Australian Government

Department of Industry, Innovation, Climate Change, Science, Research and Tertiary Education

VANGUARD

GOVERNMENT AUTHENTICATION SERVICES

S007 Security Token Service

Technical Service Contract

Contract Version 1.2

Document Revision Rev 2.6

8 Jun 2012

Unclassified

Document Administration

Service Id	S007
Service Name	Security Token Service
Contract version	1.2
Document Version	2.6
File Location	TFS:\VANguard\Architecture\Documents\Design\Technical Service Contracts\
File Name	S007 Security Token Service TSC.doc

Development History

Version	Date	Author	Details
0.1	12 Feb 2009	G. Georgiou	Initial draft TSC documentation
0.2	10 Mar 2009	G. Georgiou	Updates to TSC based on requirements finalisation
0.3	17 Mar. 09	G. Georgiou	Updates to TSC reviewed and worked examples
0.4	3 April 2009	G. Georgiou	Service endpoints updated. Minor correction to example response.
0.5	24 April 2009	G. Georgiou	Service endpoint updated.
0.6	28 May	G. Georgiou	Update to text descriptions.
0.7	3 June 2009	G. Georgiou	Added claims processing rules, and updated service endpoints reflecting contract version no.
0.8	27 July 2009	I. Otto	Change from message confidentiality to transport confidentiality for PCR 37
0.9	28 Aug 2009	I. Otto	Updated comment on maximum token lifetime as per CR_022 reducing from 8 hours to 30 minutes.
1.0	4 Sep 2009	I. Otto	Corrected Common Elements Reference
1.1	16 Feb 2010	I. Otto	Corrected Common Elements Reference, now 1.15 to include OID fix for ABR credentials
1.2	7 Apr 2010	M. Young	Expanded fault section.
1.3	5 May 2010	M. Young	Updated fault consistency with SBR fault sections
1.4	11 May 2010	M. Young	Added ActAs details.
1.5	24 May 2010	M. Young	Minor updates per comments from GG.
1.6	1 Jun 2010	M. Young	Updated to include WS-security faults.
1.7	15 Jun 2010	G. Georgiou	Update to include definition of fault contract.
1.8	21 July 2010	M. Young	Removed preprod endpoints.
1.9	22 Nov 2010	G. Georgiou	Updated Error Codes
2.0	2 Dec 2011	B. Bildstein	TSC v1.2, including new error codes, more details of delegation (ActAs) functionality, incorporating feedback from I. Otto.
2.1	20 Dec 2011	B. Bildstein	Corrected omission of event codes E2190 and E2192; Clarify current and removed error codes; Clarify fault format.
2.2	8 Feb 2012	B. Bildstein	Include changes to event codes over previous two TSC versions
2.3	29 May 2012	T. Kerin	Indicate that E2014 and E2015 are now returned in fault.
2.4	4 Jun 2012	T. Kerin	Deleted example STSFault fragment from 4.1.4 in favour of the full examples later in the section. Updated Sub Code for E2014 and E2015. Updated examples and explicitly removed E2003 from error codes.

			Changed envelope version on faults to Soap 1.2. Removed the wsse:MessageExpired fault code, as the wsse:InvalidSecurity code supersedes it. Added vanguard namespace to fault E**** codes.
2.5	6 Jun 2012	T. Kerin	Minor corrections.
2.6	8 Jun 2012	T. Kerin	Updated with short note about subject claims and delegation. Renamed “principal” to “subject” in ActAs and delegation sections.

Copyright

Approvals

Recommended by:

Name: Ian Otto

Position: Principal Researcher

Signed:_____

Date:_____

Approved by:

Name: Malcolm Young

Position: Service Development Manager

Signed:_____

Date:_____

Business Approval by:

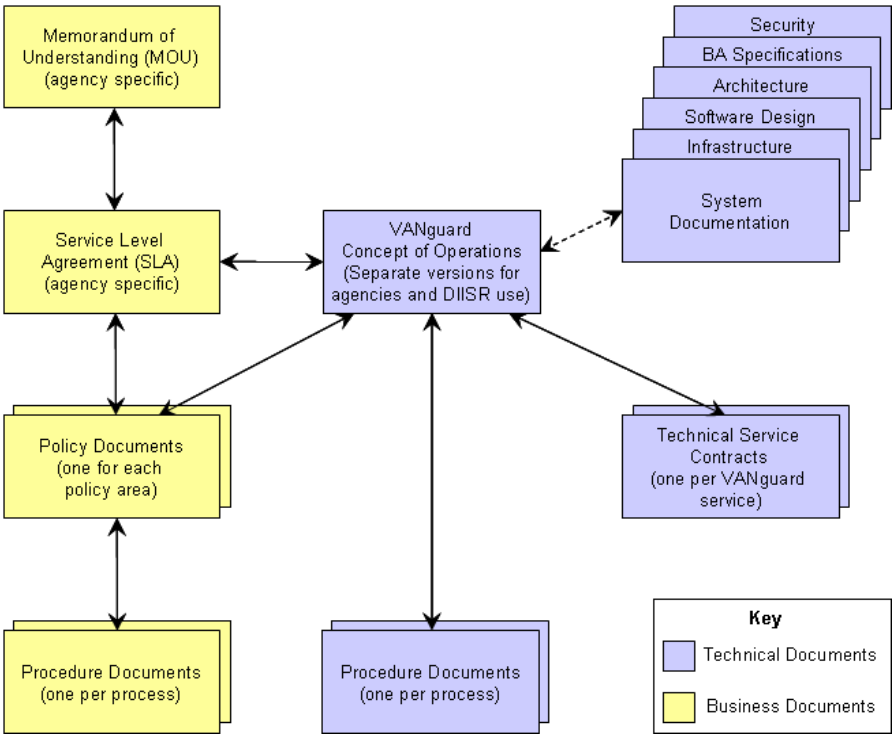
Name: Michelle Holmes

Position: Director Stakeholder Management

Signed:_____

Date:_____

VANguard Documentation Map



References and Related Documents

[VANguard CONOPS]	VANguard Team (2007) "*VANguard Concept of Operations*"
[Common Elements]	VANguard Team (2009) “Common Elements for VANguard Services” Document Revision V1.15
[WS-Trust]	*WS-Trust 1.3 OASIS Web Service Secure Exchange (*http://docs.oasis-open.org/ws-sx/ws-trust/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/)
[WS-PolicyAttachment]	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-soap-message-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/wss-v1.1-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/

Table of Contents

1. Introduction

1.1. About this Document

This document describes the interface between an Initiating Party and VANguards Security Token Service.

VANguard calls such an interface description a *Technical Service Contract* (TSC).

1.2. Who should use this Document

This document is a high level design artefact that sits on the border between requirements and design.

It 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 section 'Business Summary'

Business Analysts should read the whole document.

Developers and Testers should read the whole document

1.3. Terminology

Term	Description
SAML	Secure Assertion Markup Language – an OASIS standard
SAML Assertion	A signed XML block asserting identity of a Business User. The XML conforms to the SAML Core schema.
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, VANguard issues a signed SAML Assertion. A serialisation of the claims that are digitally signed by the VANguard.
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

1.4. Changes since Last Contract Version

- Service Endpoint has been updated to reflect new TSC version (1.2)
- Updated details of the *ActAs* element in the request
- Added details of the *Actor* element in the response
- BusinessContext element in the response has been renamed to STSFault
- Updates to error codes

2. Business View

2.1. Document Context

2.1.1. Service Identification

Service Name

S007 Security Token Service

Version

The scope and depth of VANguard's services will change over time. Hence any technical service contract must always be versioned so that parties can develop against a known stable interface.

This document describes version 1.2 of the S007 Security Token Service contract.
Since this document may change without impacting the service definition in any way, this document has a revision number separate to the contract version number. Refer to the section titled changes since last contract version (above) for a summary of changes for each revision.

Exposure

This service is accessed from the Internet and is exposed by the VANguard public zone.

2.1.2. 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 VANguard, and that the credential holder is who they claim to be for a given business transaction. The RP is leveraging VANguard's identity infrastructure to allow the IP to gain access to the RP's services. This removes the need for the RP to maintain its own authentication infrastructure. The RP can utilise externally issued credentials to expand their reach to businesses, eliminating the management overhead of provisioning, issuing and maintaining credentials.

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 VANguard 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 VANguard Security Token Service (STS). The STS issues a security token to an IP, for consumption by the relying party (RP).

2.1.3. 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 will is required to specify the token type returned, the only options supported are SAML 1.1 and SAML2.0 encoded token.

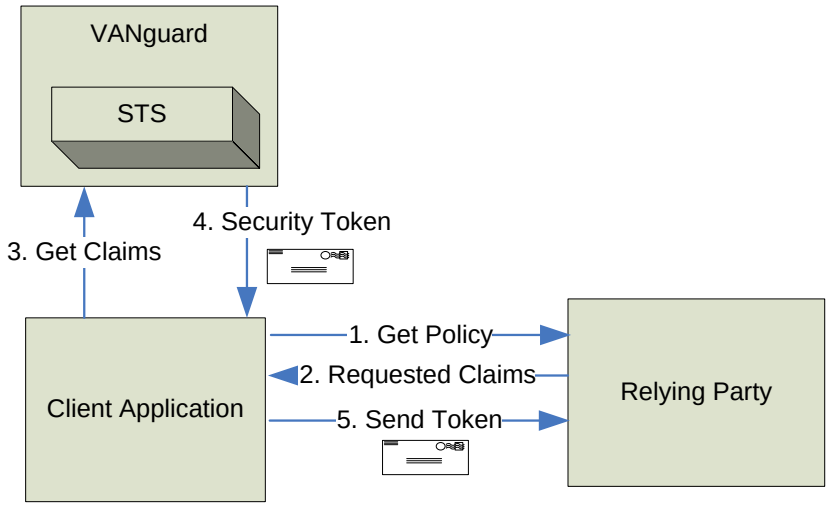


Figure 1 Basic Scenario

Figure 1 shows a basic scenario of the interaction between the initiating party, relying party and VANguard under a typical scenario.

1. The client application wants to access protected resources managed by the relying party.

2. The relying party exposes a policy [WS-Policy] or WSDL that includes the list of claims the client application needs to present to conduct the business transaction. The policy also includes the location of VANguards Security Token Service (specified using [WS-Addressing]). In this case the end point for VANguard's STS.

   Note: This negotiation could have already occurred (pre-negotiated out of band/pre-compiled in the requesting application) and the client application has the policy information detail required to request the token from VANguard bypassing this step.

3. The client application initiates a request to VANguard's STS requesting the claims required by the relying party. A default set of claims will be returned if claim elements omitted.
4. Vanguards STS 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.
5. 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.

2.1.4. Policy and Risks Security Tokens

VANguard 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.

VANguard 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.

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 VANguard 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 contains is acceptable for the transaction[1].
- 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 s session key contained VANguard 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.

[1] VANguard will issue an STS token based on any approved credential (ABR, Medicare, some Verisign, …). The type of the credential will appear as a claim in the assertion. It is up to the relying party to reject tokens based on credentials that are not suitable.

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. VANguard will be 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[2]. 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 VANguard 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.

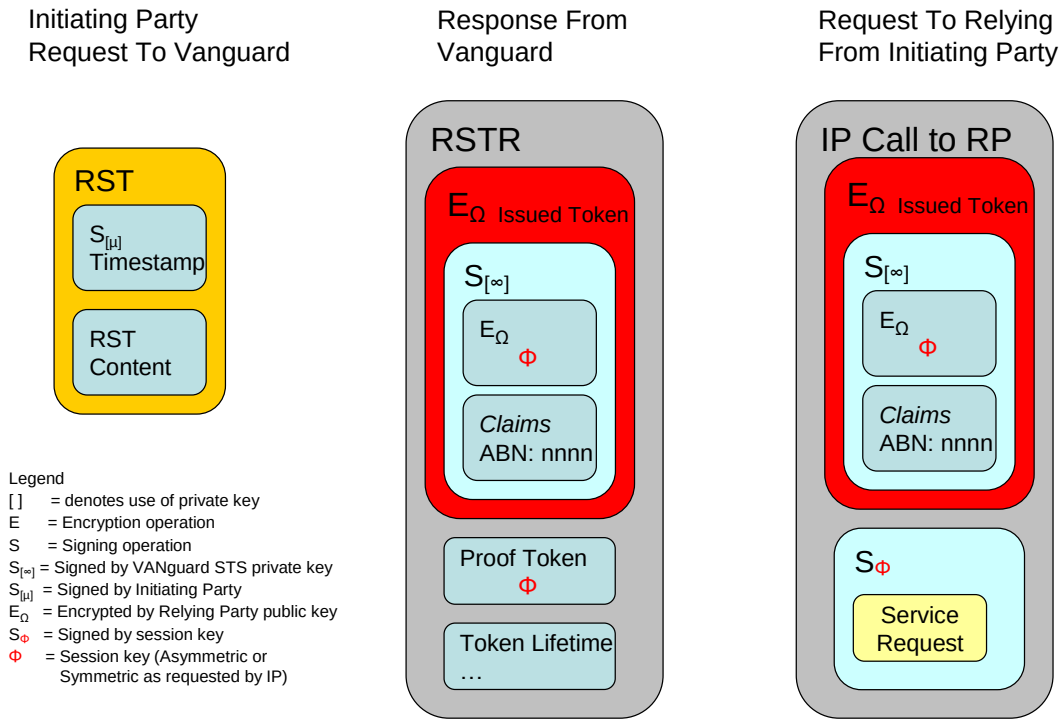
The STS returns a RSTR response message contained in the response is a signed assertion termed Issued Token and a Signed Proof Token. The RSTR contains a WS-Security secured assertion for consumption by the relying party, and a proof token for the initiating party.

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

[2] SSL is configured in accordance with the Australian Government Protective Security Manual.

2.3. WS-Trust Message Structure

The following diagram pictorially represents the WS-Trust message structure for the interactions between each of the parties from VANguards perspective. Note this is not a detailed representation and should be used as a conceptual model.



2.4. 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 VANguard.

2.5. Claims Processing

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

VANguards STS allows for three options for evaluating claims data and uses the following rules in processing claims data for a RST request. Claims are categorised in into distinct categories outlined below. VANguard will determine if a security token can be issued after successful verification of an initiating parties 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. Each certificate type will have a subset of the claims specified in [Common elements] set as the default claims.

Compulsory Claims

A compulsory claim is a claim that VANguard 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 VANguards STS 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 VANguards STS that if a claim value cannot be determined a security token should not be issued.

3. Technical View

3.1. Service Endpoints

The S007 Security Token Service will be available from the following URLs:

Environment	*Security Token URL*
Production	https://authentication.business.gov.au/R3.0/vanguard/S007v1.2/service.svc
Third Party Test	https://thirdparty.authentication.business.gov.au/R3.0/vanguard/S007v1.2/service.svc

Note: the v1.1 version of the STS will continue to be available concurrently with this release for a period of time.

3.1.1. Naming convention of Service Endpoint

```
<Protocol>://[<Environment>.]<domain>/R<Intermidiatry>/vanguard/<ServiceID>
v<ContractNo>/Service.svc
```

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 VANguard'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 VANguard to before use is authorised to the service.
4. The Initiating party has been issued with a credential compatible with VANguards verification infrastructure.

3.2.2. Post Conditions

1. The service request will be recorded by VANguard 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

Prefix	Namespace	Specification(s)
S11	http://schemas.xmlsoap.org/soap/envelope/	[SOAP]
S12	http://www.w3.org/2003/05/soap-envelope	[SOAP12]
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	[WS-Security]
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	[WS-Security]
wsse11	http://docs.oasis-open.org/wss/oasis-wss-wsecurity-secext-1.1.xsd	[WS-Security]
wst	http://docs.oasis-open.org/ws-sx/ws-trust/200512	[WS-Trust]
ds	http://www.w3.org/2000/09/xmldsig#	[XML-Signature]
xenc	http://www.w3.org/2001/04/xmlenc#	[XML-Encrypt]
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	[WS-Policy]
wsa	http://www.w3.org/2005/08/addressing	[WS-Addressing]
xs	http://www.w3.org/2001/XMLSchema	[XML-Schema1] [XML-Schema2]
i	http://schemas.xmlsoap.org/ws/2005/05/identity	[Claims]
v	http://vanguard.business.gov.au/2009/02	

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 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	Issue a WS-Trust 1.3 RequestSecurityTokenResponse: • Token Type • RequestType • AppliesTo • KeyType • KeySize • Lifetime

		• Lifetime • ActAs	• 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 VANguard.

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 VANguard private key.

Communication protocols

The following Web services standards will be used;

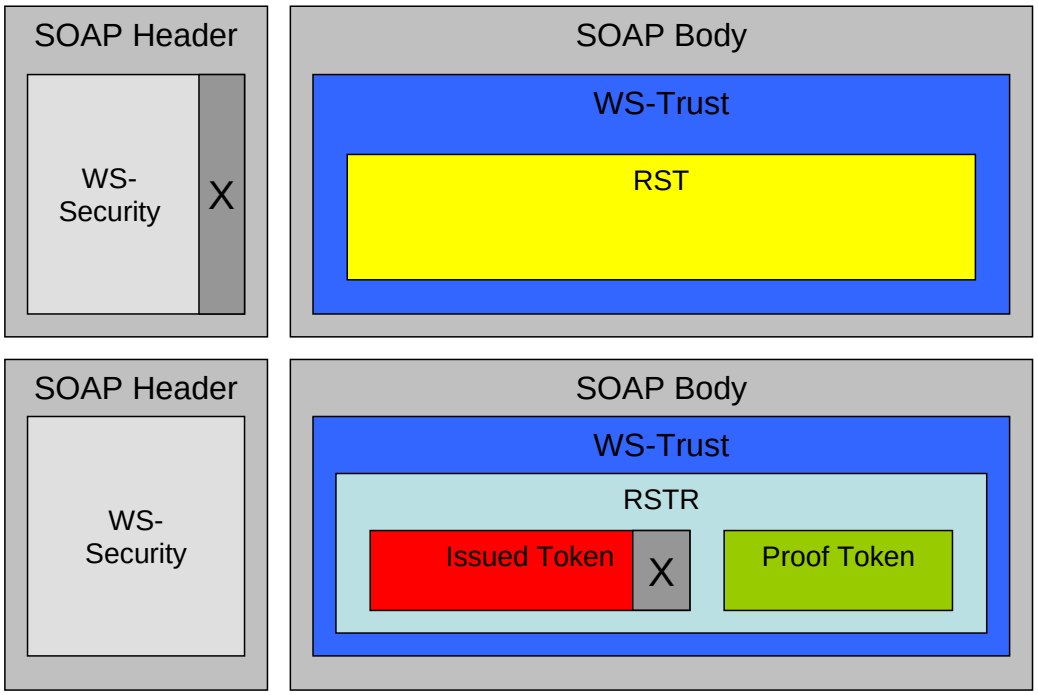
- Messages will be contained in a SOAP envelope
- Service messaging will be done using XML
- Web services descriptions will be expressed using WSDL
- An XML Request will be used to carry the request for a security token. The request will be placed in a SOAP body.
- An XML Response will be used to convey 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 VANguard private signing key.

Signing under this contract is summarised in the diagram below:



3.3. STS with delegation: ActAs and Actor details and examples

When a SAML assertion is included in the ActAs element of the request, the response will describe the subject (Initiating Party) of the original SAML assertion, not the requestor performing the request (in this case the Relying Party for the passed in SAML token).

The response will also contain claims about the Relying Party. These claims will be included in an AttributeStatement, which will in turn be included as an Actor claim in the AttributeStatement of the subject.

3.3.1. RST ActAs element

The ActAs element must be a valid VANguard-signed SAML assertion. This can be obtained from a call to either the VANguard User Authentication Service, or the VANguard Security Token Service.

Example

```
<wst:RequestSecurityToken xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">
  <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">

<Address>http://ac01wvwebd01/RelyingPartyServicev3.0/Saml11Service.svc</Address>
    </EndpointReference>
  </wsp:AppliesTo>
  <wst:Claims xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity"
Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity">
    <i:ClaimType
Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/commonname"
Optional="false" />
    <i:ClaimType
Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/givennames"
Optional="false" />
  </wst:Claims>
  <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/SymmetricKey</wst:KeyType>
  <wst:Lifetime>
    <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">2011-10-24T22:48:34.184Z</wsu:Created>
  </wst:Lifetime>
  <tr:ActAs xmlns:tr="http://docs.oasis-open.org/ws-sx/ws-trust/200802">
    <saml:Assertion Version="2.0" ID="_3e4e371c-b52c-4e32-9d8f-a06e64a90bd3"
IssueInstant="2011-10-24T22:46:42Z"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">

<saml:Issuer>https://vanguard.business.gov.au/UserAuthentication/v4.0/</saml:Issuer>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        ...
      </Signature>
      ...
      <saml:AttributeStatement>
        <saml:Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/credentialtype"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
FriendlyName="Credential Type">
          <saml:AttributeValue>ABR_User</saml:AttributeValue>
        </saml:Attribute>
        ...
      </saml:AttributeStatement>
    </saml:Assertion>
  </tr:ActAs>
  <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-
```

```
trust/200512/Issue</wst:RequestType>
  <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
</wst:RequestSecurityToken>
```

3.3.2. RSTR AttributeStatement Actor claim

When a request with an ActAs element is processed, the claims in the resulting AttributeStatement will describe the subject from the ActAs token, instead of the identity of the caller. The claims about the subject will be the same as those in the passed ActAs token (excluding any Actor claim) regardless of what claims were specifically requested. Additionally, the AttributeStatement in the response will contain an Actor claim.

This Actor claim is an XML-encoded string. It contains a list of attributes that describe the caller of the service that passed in the original ActAs token. The Actor claim contains a single Actor element, which contains a sequence of Attribute elements. The claims that will be included in the Attribute sequence of the Actor are the Default Claims for Relying Party making the call (based on the Relying Party's VANguard Agency certificate).

Note that such delegation can be chained. For example, an Initiating Party (IP) obtains a security token from VANguard, and then a Relying Party (RP1) makes an STS request acting as IP, and obtains a response. The AttributeStatement in this response will describe IP, and contain an Actor claim that describes RP1. A second Relying Party (RP2) may now invoke the STS using this new token as an ActAs token. In this case, the resulting AttributeStatement will still describe IP, and still contain an Actor claim that describes RP1. However, in this resulting AttributeStatement (about IP), in the Actor claim (which describes RP1), there will be a nested Actor claim that describes RP2.

Example (single delegation):

```
<Assertion ID="_af1ecc53-a632-4d16-af5a-b0aec9b877dd" IssueInstant="2011-11-15T22:25:37.289Z"
Version="2.0" xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
  <Issuer>VANguard Security Token Service</Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
  ...
  <AttributeStatement>
    <Attribute Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/samlsubjectid"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" FriendlyName="SAML Subject ID"
d3p1:OriginalIssuer="Vanguard"
xmlns:d3p1="http://schemas.xmlsoap.org/ws/2009/09/identity/claims">
      <AttributeValue>ABRP:12300002581_10080080</AttributeValue>
    </Attribute>
    ...
    <Attribute Name="http://schemas.xmlsoap.org/ws/2009/09/identity/claims/actor">
      <AttributeValue>&lt;Actor&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/commonname"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;DIISR Test Agency
5&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/subjectdn"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;CN=DIISR Test Agency 5,
OU=For test purposes ONLY, OU=VANguard, O=DIISR Test Agency 5,
C=AU&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/issuerdn"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;CN=Australian Government
Notary Services OCA, OU=For test purposes ONLY, OU=Australian Authentication and Notary
Services, O=Australian Government, C=AU&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/businessname"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;DIISR Test Agency
5&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/abn"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;55566677788&lt;/Attribut
```

```
eValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/notbeforedate"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;2008-05-
13T00:00:00Z&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/notafterdate"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;2012-05-
12T23:59:59Z&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/certificateserialnumber"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;769917635505441473523871
77952151692927&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/certificatepolicyoid"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;1.2.36.1.1001.50.8.1&lt;
/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/samlsubjectid"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;CERT:8A7C72A3342B1FE12A3
B92FF41130CC16F06F3BC&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/credentialtype"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;VANguard_Agency&lt;/Attr
ibuteValue&gt;&lt;/Attribute&gt;&lt;Attribute
Name="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/stalecrlminutes"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;0&lt;/AttributeValue&gt;
&lt;/Attribute&gt;&lt;Attribute
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"&gt;&lt;AttributeValue&gt;CERT:8A7C72A3342B1FE12A3
B92FF41130CC16F06F3BC&lt;/AttributeValue&gt;&lt;/Attribute&gt;&lt;/Actor&gt;</AttributeValue>
    </Attribute>
  </AttributeStatement>
  <AuthnStatement AuthnInstant="2011-11-15T22:24:17.000Z">
    <AuthnContext>
      <AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:X509</AuthnContextClassRef>
    </AuthnContext>
  </AuthnStatement>
</Assertion>
```

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()

Description

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 RequestSecuirtyToken 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 VANguard 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 VANguard.
- The mandatory claims requested are known to VANguard.
- The lifetime requested is greater than the maximum will be set to the maximum (currently 30 minutes).
- 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:

- VANguard cannot validate the certificate.
- The relying party cannot be identified.
- The certificate type presents is not supported by VANguard.
- VANguard 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 summaries 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>
```

Element Name: RequestSecurityToken

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

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 RequestSecuirtyToken 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/ws-
sx/ws-trust/200512"
Context="http://RelyingParty.gov.au/RST/UnqiueRequestID">
    ...
</wst:RequestSecurityToken>
```

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 are not specified

Error Conditions

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

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

Example: The following example requests a saml 2.0 token to be returned in response.

```
<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>
```

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 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: The following example requests the issuance of a security token.

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

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.

Processing Rules:

Example: The following example requests the issuance of a security token, specfiying 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.

Element Name: KeyType

Description: This element is used to specify the type of key required to prove possession. The proof key is place in the proof token and the issued token regardless of the KeyType option selected. The initating 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 inlcude it in the issued token and proof token.

If public key is chosen this instructs the STS to generate a key pair (AK_1,AK_2) AK_1 is placed in the proof token and AK_2 is placed in the issued token. The initiating party proves possession by decrypting proof token and encrypting the service request with AK_1, The relying party uses AK_2 to verify the service request and deny or grant access.

Element Location: /wst:RequestSecurityToken/wst:KeyType

Element Attributes: None

Validation Rules:

If specified this value must be equal to either the Public or Symmetric Key, refer to accepted values for this element.

The default is symmetric key if not specified.

Accepted Values:

To use a public key

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

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 specifies 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>
```

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 dependant on the keytype chosen. The default key size if not specified is 256 for symmetric key type. The default for assymetric key type is 1024.

Accepted Values:

A 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>
```

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 indicating the syntax of the cliams in the containing elelment.	M

Validation Rules:

If the claims element is specified then the dialect must be specified 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 lease 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 ommited, a default set of claims are returned based on the intiating party certificate type present in the request.

Accepted Values:

The claims element must be well formed 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 claims is required. Unknown Optional claims will not trigger a SOAP fault if unknown.
2. The claim type cannot be compiled for the certificate used to signing the request (IP certificate). For example if a device certificate was used to sign the request and contains a request for 'email address' claim a SOAP fault wst:BadRequest will be generated because a device certificate cannot contain an email address claim.

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://vanguard.ebusiness.gov.au/2008/06/identity/claims">
        ...
   <i:ClaimType Uri="http://vanguard.business.gov.au/2009/03/australianbusinessnumber"/>
        ...
 </wst:Claims>
    ...
</wst:RequestSecurityToken>
```

Element Name: ClaimType

Description: This element is used to request specific cliams 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 indicating the syntax of the cliams in the containing elelment.	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 ommited it defaults to mandatory “false”	O

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 can not evaluated and returned VANguard 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: 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 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"/>
   <i:ClaimType
           Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
           Optional="false"/>
        ...
 </wst:Claims>
    ...
</wst:RequestSecurityToken>
```

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. A maximum of up to thirty mminutes inclusive can be requested. If a token is request greater than the maximum it will be constrained to thirty minutes.

Accepted Values:

The lifetime element has to 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>
```

Element Name: Expires

Description: This element is used to specify the upper bound of the validaty time period for the requested secuirty 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.

VANguard implements a minimum and maximum bound for requests measured using VANguard clocks. The difference between the VANguard 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 VANguard clocks will be used to specify the creation date. VANguard does not support post dated requests and as such ignored 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>
```

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. The identity that the requestor wants to act as as is specified by placing a security token previously issued by VANguard to the identity within this element.

A security token is not valid to be used in the ActAs element unless the current request is being made by the recipient specified in the security token.

The claims in the resulting token will describe the subject of the ActAs token, not the requestor. Claims describing the requestor will be present in the response, in the Actor attribute in the AttributeStatement.

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

Element Location: `/wst:RequestSecurityToken/wst:ActAs`

Element Attributes: None

Validation Rules:

The ActAs element must not be empty.

Accepted Values:

The token contained in this element must be a valid token previously issued by VANguard, with the current requestor also being the recipient named in the security token.

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 token for the purpose of the requestor acting as a different initiating party.

```
<wst:RequestSecurityToken xmlns:wst="...">
    ...
 <wst:ActAs>
    <Assertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
       ...
    </Assertion>
 </wst:ActAs>
    ...
</wst:RequestSecurityToken>
```

The following extract shows a complete RST request for issuance of a security token specifying the relying party, the token encoding format as SAML 2.0, A request for two claims one optional and one mandatory, specifies an expiry date, the proof token use asymmetric keys, the generate key size be at least 1024 bit. *(SOAP Envelope and Security is not shown for brevity)*

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

    <wsp: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>
    </wsp:AppliesTo>

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

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

    <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>2009-02-04T05:20:15.000Z</wsu:Expires>
    </wst:Lifetime>

   <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/PublicKey
   </wst:KeyType>

   <wst:KeySize>1024</wst:KeySize>

</wst:RequestSecurityToken>
```

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	Integer	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.oasis-open.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>
```

Element Name: RequestSecurityTokenResponseCollection

Description: This element can be used to return one or more security token. 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>
```

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 is echoed from the request if specified, This field can be used to link requests with responses by initiating party applications.	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:RequestSecurityTokenResponse>
    ...
</wst:RequestSecurityTokenResponseCollection>
```

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

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>
```

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 specifiy 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 specfiying the relying parties 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>
```

Element Name: KeyType

Description: This element is used to indicate keytype used for protecting the proof key the response

This element must be in the response, and should match the corresponding element (or default value) in the RST.

Namespace:

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>
```

Element Name: KeySize

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

Namespace: None

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>
```

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>
```

Element Name: RequestedSecurityToken

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

Namespace: N/A

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

Element Attributes: None

Validation Rules:

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

Example: The following example response to the issuance of a security token, specfies 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>
```

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.

Namespace: N/A

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, specfies 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>
```

Element Name: RequestedProofToken

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

Namespace: N/A

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

Element Attributes: None

Validation Rules:

Example: The following example response to the issuance of a security token, specfiying 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>
```

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.

Namespace: N/A

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

Element Attributes: None

Validation Rules:

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>VANguard 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......+6dXvzE9QIkVJwn7AlS4o=</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>
                                897819763472042199712415668818927227 98
                            </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-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: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:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
    <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</wst:RequestType>
    <wst:KeyType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/SymmetricKey</wst:KeyType>
  </wst:RequestSecurityTokenResponse>
</wst:RequestSecurityTokenResponseCollection>
```

4.1.4. Exceptions/Faults

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

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 reccommendation

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

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 of a request.

Element Name: `Subcode`

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

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

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

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

Element Name: `EventCode`

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

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

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

Element Attributes: None

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

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

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

The following table outlines the errors that are returned.

Please note the following differences from previous versions:

- The following error codes are no longer returned: E1003, E1004, E2001, E2003, E2017, E2020, E2029, E2169, E2180
- E2014 (revoked cert) and E2015 (expired cert) are now returned once again.
- All error codes in the wsse namespace will be returned as unsecured faults.
- The following errors sub codes are new: E2040, E2041, E2042, E2043, E2044.

Sender Error Codes

Fault Code	*Sub code*	*Vanguard Sub Code*	*Description*
env:Sender	wsse:FailedAuthentication	v:E2014	The provided initiating party certificate was found to be revoked.
env:Sender	wsse:FailedAuthentication	v:E2015	The provided initiating party certificate was found to be expired.
env:Sender	wst:RequestFailed	v:E2040	Invalid ActAs Token, The ActAs token was invalid.
env:Sender	wst:InvalidRequest	v:E2041	Request type is not supported.
env:Sender	wst:RequestFailed	v:E2042	Unsupported token type. The token type specified in the TokenType element was not supported.
env:Sender	wst:RequestFailed	v:E2043	Missing AppliesTo value in request. The AppliesTo element was either missing or empty.
env:Sender	wst:RequestFailed	v:E2044	Unspecified relying party in request. The relying party referred to in the AppliesTo element could not be found.
env:Sender	N/A	v:E2182	Missing Claim data. Initiating Party claim data is missing or not available.
env:Sender	N/A	v:E2183	One or more specified claims elements are not supported.
env:Sender	wsse:FailedAuthentication		The caller could not be authenticated.
env:Sender	wsse:FailedCheck		The signature or decryption was invalid.
env:Sender	wsse:InvalidSecurity		An error was discovered processing the `<wsse:Security>` header. This can occur due to clock skew with the originating machine.
env:Sender	wsse:InvalidSecurityToken		An invalid security token was provided
env:Sender	wsse:SecurityTokenUnavailable		Referenced security token could not be retrieved.

Receiver Error Codes

Fault Code	*Sub Code*	*Vanguard Sub Code*	*Description*
env:Receiver	N/A	v:E1001	Service is not available
env:Receiver	N/A	v:E2190	Claim data currently not available. Initiating party certificate is newly issued, and VANguard does not yet have claims data for this certificate.
env:Receiver	N/A	v:E2192	Claim data currently not available.

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.</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/XMLSchema-
instance">
      <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 DIISR Service Desk.</UserAdvice>
      <VanguardReference>3aef21e-d9cd-4839-f125-b094dd3e9803</VanguardReference>
    </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: []. VANguard 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/XMLSchema-
instance">
      <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 DIISR Service Desk.</UserAdvice>
      <VanguardReference>7dc57c24-7d1a-42b7-9ab2-9ba43458e7e0</VanguardReference>
    </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: []. VANguard 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/XMLSchema-
instance">
      <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>
      <VanguardReference>19cada11-77fc-4794-8ddf-2c28744fbe8d</VanguardReference>
    </STSFault>
  </Detail>
</Fault>
```