequestedProofToken>

<wst:BinarySecret>Fr2YXrFAZ+1/ESGQN0+j/w9C5kQve2KmVr/SfbCth60=</wst:BinarySecret>
      </wst:RequestedProofToken>
    </wst:RequestedSecurityToken>
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.3.11 Element Name: RequestedUnAttachedReference

Description: This element is used to identify key material required to make use of the RSTR, this element is used to identify key material not in the message. This element format is a WS-Security Security token reference.

Element Location:
/wst:RequestSecurityTokenResponseCollection/wst:RequestSecurityTokenResponse/wst:
RequestedUnAttachedReference

Element Attributes: None

Validation Rules: None

Example: The following example response to the issuance of a security token, specifies how to reference a key that is not contained in the message using the key identifier.

```
<wst:RequestSecurityTokenResponseCollection xmlns:wst="...">
  <wst:RequestSecurityTokenResponse>
    <wst:RequestedSecurityToken>
      <wst:RequestedUnattachedReference>
        <SecurityTokenReference a:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
token-profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:a="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-
1.1.xsd">
          <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLID">_91b7228b-4fba- 4aac-b476-2079888d3929</KeyIdentifier>
        </SecurityTokenReference>
      </wst:RequestedUnattachedReference>
    </wst:RequestedSecurityToken>
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

Example: The following example is a complete response to the issue request for a security token.

Note: The content of the EncryptedAssertion element below is left unencrypted for readability purposes. In a live system this element would actually contain an EncryptedData element as specified by [XML-Encrypt].

```
<wst:RequestSecurityTokenResponseCollection xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">
  <wst:RequestSecurityTokenResponse>
    <wst:KeySize>256</wst:KeySize>
    <wst:Lifetime>
      <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-
1.0.xsd">2009-02-04T05:20:15.000Z</wsu:Created>
      <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-
1.0.xsd">2009-02-04T15:20:15.000Z</wsu:Expires>
    </wst:Lifetime>
    <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
        <Address>http://RelyingPartyService/Service</Address>
      </EndpointReference>
    </wsp:AppliesTo>
  <wst:RequestedSecurityToken>
   <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
    <Assertion ID="_91b7228b-4fba-4aac-b476-2079888d3929" IssueInstant="2009-02-
04T05:21:29.277Z" Version="2.0" xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
     <Issuer>MAS-ST Security Token Service</Issuer>
     <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#"/>
       <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
       <ds:Reference URI="#_91b7228b-4fba-4aac-b476-2079888d3929">
        <ds:Transforms>
         <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
         <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>rXwz1qJFjpf0zVDtkUfTewYtZ5c=</ds:DigestValue>
       </ds:Reference>
      </ds:SignedInfo>

<ds:SignatureValue>PcwYYR7hxm/33yZNv......+6dXvzE9QIkVJwn7AIS4o=</ds:SignatureValue>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
       <X509Data>

<X509Certificate>MIIFbjCCBFagAwIB......ZzE9QzE9QzE9QzE9QzE9QzE9QMjI=</X509Certificat
e>
       </X509Data>
      </KeyInfo>
     </ds:Signature>
     <Subject>
      <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
       <SubjectConfirmationData a:type="KeyInfoConfirmationDataType"
xmlns:a="http://www.w3.org/2001/XMLSchema-instance">
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
         <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
          <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgf1p">
           <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          </e:EncryptionMethod>
          <KeyInfo>
           <o:SecurityTokenReference xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
            <X509Data>
             <X509IssuerSerial>
              <X509IssuerName>CN=Aust Govt OCA, OU=test, OU=Aust Services, O=Aust
Govt, C=AU</X509IssuerName>

<X509SerialNumber>89781976347204219971241566881892722798</X509SerialNumber>
             </X509IssuerSerial>
            </X509Data>
           </o:SecurityTokenReference>
```

```
                    </KeyInfo>
                    <e:CipherData>
                      <e:CipherValue>BW7hjWajXc1STVs......Gx8Q8qE=</e:CipherValue>
                    </e:CipherData>
                  </e:EncryptedKey>
                </KeyInfo>
              </SubjectConfirmationData>
            </SubjectConfirmation>
          </Subject>
          <Conditions NotBefore="2009-02-04T05:20:15.000Z" NotOnOrAfter="2009-02-
04T15:20:15.000Z">
            <AudienceRestriction>
              <Audience>http://RelyingParty.gov.au/RelyingPartyService13</Audience>
            </AudienceRestriction>
          </Conditions>
          <AttributeStatement>
            <Attribute Name="http://VANguard.business.gov.au/2009/03/australianbusinessnumber">
              <AttributeValue a:type="tn:string" xmlns:a="http://www.w3.org/2001/XMLSchema-
instance" xmlns:tn="http://www.w3.org/2001/XMLSchema">TESTABN123</AttributeValue>
            </Attribute>
          </AttributeStatement>
        </Assertion>
      </EncryptedAssertion>
    </wst:RequestedSecurityToken>
    <wst:RequestedProofToken>

<wst:BinarySecret>Fr2YXrFAZ+1/ESGQN0+j/w9C5kQve2KmVr/SfbCth60=</wst:BinarySecret>
    </wst:RequestedProofToken>
    <wst:RequestedAttachedReference>
      <SecurityTokenReference a:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wsswssecurity-secext-1.0.xsd" xmlns:a="http://docs.oasis-open.org/wss/oasis-wss-
wssecuritysecext-1.1.xsd">
        <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wsssaml-token-profile-
1.1#SAMLID">_91b7228b-4fba-4aac-b476-2079888d3929</KeyIdentifier>
      </SecurityTokenReference>
    </wst:RequestedAttachedReference>
    <wst:RequestedUnattachedReference>
      <SecurityTokenReference a:TokenType="http://docs.oasis-open.org/wss/oasis-wss-samltoken-
profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wsswssecurity-secext-1.0.xsd" xmlns:a="http://docs.oasis-open.org/wss/oasis-wss-
wssecuritysecext-1.1.xsd">
        <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wsssaml-token-profile-
1.1#SAMLID">_91b7228b-4fba-4aac-b476-2079888d3929</KeyIdentifier>
      </SecurityTokenReference>
    </wst:RequestedUnattachedReference>
    <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-tokenprofile-
1.1#SAMLV2.0</wst:TokenType>
    <wst:RequestType>http://docs.oasis-open.org/ws-sx/wstrust/200512/Issue</wst:RequestType>
    <wst:KeyType>http://docs.oasis-open.org/ws-sx/wstrust/200512/SymmetricKey</wst:KeyType>
```

```
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.4 Exceptions/Faults

Errors use SOAP 1.2 fault mechanisms. The following table summarises the soap faults that may be returned from this service.

4.1.4.1 Element Name:Fault

Description: This element is used to return a fault as described in the SOAP 1.2 specification Section 5.4 of the W3C recommendation.

The structure of the soap fault is code, reason and detail. The only elements that require description are covered in the following element definitions.

Namespace: http://www.w3.org/2003/05/soap-envelope

Element Location: Envelope/Body/Fault

Element Attributes: None

Validation Rules: The following table details the errors returned as SOAP faults when an exception occurs whilst processing a request.

4.1.4.2 Element Name:Subcode

Description: The subcode element contains the value that maps to the error table below. This element is defined in the SOAP 1.2 specification Section 5.4 of the W3C recommendation.

The subcode is not included in all faults. For further details see the table below.

Namespace: http://www.w3.org/2003/05/soap-envelope

Element Location: Envelope/Body/Fault/Code/SubCode

Element Attributes: None

4.1.4.3 Element Name:Detail

Description: The detail element contains a structured error object. For this release of the service, this object will always be an instance of STSFault as described below.

The detail element is not included in all faults. It is included only where additional information is needed to further describe the event.

Namespace: http://www.w3.org/2003/05/soap-envelope

Element Location: Envelope/Body/Fault/Detail

Element Attributes: None

4.1.4.4 Element Name:STSFault

Description: The STSFault element provides structured detail regarding the nature of the fault. It reproduces the information contained within the Reason/Text field in a typed format.

Namespace: http://vanguard.business.gov.au/2009/02

Element Location: Envelope/Body/Fault/Detail/STSFault

Element Attributes: None

4.1.4.5 Element Name:EventCode

Description: This element contains the MAS-ST specific error code. This is useful for diagnostic and debugging purposes. Specific event codes are described in the table below.

Namespace: http://vanguard.business.gov.au/2009/02

Element Location: Envelope/Body/Fault/Detail/STSFault /EventCode

Element Attributes: None

4.1.4.6 Element Name:EventSeverity

Description: This element describes the event severity. The value will be one of Normal, Warning, Severe or Critical. This element is used for diagnostic and debugging purposes.

Namespace: http://vanguard.business.gov.au/2009/02

Element Location: Envelope/Body/Fault/Detail/STSFault/EventSeverity

Element Attributes: None

4.1.4.7 Element Name:EventDescription

Description: This element provides a verbose, human readable description of the fault. This element is used for diagnostic and debugging purposes.

Namespace: http://vanguard.business.gov.au/2009/02

Element Location: Envelope/Body/Fault/Detail/STSFault/EventDescription

Element Attributes: None

4.1.4.8 Element Name:UserAdvice

Description: This element provides advice targeted at a non-technical user. It may assist in resolving the conditions that produced the fault.

Namespace: http://vanguard.business.gov.au/2009/02

Element Location: Envelope/Body/Fault/Detail/STSFault/UserAdvice

Element Attributes: None

4.2 Error Codes

The following table outlines the errors that are returned.

Note: Draft. These codes have not been finalised.

4.2.1 Sender Error Codes

Fault Code: env:Sender

Sub code	MAS-ST Sub Code	Description
wst:RequestFailed	v:E2001	The token type specified in the request was not recognised. Only SAML2.0 tokens should be requested.
wst:RequestFailed	v:E2003	The relying party specified in the AppliesTo element is not recognized.
wsse:FailedAuthentication	v:E2014	The credential supplied by the initiating party has been revoked.
wsse:FailedAuthentication	v:E2015	The credential supplied by the initiating party has expired.
wsse:FailedAuthentication	v:E2017	The validity start date of the credential supplied by the initiating party is in the future.
wsse:FailedAuthentication	v:E2020	The Credential Authority that issued the credential supplied by the initiating party is not recognized.
wsse:FailedAuthentication	v:E2029	The credential supplied by the initiating party could not be processed and may be corrupt.
wsse:FailedAuthentication	v:E2169	The credential supplied by the initiating party is not recognized.
wsse:FailedAuthentication	v:E2180	No usage policy for the credential supplied could be found. This would occur if a certificate that was valid but not supported by the STS was presented.
wst:RequestFailed	v:E2182	A mandatory claim specified in the request could not be provided. Check the claim types being specified in the request.
InvalidRequest	v:E2183	A mandatory request was made for an unrecognised claim.
wst:RequestFailed	E9000	An unknown request type was encountered in the message. Typically the request type should be: http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue
MissingAppliesTo	E9001	The AppliesTo element of the RST was not supplied. This element must be supplied in any request to the STS.
UnsupportedSecurityToken	E9002	An unsupported token was provided.
UnsupportedAlgorithm	E9003	An unsupported signature or encryption algorithm was used.
InvalidSecurity	E9004	An error was discovered processing the wsse:Security header.
InvalidSecurityToken	E9005	An invalid security token was provided.
wsse:FailedAuthentication	E9006	The security token could not be authenticated or authorized.

4.2.2 Receiver Error Codes

Fault Code: env:Receiver

Sub code	MAS-ST	Description

	Sub Code	
N/A	v:E1001	The request could not be satisfied due to an internal error.
N/A	v:E2190	Claim data could not be found due to an internal error. Attempt the request again..

4.2.3 Examples

Example 1: Failed Authentication

```
<Fault xmlns="http://www.w3.org/2003/05/soap-envelope">
  <Code>
    <Value>Sender</Value>
    <Subcode>
      <Value xmlns:a="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">a:FailedAuthentication</Value>
    </Subcode>
  </Code>
  <Reason>
    <Text xml:lang="en-AU">ID3242: The security token could not be authenticated or
authorized.</Text>
  </Reason>
</Fault>
```

Example 2: E2192

```
<Fault xmlns="http://www.w3.org/2003/05/soap-envelope">
  <Code>
    <Value>Receiver</Value>
    <Subcode>
      <Value xmlns:a="http://vanguard.business.gov.au/2009/02">a:E2192</Value>
    </Subcode>
  </Code>
  <Reason>
    <Text xml:lang="en-AU">Claim data not sychronised in data source. MAS-ST Reference:
[3aef21e-d9cd-4839-f125-b094dd3e9803]</Text>
  </Reason>
  <Detail xmlns:s="http://www.w3.org/2003/05/soap-envelope">
    <STSFault xmlns="http://vanguard.business.gov.au/2009/02"
xmlns:i="http://www.w3.org/2001/XMLSchemainstance">
      <EventCode>E2192</EventCode>
      <EventSeverity>Severe</EventSeverity>
      <EventDescription>Dynamic claims data not in database.</EventDescription>
      <UserAdvice>Agency to advise Business User to re-attempt the request. If the problem
persists the Agency should contact the ATO Service Desk.</UserAdvice>
      -
    </STSFault>
  </Detail>
</Fault>
```

Example 3: E1001

```
<Fault xmlns="http://www.w3.org/2003/05/soap-envelope">
  <Code>
    <Value>Receiver</Value>
    <Subcode>
      <Value xmlns:a="http://vanguard.business.gov.au/2009/02">a:E1001</Value>
    </Subcode>
  </Code>
  <Reason>
    <Text xml:lang="en-AU">Service is not available.Event Code: [E1001]. Event Severity: [Severe].
Event Description: [Service is not available.]. User Advice: [Agency to advise Business User to re-
attempt the request. If the problem persists the Agency should contact the DIISR Service Desk.].
Agency Reference: []. MAS-ST Reference: [3cfd2359-0608-414a- 8385-2ef894326763].
Transaction Id: [].</Text>
  </Reason>
  <Detail xmlns:s="http://www.w3.org/2003/05/soap-envelope">
    <STSFault xmlns="http://vanguard.business.gov.au/2009/02"
xmlns:i="http://www.w3.org/2001/XMLSchemainstance">
      <EventCode>E1001</EventCode>
      <EventSeverity>Severe</EventSeverity>
      <EventDescription>Service is not available.</EventDescription>
      <UserAdvice>Agency to advise Business User to re-attempt the request. If the problem
persists the Agency should contact the ATO Service Desk.</UserAdvice>
          </STSFault>
  </Detail>
</Fault>
```

Example 4: Failed Authentication due to expired initiating party certificate

```
<Fault xmlns="http://www.w3.org/2003/05/soap-envelope">
  <Code>
    <Value>Sender</Value>
    <Subcode>
      <Value xmlns:a="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">a:FailedAuthentication</Value>
      <Subcode>
        <Value xmlns:a="http://vanguard.business.gov.au/2009/02">a:E2015</Value>
      </Subcode>
    </Subcode>
  </Code>
  <Reason>
    <Text xml:lang="en-AU">Business User Credential expired on [25/04/2009 12:02:56 AM].
Event Code: [E2015]. Event Severity: [Normal]. Event Description: [Business User Credential
expired on [25/04/2009 12:02:56 AM].]. User Advice: [Agency to advise Business User that their
Credential has expired and they must contact the issuing Certificate Authority for a new
Credential.]. Agency Reference: []. MAS-ST Reference: [19cada11-77fc-4794-8ddf-
2c28744fbe8d]. Transaction Id: [].</Text>
  </Reason>
  <Detail xmlns:s="http://www.w3.org/2003/05/soap-envelope">
    <STSFault xmlns="http://vanguard.business.gov.au/2009/02"
xmlns:i="http://www.w3.org/2001/XMLSchemainstance">
```

```
      <EventCode>E2015</EventCode>
      <EventSeverity>Normal</EventSeverity>
      <EventDescription>Business User Credential expired on [25/04/2009 12:02:56
AM].</EventDescription>
      <UserAdvice>Agency to advise Business User that their Credential has expired and they must
contact the issuing Certificate Authority for a new Credential.</UserAdvice>

    </STSFault>
  </Detail>
</Fault>
```