RealMe

Assertion Service Messaging Specification

Version: 1.3 – APPROVED

Author: Richard Bergquist
Datacom Systems
(Wellington) Ltd

Date: 17 October 2013

Document control

Revisions

Version	Date	Owner	Changes
0.1	24/05/2012	Richard Bergquist Joachim Davies	RealMe Assertion SAML2.0 messaging specification draft.
0.2	01/08/2012	Richard Bergquist	Consolidation of RealMe Logon and Assert SAML2.0 messaging specifications into one consistent document.
0.3	15/08/2012	Richard Bergquist	V0.2 Review and feedback. Revisit NZCIQ bindings
0.4	18/09/2012	Richard Bergquist	V0.3 review feedback. Fork document into specification for only the Assertion Service.
0.5	23/10/2012	Richard Bergquist	Reformat and refactor.
1.0	15/11/2012	Mick Clarke	Marked as APPROVED
1.1	21/03/2013	Richard Bergquist	Edits for the SAML attributes from IAPs. Only the identity attributes are passed through. The XML attributes are not individually signed. Added new attribute for IVS FIT. Added sample corrected CIQ address for AVS. Inclusion of v0.2 of RealMe SAML v2.0 Messaging Errata.
1.2	16/05/2013	Richard Bergquist	JIRA REALME-374 Add a row to the messaging specification table in section 4.5.5 to translate a "urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal" error status code received from the Logon Service to an equivalent error status code returned from RealMe. This supports the case where the Customer has attempted tried to log on to in the RealMe Assert use case without having a current RealMe logon.
1.3	17/10/2013	Mark Pritchard	Updated Opaque token to reflect its safe base 64 encoding. Updated IVS FIT to contain the NameId element rather than a simple string value.

Contributors

Person	Role
Richard Bergquist	Datacom RealMe Solution Architect

Approvers

The approvers for this and other documentation will be the major members of the *Joint Technology Governance Group* (JTGG), namely:

Name	Role	Approve (Y | N | Condition)	Version	Date
Mick Clarke	Lead RealMe Architect	Y	1.1	21/03/2013
Steffen Sorensen	DIA Enterprise Architect	Y	1.1	21/03/2013
Venkat Maddali	DIA Solution Architect	Y	1.1	21/03/2013

Table of Contents

1 Introduction

1.1 Overview

RealMe exposes a SAML v2.0 interface to integrating Client organisations which conforms to the New Zealand Security Assertion Messaging Standard (NZ SAMS). This document specifies the SAML v2.0 interface.

1.2 Document Purpose

This document has been created to describe the interface that exists between the RealMe Assertion Service ('RealMe') and a Client Service Provider's application ('the SP').

This document serves two key purposes:

1. It describes the messaging interfaces sufficiently for third party software developers to design and build SAML based interfaces to RealMe;
2. It describes the messaging interfaces sufficiently for RealMe vendor to develop the interfaces.

This document will be used to describe an NZ SAMS profile where unless specified all intent from the OASIS SAML v2.0 standard hold. This document will specify constraints on a NZ SAMS implementation for RealMe's specific use by narrowing certain functionality.

The relationship between the OASIS SAML v2.0, NZ SAMS and this specification are modelled as:

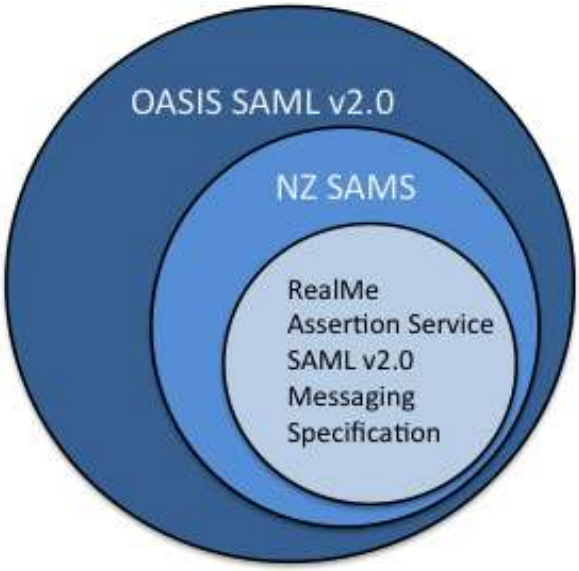

Figure 1 - Standards Nesting

In general SAML v2.0 implementations are expected to conform and be certified against the wider SAML v2.0 standard which is a superset of NZ SAMS and, hence, this document. By narrowing the NZ SAMS requirements as specified in this document some implementers that are not fully SAML v2.0 and NZ SAMS conformant may be still conformant to this specification. As a consequence this specification can broaden the SAML v2.0 product procurement base for SP implementers.

There are aspects of SAML v2.0 interoperability that the SAML v2.0 standard does not specify. This document addresses those gaps by defining the use of some parameters to values that apply in the RealMe context.

1.3 Document References

The following NZ government standard references are used throughout this document:

Reference	Name	Description
NZ SAMS	New Zealand Security Assertion Messaging Standard. (June 2008 version 1.0 - ISBN 978-0-478-30344-5) http://ict.govt.nz/guidance-and-resources/standards-compliance/authentication-standards/new-zealand-security-assertion-messaging-standard	Prescribes messaging standards for communicating a range of security assertions (authentication, identity attributes and authorisation) in New Zealand government online services.
NZCIQ	New Zealand Government OASIS CIQ Profile. http://www.authentication.webstandards.govt.nz/new-zealand-government-oasis-ciq-profile/	A New Zealand Government Profile of the OASIS CIQ v3 Standard. The OASIS CIQ Standard is an international standard for Customer Information Quality. It deals specifically with data exchange of Customer Information by providing a set of predefined XML Schemas for data exchange structures.
NZISM	The New Zealand Information Security Manual v1.01. http://www.gcsb.govt.nz/newsroom/nzism.html	The New Zealand Information Security Manual (NZISM) provides up-to-date technical policy to assist government departments and agencies in securing information systems and the data stored in those systems

Table 1 - NZ Government Standards

The following RealMe references are used throughout this document:

Reference	Name	Description
realme-int-guide	SAML v2.0 RealMe Integration Guide	A guide for SP integrating to RealMe using SAML v2.0. Contains the specific technical steps and requirements for an integrator to follow.

Table 2 - RealMe References

The following SAML v2.0 references are used throughout this document:

Reference	Name	Description
saml-core-2.0-os	Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	The core SAML v2.0 specification from OASIS.
saml-profiles-2.0-os	Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0	The profiles SAML v2.0 specification from OASIS.
saml-bindings-2.0-os	Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0	The bindings SAML v2.0 specification from OASIS.
saml-metadata-2.0-os	Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0	SAML profiles require agreements between system entities regarding identifiers, binding support and endpoints, certificates and keys, and so forth. A metadata specification is useful for describing this information in a standardized way. This document defines an extensible metadata format for SAML system entities.
saml-glossary-2.0-os	Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0	This normative document defines terms used throughout the OASIS Security Assertion Markup Language (SAML) specifications and related documents.

Table 3 - SAML v2.0 References

These documents can be found at http://saml.xml.org/wiki

The following third party references are used throughout this document:

Reference	Name	Description
ciq-3.0	Customer Information Quality v3.0 Specifications http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq	A XML based standard from OASIS to define a vocabulary to represent customer data, including identity related attributes.
xml-schema-datatypes	XML Schema Part 2: Datatypes. http://www.w3.org/TR/xmlschema-2	XML Schema datatypes from W3C.

Table 4 - Other References

Where these documents are referenced they will be surrounded by [ ] to contain the reference name. e.g. [saml-core-2.0-os].

1.4 Definitions

The following abbreviations will be used for the terms in the above documents listed in section 1.3.

Acronym	Description
Assertion Service	**RealMe Assertion Service** The RealMe hosted shared service that allows Customers to provide personal information about themselves from multiple authoritative sources to the client organisations online. A Customer is transferred to the ReaMe Assertion Service by a RealMe Client to verify their identity.
AVS	**Address Verification Service** A RealMe identity attribute provider; the NZ Post service that provides a mechanism to verify a customer's address.
Base64	A data encoding scheme whereby binary-encoded data is converted to printable ASCII characters. The only characters used are the upper- and lower-case Roman alphabet characters (A–Z, a–z), the numerals (0–9), and the "+" and "/" symbols, with the "=" symbol as a special suffix code.
Client	For the purposes of RealMe a Client is an integrated entity that relies on an Identity Assertion or Logon Assertion. The entity may be a sector, an individual agency or organisation, a set of services within an individual agency or organisation or a single service within an individual agency or organisation. An Client acts as a SAML SP and utilises the services of RealMe as IdP's
Customer	End user of the RealMe services and the owner of identity data. The Customer is the person who establishes the RealMe account and is the owner of the account and its contents.
FIT	**Federated Identity Tag** The unique value that references the Customer's igovt identity to a Client organisation. A FIT contains no identity information itself, and is opaque. It is directly related to an igovt id. For a given Customer, the FIT will be different for each Client organisation. Wraps the NameID that is returned in a SAML attribute statement from the IVS IAP.
FLT	**Federated Logon Tag** The unique value that references the Customer's logon to a Client organisation. A FLT contains no logon information itself, and is opaque. For a given Customer, the FLT will be different for each Client organisation. A synonym for the SAML NameID, returned in a SAML assertion from the RealMe Logon Service as an opaque reference of the logon of a user in a federated environment.
HTTP	**HyperText Transfer Protocol** An application layer protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for

Acronym	Description
	many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers.
HTTPS	**HTTP over SSL or HTTP Secure** HTTPS is the use of Secure Socket Layer (SSL) or Transport Layer Security (TLS) as a sub-layer under regular HTTP application layering. HTTPS encrypts and decrypts user page requests as well as the pages that are returned by the Web server. The use of HTTPS protects against eavesdropping and man-in-the-middle attacks. HTTPS was developed by Netscape.
IdP	**Identity Provider** A system entity that creates, maintains, and manages identity information for principals and provides principal authentication to other service providers within a federation, such as with web browser profiles. [saml-glossary-2.0-os] The SAML role of the RealMe Logon and Assertion Services.
IVS	**Identity Verification Service** A RealMe identity attribute provider; the DIA service that holds identities verified to a high level of confidence.
Logon Service	**RealMe Logon Service** The RealMe hosted shared service to perform the logon process for online services of participating Client organisations. A Customer is transferred to the ReaMe Logon Service by a RealMe Client or the RealMe Assertion Service for an authentication.
Logon	(noun) The combination of a username (logon identifier component) with one or more authentication keys (the authentication component) that is authenticated by the RealMe Logon Service. (verb) The action a user performs to supply their authentication credentials. RealMe shall require the Customer to logon at the RealMe Logon Service.
MTS	**Messaging Test Site** A web based tool for Client organisations to develop and test their RealMe SAML v2.0 integration and messaging. Implements this messaging specification exactly and provides an interactive tool to support the diagnosis of messaging errors.
Mutual SSL	Mutual SSL refers to two parties authenticating each other suitably using digital signatures. The client authenticates themselves to a server and that server authenticates itself to the client in such a way that both parties are assured of the others' identity by digital signatures. Mutual SSL provides the same things as SSL, with the addition of authentication and non-repudiation of the client and server using digital signatures. Mutual SSL is also known as Client TLS or Client SSL.
NTP	**Network Time Protocol** A protocol to exchange and synchronize time on computer networks.
Principal	A system entity whose identity can be authenticated.

Acronym	Description
Identity Privacy Domain	An identity privacy domain is a SAML v2.0 NameID generation space. SP's that reside in the same privacy domain will be returned the same SAML v2.0 NameID in the Assertion of the user's logon in the SAML response.
Safe Base64	A data encoding scheme whereby binary-encoded data is converted to printable ASCII characters. The only characters used are the upper- and lower-case Roman alphabet characters (A–Z, a–z), the numerals (0–9), and the "-" and "_" symbols, with the "=" symbol as a special suffix code. Used to encode XML formatted data without using XML characters in the encoding. See RFC 3548.
SAML	**Security Assertion Markup Language:** an XML-based standard that defines messages for communicating a range of security related statements about individual parties, including their authentication.
SOAP	**Simple Object Access Protocol:** A lightweight protocol that defines how information may be exchanged in a distributed and decentralized environment using XML.
SSO	**Single Sign On:** the ability for a user to be logged on via a single session authority to multiple SPs without having to re-enter their authentication credentials.
SP	**Service Provider.** A role donned by a system entity where the system entity provides services to principals or other system entities.
SSL	**Secure Socket Layer:** A protocol for transmitting sensitive information across the Internet in a secure way. The later TLS standard may also be used instead of SSL.
TLS	**Transport Layer Security:** TLS guarantees privacy and data integrity between client/server applications communicating over the Internet. TLS is the successor protocol to Secure Socket Layer [SSL], created by the Internet Engineering Task Force (IETF) for general communication authentication and encryption over TCP/IP networks.
UTC	**Coordinated Universal Time:** An international, highly accurate and stable uniform atomic time system.
User Agent	Web browser software that end users operate to access online services
XML	**Extensible Markup Language:** A markup language for Internet documents which allows designers to create their own tags (hence extensible).

Table 5 - Definitions

1.5 Assumptions

This document assumes the reader has:

- A working knowledge of the documents listed in section 1.3, and;
- An appreciation of the terms and abbreviations used in the documents listed and in section 1.4.

1.6 Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in IETF RFC 2119 [RFC2119].

1.7 Specification Compliance

For an implementation or deployment to call itself compliant with this specification it MUST satisfy all aspects of this document marked as MUST as well as all NZ SAMS and SAML v2.0 conformance and specification requirements that are relevant to the functionality covered in this document.

2 Messaging Flow

The following sequence diagram introduces the messaging flow between a Customer, a Client SP and RealMe Assertion Service acting as an IdP.

The sequence of messages is driven from the Web Browser SSO Profile and chosen SAML v2.0 bindings for the request and response.

The RealMe Assertion Service messaging is depicted in Figure 2. In this scenario the Client SP requests an identity assertion via sending an AuthnRequest to the RealMe Assertion Service. The AuthnRequest message is chosen to initiate the assertion messaging as it reuses the SAML 2.0 standard with a proven security model in a parallel messaging profile as the Logon Service. The reuse of the SAML v2.0 messaging profile for logon and identity assertions provides the ability to reuse in messaging components between the Logon and Assertion services and simplified integration over the services for the Clients.

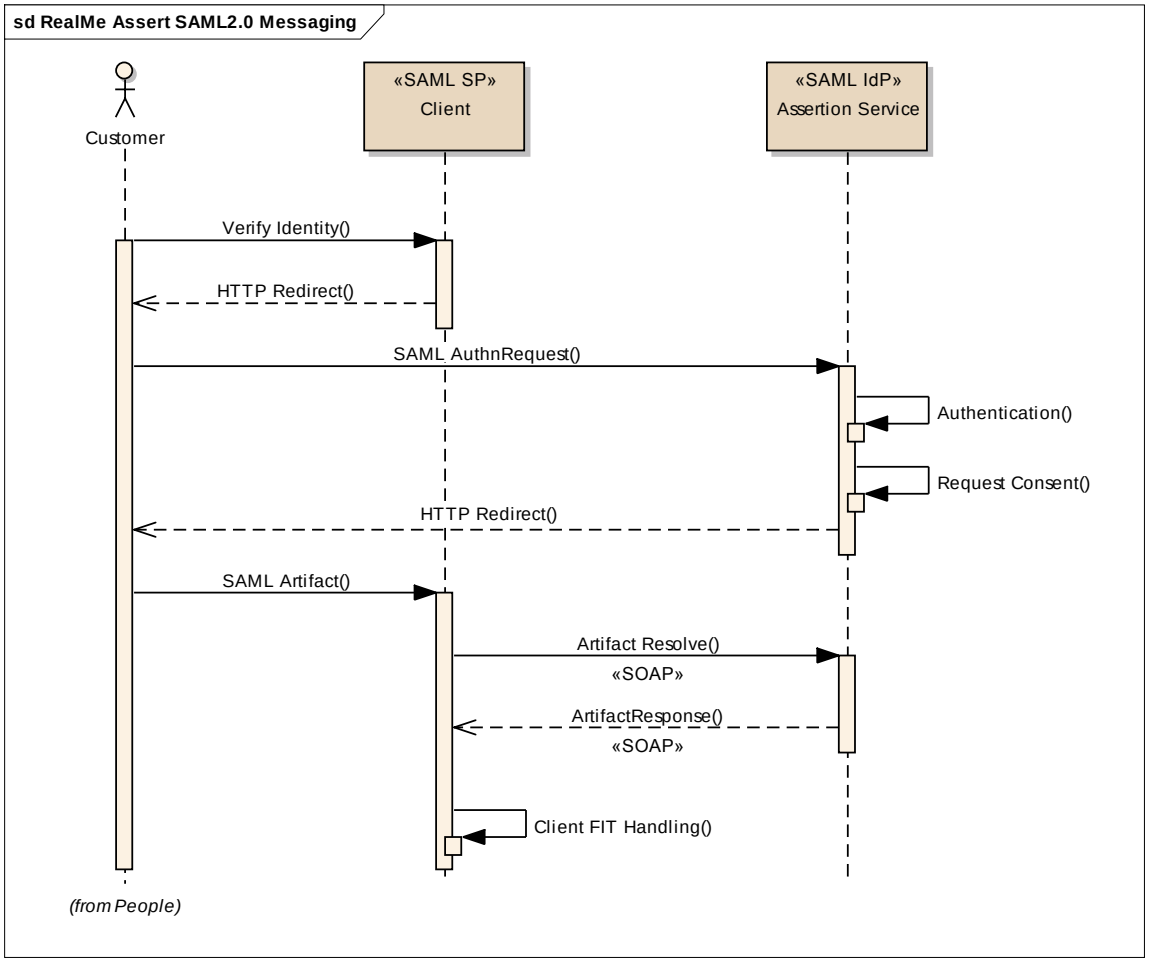

Figure 2 - Assert SAML v2.0 Flow

Message	Description
1. Verify Identity	A Customer interacts with a Client site to require an assertion of their identity information.
2. HTTP Redirect	The Client redirects the Customer's user-agent to the RealMe Assertion Service IdP. The redirect contains a Client AuthnRequest.
3. SAML AuthnRequest	The user-agent responds to [2] to send a Client AuthnRequest to the RealMe Assertion Service IdP.
4. Authentication	RealMe Assertion Service IdP presents system functionality to the Customer for authentication. Included in this step are client side redirects to the user agent that are RealMe design specific and are extraneous to RealMe SAML v2.0 messaging.
5. Request Consent	This step may encapsulate a set of tasks on RealMe Assertion Service side to seek Customer consent to release identity attributes to the Client. Included in this step may also be client side redirects to the user-agent that are RealMe design specific and SHOULD be considered as extraneous to RealMe SAML v2.0 messaging.
6. Redirect	RealMe Assertion Service IdP redirects the user-agent to the Client's "assertion consuming service". The redirect contains the SAML v2.0 artifact.
7. RealMe SAML Artifact	The user-agent responds to [6] to send a SAML v2.0 artifact to the Client's "assertion consuming service".
8. RealMe Artifact Resolve	The Client SP sends a request to RealMe Assertion Service IdP's "artifact resolution service". This request includes the value of the SAML v2.0 artifact received in [7]. In this message the Client is asking the RealMe Assertion Service IdP to "resolve the artifact" into a SAML v2.0 response. The protocol for this request message is SOAP and is independent of the user-agent.
9. RealMe Artifact Response	RealMe Assertion Service IdP responds to the Client's artifact resolution request [8] over SOAP. The artifact response contains the SAML v2.0 Response which contains the SAML v2.0 Assertion message element. The SAML v2.0 Assertion contains the identity attributes conveyed from IAP's as SAML v2.0 attribute statements. As the protocol for this response message is SOAP, transmission of the SAML v2.0 Assertion does not occur via the user-agent.
10. Client identity attributes	The Client SP obtains the identity attributes from the SAML v2.0 Assertion's attribute statements. The Client invokes its own handling as required.

Table 6 – Assertion Service SAML v2.0 Flow

3 The SAML v2.0 Request

The SAML v2.0 Request contains a message from a SP on behalf of a Customer that is requesting an SAML v2.0 Assertion at the IdP.

3.1 Protocol

For the 'Web Browser SSO Profile' the SAML v2.0 request protocol is an **<AuthnRequest>** message. This is described in [saml-core-2.0-os] section 3.4 and constrained by NZ SAMS in section 10.

3.2 Message Elements

3.2.1 Element <AuthnRequest>

This message will contain the following elements and attributes.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 3.5.	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] line 1467.
Version	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 3.5.	Per OASIS. ☑ SAML v2.0	MUST be provided. The identifier is "2.0". Ref [saml-core-2.0-os] line 1471.
IssueInstant	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • If not within the accepted timeframe as per the tolerances in the messaging introduction pre-requisites will result in RealMe returning a status code of ***:RequestDenied** as specified in section 4.5. • Invalid XML SHALL result in the handling as specified in section 3.5.	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] line 1474.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
Destination	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 3.5.	Per OASIS. ☑ SAML v2.0	MUST be provided for the HTTP-Redirect and HTTP-POST bindings in conjunction with signing. Ref [saml-bindings-2.0-os] lines 661,843.
ForceAuthn	MAY be provided. If provided will be ignored. RealMe Assertion Service authentication behaviour SHALL NOT be able to be specified by the SP and will be independent of this request parameter. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section **3.5**. Constraint on NZ SAMS: Value ignored.	Per OASIS. ☑ SAML v2.0	MAY be provided. If not passed then will be defaulted to the XML data-type of false. Ref [saml-core-2.0-os] line 2042.
IsPassive	MAY be provided. If provided MUST be set to the XML data-type of false. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • If provided and set to the XML data-type of true will result in RealMe returning a status code of ***:NoPassive** as specified in section 4.5. • Invalid XML SHALL result in the handling as specified in section 3.5. Constraint on NZ SAMS: IsPassive not supported.	Per OASIS. NZ SAMS line 465. ☑ SAML v2.0	MAY be provided. If not passed then the XML data-type of false will be used. Ref [saml-core-2.0-os] line 2047.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
Assertion Consumer Service Index	RECOMMENDED. This attribute MUST be provided if 'ProtocolBinding' and 'AssertionConsumer ServiceURL' attributes are not provided. In this instance the mechanism defined in the SAML metadata SHALL be used. The SP's metadata is used to locate an <AssertionConsumerService> with a matching index value. Ref [saml-profiles-2.0-os] section 4.1.6. If an 'AssertionConsumerServiceIndex' is sent with a value that does not match a published <AssertionConsumerService> index in the SP's metadata then another <AssertionConsumerService> SHALL be used by RealMe. See section 7.1.5 for rules on determining the matching index value supplied in the SP metadata. The SP's <AssertionConsumerService> used by RealMe MUST be over the HTTP-Artifact binding. A SP MUST not use an index value to cause RealMe to use any other binding. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • If 'AssertionConsumerServiceIndex' is not provided and 'Protocol Binding' and 'AssertionConsumerServiceURL' attributes are not provided it will result in RealMe returning a status code of ***:RequestUnsupported** as specified in section 4.5. • If the SP metadata specifies an <AssertionConsumerService> that is an invalid URL then standard browser error behaviour will occur. The SAML v2.0 exchange will not occur. • Invalid XML SHALL result in the handling as specified in section 3.5.	RECOMMENDED Ref NZ SAMS line 446. ☑ SAML v2.0	MAY be provided. This indirectly identifies the location to which the <Response> message SHOULD be returned to the requestor. Ref [saml-core-2.0-os] line 2051.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
Protocol Binding	MAY be provided. If populated it MUST indicate HTTP-Artifact Binding by containing the value `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact` This method of specifying the response binding SHOULD NOT be preferred over the 'AssertionConsumerServiceIndex'. If a valid ProtocolBinding is sent without an AssertionConsumerServiceURL value then RealMe SHALL choose a matching <AssertionConsumerService> from the SP's metadata. ☑ NZ SAMS	Per OASIS. NZ SAMS line 478. ☑ SAML v2.0	MAY be provided. A URI reference that identifies a SAML protocol binding to be used when returning the <Response> message. MUST NOT be provided if 'AssertionConsumer ServiceIndex' is provided. Ref [saml-core-2.0-os] line 2068.
	Exceptions SHALL result for the following conditions: • If populated and contains a value other than ***:HTTP-Artifact** will result in RealMe returning a status code of ***:RequestUnsupported** as specified in section 4.5. • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: HTTP-Artifact only.		
Assertion Consumer Service URL	MAY be provided. If this attribute is populated it MUST be accompanied by a ProtocolBinding attribute containing the value `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact` The user's browser will be redirected to any URL supplied. This method of specifying the response binding SHOULD NOT be preferred over the 'AssertionConsumerServiceIndex'. ☑ NZ SAMS	MAY be provided. NZ SAMS line 472. ☑ SAML v2.0	MAY be provided. Specifies by value the location to which the <Response> message MUST be returned to the requester. MUST NOT be provided if 'AssertionConsumer ServiceIndex' is

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	Exceptions SHALL result for the following conditions: • If both 'AssertionConsumerServiceURL' and 'AssertionConsumerServiceIndex' are provided then RealMe will respond with a status code of ***:RequestUnsupported** as specified in section 4.5. • If the attribute is populated with an invalid URL then standard browser error behaviour will occur. The SAML v2.0 exchange will not occur. • Invalid XML SHALL result in the handling as specified in section 3.5.		provided. Typically accompanied by the ProtocolBinding attribute. Ref [saml-core-2.0-os] line 2061.
	Constraint on NZ SAMS: Only the HTTP-Artifact binding is supported.		
Provider Name	MAY be provided. If provided then it will be ignored. ☑ NZ SAMS	MAY be provided. Can be used to communicate branding information in selected cases across igovt. NZ SAMS line 479, 489. ☑ SAML v2.0	MAY be provided. Ref [saml-core-2.0-os] line 2080.
	Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: Value ignored.		
<Issuer>	MUST be provided. It is REQUIRED that this is in the format of an identity privacy domain. See section 4.2.2. The Issuer SHALL be used by RealMe to trigger SP specific co-branding functionality at the Logon Service. ☑ NZ SAMS	Per OASIS. NZ SAMS line 454. ☑ SAML v2.0	MUST be provided. Ref [saml-profiles-2.0-os] line 516.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	Exceptions SHALL result for the following conditions: • If the <Issuer> is not in the format of an identity privacy domain then it will cause RealMe to respond with a status code ***:RequestUnsupported** as specified in section 4.5. • If the <Issuer> is not known then it will cause RealMe to display an error page to the end user. The SAML v2.0 exchange will not occur. • Invalid XML SHALL result in the handling as specified in section 3.5. This includes the circumstance where the element is not provided.		
	Constraint on NZ SAMS: Format as per section 7.		
<NameID Policy>	MAY be provided. If provided then restrictions SHALL apply on the child elements as in section 3.2.3. ☑ NZ SAMS	RECOMMENDED to be provided. NZ SAMS line 456 ☑ SAML v2.0	MAY be provided. Ref [saml-core-2.0-os] line 2025.
	Exceptions SHALL result for the following conditions: • If not provided will result in RealMe returning a status code of ***:RequestUnsupported** specified in section 4.5. • Violation of restrictions in child elements as in section 3.2.3. • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: If provided then restrictions SHALL apply on the child elements as in section 3.2.4.		
<Requested Authn Context>	MAY be provided. If provided see restrictions in child elements as in section 3.2.4. ☑ NZ SAMS	MAY be provided. NZ SAMS line 464. ☑ SAML v2.0	MAY be provided. Ref [saml-core-2.0-os] line 2034.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	Exceptions SHALL result for the following conditions: • If provided and violates restrictions in child elements as in section 3.2.4 then it SHALL result in the handling as specified in 3.2.4. • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: Restrictions on child elements.		

Table 7 - <AuthnRequest> Elements

3.2.2 Element <Issuer>

This element MUST be provided to identify the calling Service Provider. Conventionally in SAML v2.0 this is the URL of the SP SAML provider. The value is equal to the entityID of the Service Provider's SAML v2.0 Metadata.

It is REQUIRED that the format is of an identity a privacy domain by the following pattern:

[protocol]://[client-domain]/[privacy-context-name]/[service-name]{-client-environment}

e.g. `https://www.sample-client.co.nz/onlineservices/service1`

Section 6 defines the concept of identity privacy domains.

The Issuer SHALL be used by RealMe to trigger SP specific co-branding functionality.

3.2.3 Element <NameIDPolicy >

This element MAY be present to describe the policy to be used in returning and creating a nameID in the response's Assertion. The RealMe Assertion Service does not generate federated identifiers directly, these are the responsibility of the Identity Attribute Provider systems. However most SAML products assume the presence of this element since it is fundamental to normal SAML v2.0 messaging to achieve federation goals. Therefore to support interoperability over a range of SAML products the Assertion will return a Subject statement with a 'transient' nameID format. The value of the nameID in the response's Assertion SHOULD be ignored by local processing of the receiving Client SP.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
AllowCreate	MAY be provided. If provided will be ignored. RealMe Assertion Service authentication behaviour SHALL NOT be able to be specified by the SP and will be independent of this request parameter. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 3.5. Constraint on NZ SAMS: Value ignored.	RECOMMENDED to be provided and MUST be set to the XML data-type of true. NZ SAMS line 456. ☑ SAML v2.0	MAY be provided. A Boolean value used to indicate whether the identity provider is allowed, in the course of fulfilling the request, to create a new identifier to represent the principal. Defaults to "false". Ref [saml-core-2.0-os] line 2123.

Format	MUST be provided.
	MUST be either :
	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
	or
	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
	RealMe SHALL always supply transient nameID's in the assertion. The unspecified setting allows the IdP to determine the format (Ref [saml-core-2.0-os] line 3281).
	☑ NZ SAMS
	Exceptions SHALL result for the following conditions:
	• If not set to one of the specified values then it will result in RealMe returning a status code of ***:RequestUnsupported** as specified in section 4.5.
	• Invalid XML SHALL result in the handling as specified in section 3.5.
		MUST be provided.
		NZ SAMS lines 53, 873, 876 narrows to allow only the two stated values as used by RealMe.
		☑ SAML v2.0
			MAY be provided. Ref [saml-core-2.0-os] line 2113.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
SPName Qualifier	MAY be provided. RealMe SHALL return the subject's nameID in the namespace of the requester's identity privacy domain. An SP MAY not specify another namespace other than the SP's identity privacy domain. If provided it MUST be equal to the value in the <Issuer>. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MAY be provided. Optionally specifies that the assertion subject's identifier be returned (or created) in the namespace of a service provider other than the requester, or in the namespace of an affiliation group of service providers. Ref [saml-core-2.0-os] line 2118.
	Exceptions SHALL result for the following conditions: • If provided and is not equal to the value in the <Issuer> then it will result in RealMe returning a status code of ***: RequestDenied** as specified in section 4.5. • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: Must be the same value as the <Issuer>.		

Table 8 - <NameIDPolicy> elements

3.2.4 Element <RequestedAuthnContext>

This element MAY be sent by the SP to the RealMe Assertion Service, but will not affect RealMe behaviour. The RealMe Assertion Service SHALL transfer the user to the RealMe Logon Service for authentication with an authentication type that RealMe will determine. RealMe Assertion Service to RealMe Logon Service authentication type SHALL NOT be able to be specified by the SP and will be independent of this request parameter.

The <RequestedAuthnContext> is accepted in so far to support for SAML providers with behaviour that treat this element as mandatory. If the <RequestedAuthnContext> element is present the following child elements SHALL be required.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<Authn Context ClassRef>	MUST be provided. The URI reference MUST be of the specified set below in Table 10. This specification constitutes the out of band communication by RealMe IdP.	MAY be provided but MUST be communicated out of band by the IdP. NZ SAMS line	Either <AuthnContext ClassRef> or <AuthnContext DeclRef> MUST be provided. Ref [saml-core-2.0-os] line 1808.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	☑ NZ SAMS	863. ☑ SAML v2.0	
	Exceptions SHALL result for the following conditions: • If not provided RealMe will return a status code of ***:NoAuthnContext** as specified in section 4.5. • If provided but not in the specified set of values will result in RealMe returning a status code of ***:RequestUnsupported** as specified in section 4.5. • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: Must be provided and set to ***:ModStrength**		
<Authn Context DeclRef>	MUST NOT be provided. ☑ NZ SAMS	MUST NOT be used. NZ SAMS lines 862, 891. ☑ SAML v2.0	Either <AuthnContext ClassRef> or <AuthnContext DeclRef> MUST be provided. Ref [saml-core-2.0-os] line 1808.
	Exceptions SHALL result for the following conditions: • If provided will result in RealMe returning a status code of ***:RequestUnsupported** as specified in section 4.5. • Invalid XML SHALL result in the handling as specified in section 3.5.		
Comparison	MAY be provided. If provided then it will be ignored and will not affect RealMe behaviour. ☑ NZ SAMS	Per OASIS. NZ SAMS line 861. ☑ SAML v2.0	MAY be provided. Specifies the comparison method used to evaluate the requested context classes or statements, one of ‘exact’, ‘minimum’, ‘maximum’, or ‘better’. The default is ‘exact’. Ref [saml-core-2.0-os] line 1812.
	Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 3.5.		
	Constraint on NZ SAMS: Value ignored.		

Table 9 - <RequestedAuthnContext> specialisations for the Assertion Service

The supported set of AuthnContextClassRef values will be:

AuthnContextClassRef
urn:nzl:govt:ict:stds:authn:deployment:GLS:SAML:2.0:ac:classes:ModStrength

Table 10 – Supported AuthnContextClassRef values for the Assertion Service

3.3 Binding

In accordance with NZ SAMS Section 6.4, the HTTP-Redirect Binding will be used to send the <AuthnRequest>. This binding will send the SAML v2.0 request encoded as a URL parameter.

The HTTP-Redirect Binding is conventionally used in SAML v2.0 to send small amounts of data as a URL parameter.

The RealMe Logon and Assertion Service IdPs SHALL expose an HTTP-Redirect Binding in its metadata that will be over HTTPS.

The specifics of the encoding are identified in section 11 of NZ SAMS and 3.4.4.1 of [saml-bindings-2.0-os]. These specify that the contents of the URL parameter are compressed using the DEFLATE compression mechanism, as specified in [RFC1951] and then Base64 and URL encoded. As the SAML <AuthnRequest> is a small message this binding is well suited. The ability to compress the message also has the advantage of improving network performance.

URL Parameter	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
SAMLRequest	MUST be provided. ☑ NZ SAMS Exceptions SHALL result in RealMe displaying an error page to the end user and the SAML v2.0 exchange will not occur.	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-binding-2.0-os] line 587 Contains the compressed [RFC1951], Base64, URL encoded SAML Request as per 3.4.4.1 of [saml-bindings-2.0-os].

URL Parameter	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
RelayState	MAY be provided. Note that the RelayState is a mechanism available to the SP to broker the chain-of-trust with RealMe. The intended use of RelayState is reserved in the SAML specification for the SP private use, and as such is deemed an appropriate mechanism for the SP to implement such brokering functionality. This is consistent with comments in NZ SAMS line 393. Section 4.2.1 specifies that the value of this parameter will be returned in the send of the artifact to the requestor. ☑ NZ SAMS Exceptions SHALL result in RealMe displaying an error page to the end user and the SAML v2.0 exchange will not occur.	Per OASIS. NZ SAMS 390. ☑ SAML v2.0	MAY be provided. If provided it MUST be URL encoded. Ref [saml-binding -2.0-os] line 590. As per 3.4.3 of [saml-binding -2.0-os] the value MUST NOT exceed 80 bytes in length and SHOULD be integrity protected by the entity creating the message independent of any other protections that may or may not exist during message transmission. If passed then it SHALL be returned by the responder.
SigAlg	MUST be provided. ☑ NZ SAMS Exceptions SHALL result in RealMe displaying an error page to the end user and the SAML v2.0 exchange will not occur.	Per OASIS. ☑ SAML v2.0	MUST be provided since <AuthnRequest> SHALL be signed. The value of this parameter MUST be a URI that identifies the algorithm used to sign the URL-encoded SAML protocol message. Ref 3.4.4.1 of [saml-bindings-2.0-os] line 597 for accepted values.
Signature	MUST be provided. ☑ NZ SAMS Exceptions SHALL result in RealMe displaying an error page to the end user and the	Per OASIS. ☑ SAML v2.0	MUST be provided since <AuthnRequest> SHALL be signed. Ref 3.4.4.1 of [saml-bindings-2.0-os] line 608.

URL Parameter	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	SAML v2.0 exchange will not occur.		

Table 11 – HTTP-Redirect Binding parameters

3.4 Signing

RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<AuthnRequest> MUST be signed to: 1. Verify message integrity. 2. Authenticate the request came from the expected party. RealMe SHALL not issue a SAML v2.0 response to an incorrectly signed <AuthnRequest>. ☑ NZ SAMS	Per OASIS. NZ SAMS Section 6.4.2. ☑ SAML v2.0	<AuthnRequest> MAY be signed. Ref 4.1.3.3 of [saml-profiles-2.0-os] line 469.
Exceptions SHALL result in RealMe displaying an error page to the end user and the SAML v2.0 exchange will not occur.		
Constraint on NZ SAMS: <AuthnRequest> signing enforced.		

Table 12 – <AuthnRequest> Signing Requirements

3.5 Handling of Invalid <AuthnRequest> XML

The <AuthnRequest> is a XML element bound to the OASIS SAML v2.0 XML schema. An <AuthnRequest> sent by a SP must conform to the XML schema. If any XML schema exceptions exist it SHALL result in RealMe displaying an error page to the end user. No SAML v2.0 response will be returned to the SP as the request is not syntactically valid XML.

3.6 Sample Request

The following is a sample URL that contains a SAML request containing an <AuthnRequest> sent over the HTTP-Redirect Binding.

```
https://realme.govt.nz/sso/SSORedirect/metaAlias/assert-idp?

?SAMLRequest=fZJbb9swDIX%2FiqF339tsEeIARoICAdatiIc%2B7E2VmUaALGkinUt%2F%2FSQvKQJ066vIo%2F
MdkgsUg3a8HWlvtvB7BKTkNGiDPBYaNnrDrUCF3IgBkJPkXfv4jVdZwQUieFLWsBuJ%2B1zjvCUrrWZJe1WvrMFxA
N%2BBPygJG9PDqWEFS9YBRhkRexq2J3LI81z1YEjROVPZqz1QZt5yRJt33Y8t9MqDpHwAEq1WAnN1wFT1jiWbdcPE
%2FP6rqAoo7ueymvWl3L18qctZ%2FVKKmbzbzUMX4hj8kYShhlVFMU%2FLMi3qn0XB6zmvq18sefL2EBj89xCtYZ0
YnIbkgv5eZMkzeJy4Q2i2XMTJ8Ol7v7wmOR6PGU76FN17GA9CD3nsF86Vi%2FxWufi7rGi9WT9ZreQ5abW2x1VQUc
AhPwJLHqwfBH2%2Bh%2Fii%2BnQ3tXIXaZHCZANr%2FtHl6ny5EOinewmLIzjRJd3t00qH29jCbhkRzJvmMR1Xkjh
Sj1zEVt6D0%2FY8BFO%2Bee5uzkpyGfWB%2BNH2HXkwr7S%2FTOKfLlfk%2F%2BDlH298%2BQc%3D

&RelayState=SzQzTjK0NDCxMDVITTY2N0gxNTAwMTFPMzROS0o0MDAAAA%253D%253D

&SigAlg=http%3A%2F%2Fwww.w3.org%2F2000%2F09%2Fxmldsig%23rsa-sha1

&Signature=VwZiHU0VDHcklgo83dmTnF0DZXnHQvVBkC3hQxTmXg9HLtgneehwgrwp3pthgegNgKBiGMLYMpWN8F
P%2BsbkaPPoUOZHnBXnpUDAPj%2F2vvBNlhd0z2GrED%2Fi2K54%2FycbwA0rH%2BlTOKl6OQUXZ2PGHPwEQ14LPM
spmpSnCEoLTl9M%3D
```

The following is a sample decoded <AuthnRequest>:

```
<samlp:AuthnRequest xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  AssertionConsumerServiceIndex="0"
  Destination="https://realme.govt.nz/sso/SSORedirect/metaAlias/assert-idp"
  ID="a958a20e059c26d1cfb73163b1a6c4f9"
  IssueInstant="2012-05-21T00:39:32Z"
  ProviderName="Sample Service Provider"
  Version="2.0">
  <saml:Issuer>https://www.sample-client.co.nz/onlineservices/service1</saml:Issuer>
  <samlp:NameIDPolicy AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
  </samlp:NameIDPolicy>
  <samlp:RequestedAuthnContext>
    <saml:AuthnContextClassRef>
      urn:nzl:govt:ict:stds:authn:deployment:GLS:SAML:2.0:ac:classes:ModStrength
    </saml:AuthnContextClassRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```

4 The SAML v2.0 Response

The response will complete the SAML v2.0 transaction by the IdP sending a message to the SP that provides statements about the Customer's logon or identity. This section defines the SAML v2.0 messaging for the response, constrained by NZ SAMS.

4.1 Binding

In accordance with NZ SAMS Section 6.4, the HTTP-Artifact Binding will be used to initiate the SAML v2.0 response to the SP.

In the HTTP Artifact binding the SAML v2.0 response is transmitted by reference using a small stand-in called an artifact. A separate, synchronous binding, such as the SAML v2.0 SOAP binding, is used to exchange the artifact for the actual protocol message using the artifact resolution protocol defined in the SAML assertions and protocols specification [saml-core-2.0-os].

The mechanics of this binding are described in full in [saml-bindings-2.0-os] section 3.6 and constrained by NZ SAMS section 11.

The SP metadata MUST specify a location for this binding in its set of AssertionConsumerServices. This binding MUST be over HTTPS.

An example <AssertionConsumerService> in the SP metadata is provided below. This is extracted from Appendix A: Sample Service Provider Metadata where a complete sample is provided.

```
<md:AssertionConsumerService
  Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
  Location="https://www.sample-client.co.nz/sso/ACS"
  index="0"
  isDefault="false">
</md:AssertionConsumerService>
```

In the above sample the 'index' attribute allows the SP to make use of the AssertionConsumerServiceIndex attribute in the SAML <AuthnRequest>. This was as defined in section 3.2.1 and exampled in section 3.6.

4.2 Message Elements

In accordance with NZ SAMS section 6.4.2, the SAML v2.0 Response will be conveyed by the use of the HTTP-Artifact Binding in conjunction with the Artifact Resolution Profile.

The HTTP-Artifact Binding will be used to pass the artifact value to the SP. The SP will use the Artifact Resolution Profile to resolve the artifact into a SAML v2.0 assertion by a SOAP call. The mechanics of this profile is described in [saml-profiles-2.0-os] section 5.

The HTTP-Artifact Binding in conjunction with the Artifact Resolution Profile specifies that there are three types of messages involved to support this binding.

1. The message from the IdP to the SP's 'assertion consuming service' containing the artifact.
2. The message from the SP to the IdP to request a resolution of the artifact.
3. The message from the IdP to the SP containing the artifact response, which contains the SAML v2.0 response that includes the SAML v2.0 assertion message element.

These correspond to steps: 7, 8, and 9 respectively in section 2.

4.2.1 IdP to SP : Redirect to Assertion Consuming Service

This message will be as per [saml-bindings-2.0-os] section 3.6.3 and constrained by NZ SAMS section 11.

The URL encoding will be used to send the artifact via an HTTP GET over SSL.

<table>
<tr><th>URL Parameter</th><th>RealMe Requirement</th><th>NZ SAMS 1.0 Requirement</th><th>OASIS SAML v2.0 Requirement</th></tr>
<tr><td>SAMLart</td><td>MUST be provided.

☑ NZ SAMS</td><td>Per OASIS.
NZ SAMS line 433.

☑ SAML v2.0</td><td>MUST be provided.
See section 3.6.4 of [saml-binding-2.0-os].
SAML v2.0 defines an artifact type is defined as follows:
SAML_artifact := base64(TypeCode EndpointIndex RemainingArtifact)
Where :
<table>
<tr><td>TypeCode</td><td>0x0004</td></tr>
<tr><td>EndpointIndex</td><td>2 byte sequence</td></tr>
<tr><td>RemainingArtifact</td><td>SourceID MessageHandle</td></tr>
<tr><td>SourceID</td><td>20-byte_sequence.</td></tr>
<tr><td>MessageHandle</td><td>20-byte_sequence</td></tr>
</table>
SourceID is a 20-byte sequence used by the artifact receiver to determine artifact issuer identity and the set of possible resolution endpoints. This is SHA1 hash of RealMe IdP entityID.

On receiving the SAML v2.0 artifact, the</td></tr>
</table>

URL Parameter	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
			artifact receiver determines if the SourceID belongs to a known artifact issuer and obtains the location of the SAML responder using the EndpointIndex before sending a SAML v2.0 ArtifactResolve message to it. The EndpointIndex maps to the index of the ArtifactResolutionService in the IdP metadata.
RelayState	MAY be provided. ☑ NZ SAMS	Per OASIS. NZ SAMS line 390. ☑ SAML v2.0	MAY be provided. Ref [saml-binding-2.0-os] line 1012. It SHALL be provided if the requestor passes in the parameter in the request as specified in section 3.3. If passed then it MUST be the same value as supplied by the requestor. As per 3.6.3.1 of [saml-binding-2.0-os] the value MUST NOT exceed 80 bytes in length and SHOULD be integrity protected by the entity creating the message independent of any other protections that may or may not exist during message transmission.

Table 13 – HTTP-Artifact Binding parameters

A sample message is as follows:

```
HTTP/1.1 302 Object Moved

Location:https://www.sample-client.co.nz/sso/ACS
?SAMLart=AAQAABWFTOPhANZhf21nl9DmXsAkiSM0ocx7GdxUfXFttmS954%2BP6Vb01I0%3D

&RelayState=SzQzTjK0NDCxMDVITTY2N0gxNTAwMTFPMzROS0o0MDAAAA%253D%253D
```

4.2.2 SP to IdP : <ArtifactResolve>

This message will be as per [saml-profile-2.0-os] section 5.3.1 and constrained by NZ SAMS section 10. See also [saml-core-2.0-os] section 3.5.1, which is constrained by NZ SAMS section 12.

Within the SOAP body the following elements and attributes are expected:

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<ArtifactResolve>	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 4.2.2.1	Per OASIS. NZ SAMS line 584. ☑ SAML v2.0	MUST be provided by [saml-profiles-2.0-os] section 5.3, line 1467.

Table 14 –Artifact Resolve SOAP body elements

Within the <ArtifactResolve> the following elements and attributes are expected:

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 4.2.2.1	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] line 2349, 1467.
Version	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 4.2.2.1	Per OASIS. ☑ SAML v2.0	MUST be provided. The identifier is “2.0”. Ref [saml-core-2.0-os] line 2349, 1471.
IssueInstant	MUST be provided. SHALL not affect RealMe behaviour. The tolerances in the messaging introduction pre-requisites are not applicable for this message. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • Invalid XML SHALL result in the handling as specified in section 4.2.2.1	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] line 2349, 1474.
<Issuer>	MUST be provided. The <Issuer> MUST be an integrated RealMe SP. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • If the <Issuer> is not an integrated SP a HTTP error	Per OASIS. ☑ SAML v2.0	MUST be provided. Required for SAML products to identify the IdP. Ref [saml-profiles-2.0-os] section 5.4.1, line 1488.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	response SHALL be returned and the artifact SHALL NOT be resolved. • Invalid XML SHALL result in the handling as specified in section 4.2.2.1		
<Artifact>	MUST be provided. ☑ NZ SAMS Exceptions SHALL result for the following conditions: • If the artifact has expired or does not exist RealMe will respond with an <ArtifactResponse> element with no embedded message. • Invalid XML SHALL result in the handling as specified in section 4.2.2.1	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] section 3.5.1.

Table 15 – Artifact Resolve elements

4.2.2.1 <ArtifactResolve> Error Handling

The <ArtifactResolve> is a XML element bound to the OASIS SAML v2.0 XML schema. An <ArtifactResolve > and its contents sent by a SP must conform to the XML schema. If any XML schema exceptions exist it will result in an HTTP error response and the artifact not being resolved.

4.2.2.2 Sample <ArtifactResolve> Request

A sample message is as follows :

```
POST /sso/ArtifactResolver/metaAlias/assert-idp
HTTP/1.1
Host:as.realme.govt.nz
Content-Type: text/xml
Content-Length: nnn
SOAPAction:http://www.oasis-open.org/committees/security

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <samlp:ArtifactResolve
      xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
      xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
      ID="_6c3a4f8b9c2d"
```

```
      Version="2.0"
      IssueInstant="2012-05-21T00:39:45Z">
      <Issuer>https://www.sample-client.co.nz/onlineservices/service1</Issuer>
      <Artifact>
        AAQAABWFTOPhANZhf21nl9DmXsAkiSM0ocx7GdxUfXFttmS954 BP6Vb01I0=
      </Artifact>
    </samlp:ArtifactResolve>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2.3 IdP to SP : <ArtifactResponse>

This message will be as per [saml-profile-2.0-os] section 5.3.1 and constrained by NZ SAMS section 10. See also [saml-core-2.0-os] section 3.5.2, which is constrained by NZ SAMS section 12.

If an artifact sent by a SP is successfully resolved[1] a SAML response containing an assertion will be returned that is encapsulated in the <ArtifactResponse> element.

Within the SOAP body the following elements and attributes are expected:

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<ArtifactResponse>	SHALL be provided. ☑ NZ SAMS	Per OASIS. NZ SAMS line 591. ☑ SAML v2.0	MUST be provided. Ref [saml-profiles-2.0-os] section 5.3, line 1477.

Table 16 – Artifact Response SOAP body elements

Within the ArtifactResponse the following elements and attributes are expected:

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	SHALL be provided.	Per OASIS.	MUST be provided. Ref [saml-core-2.0-os]

1 Successful artifact resolution is where the IdP successfully matches the sent artifact in the <ArtifactResolve> to an artifact in its local store. It will result in a SAML v2.0 response that may either contain a success response or an error response.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	☑ NZ SAMS	☑ SAML v2.0	lines 2370, 1539.
InResponseTo	SHALL be provided. ☑ NZ SAMS	Per OASIS. NZ SAMS line 534. ☑ SAML v2.0	MUST be provided. In response to a request and MUST match the value of the request's ID attribute. Ref [saml-core-2.0-os] lines 2370, 1542.
Version	SHALL be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MUST be provided. The identifier is "2.0". Ref [saml-core-2.0-os] lines 2370, 1548.
IssueInstant	SHALL be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] lines 2370, 1551.
<Issuer>	SHALL be provided. ☑ NZ SAMS	Per OASIS. NZ SAMS line 503. ☑ SAML v2.0	MUST be provided. Ref [saml-profiles-2.0-os] line 1495.
<Status>	SHALL be provided. ☑ NZ SAMS Constraint on NZ SAMS: Status is provided.	SHOULD be provided. NZ SAMS line 840. ☑ SAML v2.0	MUST be provided. Ref [saml-core-2.0-os] lines 2370, 1578.
<Response>	SHALL be provided to pass back the SAML assertion to the SP. ☑ NZ SAMS	Per OASIS. NZ SAMS line 503. ☑ SAML v2.0	MUST be provided. Ref [saml-profiles-2.0-os] section 4.1.2, line 420.

Table 17 – Artifact Response elements

The central element contained inside the <ArtifactResponse> is the SAML response message containing the assertion. This message is a complex structure and will be supplied strictly as per the SAML v2.0 specification.

See [saml-core-2.0-os] section 3.2.2 for the structure of a <Response> element.

See [saml-core-2.0-os] section 2.3 for the structure of the nested <Assertion> element.

A sample message is as follows:

```
HTTP/1.1 200 OK
Date: 09 Oct 2007 09:00:49 GMT
Content-Type: text/xml
Content-Length: nnnn

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <samlp:ArtifactResponse xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
      xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
      ID="_FQvGknDfws2Z"
      Version="2.0"
      InResponseTo="_6c3a4f8b9c2d"
      IssueInstant="2012-05-21T00:39:45Z">
      <Issuer>https://realme.govt.nz/realme/assert-idp</Issuer>
      <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
      </samlp:Status>
      <samlp:Response xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
        xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
        Destination="https://www.sample-client.co.nz/sso/ACS"
        ID="f9a6d769fb6920cc986a855cbd771960"
        InResponseTo="a958a20e059c26d1cfb73163b1a6c4f9"
        IssueInstant="2012-05-21T00:39:45Z"
        Version="2.0">
        <saml:Issuer>https://realme.govt.nz/realme/assert-idp</saml:Issuer>
        <samlp:Status>
          <samlp:StatusCode
            Value="urn:oasis:names:tc:SAML:2.0:status:Success">
          </samlp:StatusCode>
        </samlp:Status>
        <saml:Assertion ID="d31aefd7f40818a0bec68a79779a397f"
```

```
          IssueInstant="2012-05-21T00:39:45Z"
          Version="2.0">
          <saml:Issuer>https://realme.govt.nz/realme/assert-idp</saml: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:CanonicalizationMethod>
              <ds:SignatureMethod
                Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1">
              </ds:SignatureMethod>
              <ds:Reference URI="#f6420eb02d42fc7a10b230d91ed620d7">
                <ds:Transforms>
                  <ds:Transform
                    Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature">
                  </ds:Transform>
                  <ds:Transform
                    Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                  </ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod
                  Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
                </ds:DigestMethod>
                <ds:DigestValue>n+0V5Z2pswNK0Om2bWZiDruONyo=</ds:DigestValue>
              </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>
QgedySIgImhS/UZ+oibEslOwt+5iAPNsohjA+11nKFuNnb/yJIcXjVfBFUbF9xAn1lrRUfS9m7Ya
YNqS1k9ehoa4qQmjILhAkXU5jDY1Rj8yMWJuraIM58Tb9M8/LdW963MDEbiv+xwPAxEJO9CQ6u1c
x197iGvC5i3yD/HEeoY=
            </ds:SignatureValue>
          </ds:Signature>
          <saml:Subject>
            <saml:NameID
              Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
              NameQualifier="https://realme.govt.nz/realme/assert-idp"
              SPNameQualifier="https://www.sample-client.co.nz/onlineservices/service1">
              d31aefd7f40818a0bec68a79779a397f
            </saml:NameID>
            <saml:SubjectConfirmation
              Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
              <saml:SubjectConfirmationData
                InResponseTo="a958a20e059c26d1cfb73163b1a6c4f9"
                NotOnOrAfter="2012-05-21T00:49:45Z"
                Recipient="https://www.sample-client.co.nz/sso/ACS">
              </saml:SubjectConfirmationData>
```

```
            </saml:SubjectConfirmation>
          </saml:Subject>
          <saml:Conditions NotBefore="2012-05-21T00:29:45Z"
            NotOnOrAfter="2012-05-21T00:49:45Z">
            <saml:AudienceRestriction>
              <saml:Audience>
                https://www.sample-client.co.nz/onlineservices/service1
              </saml:Audience>
            </saml:AudienceRestriction>
          </saml:Conditions>
          <saml:AuthnStatement AuthnInstant="2012-05-21T00:39:45Z"
            SessionIndex="d31aefd7f40818a0bec68a79779a397f">
            <saml:SubjectLocality Address="192.168.1.103"
              DNSName="rjb.somedomain.net">
            </saml:SubjectLocality>
            <saml:AuthnContext>
              <saml:AuthnContextClassRef>
                urn:nzl:govt:ict:stds:authn:deployment:GLS:SAML:2.0:ac:classes:ModStrength
              </saml:AuthnContextClassRef>
            </saml:AuthnContext>
          </saml:AuthnStatement>
          <saml:AttributeStatement xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <saml:Attribute Name="urn:nzl:govt:ict:stds:authn:safeb64:attribute:igovt:
IVS:Assertion:Identity"
              NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
              <saml:AttributeValue xsi:type="xs:string">
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9InllcyI PjxuczE6UGFydHkgeG1
sbnM6bnMxPSJ1cm46b2FzaXM6bmFtZXM6dGM6Y2lxOnhwaWw6MyIgeG1sbnM6bnMyPSJ1cm46b2FzaXM6bmFtZXM6dG
M6Y2lxOnhubDozIiB4bWxuczpuczM9InVybjpvYXNpczpuYW1lczp0YzpjaXE6Y3Q6MyIgeG1sbnM6bnM0PSJodHRwO
i8vd3d3LnczLm9yZy8xOTk5L3hsaW5rIiB4bWxuczpuczU9InVybjpvYXNpczpuYW1lczp0YzpjaXE6eGFsOjMiPjxu
czE6UGFydHlOYW1lPjxuczI6UGVyc29uTmFtZT48bnMyOk5hbWVFbGVtZW50IG5zMjpFbGVtZW50VHlwZT0iRmlyc3R
OYW1lIj5BbWVsaWE8L25zMjpOYW1lRWxlbWVudD48bnMyOk5hbWVFbGVtZW50IG5zMjpFbGVtZW50VHlwZT0iTWlkZG
xlTmFtZSI-THVjeTwvbnMyOk5hbWVFbGVtZW50PjxuczI6TmFtZUVsZW1lbnQgbnMyOkVsZW1lbnRUeXBlPSJMYXN0T
mFtZSI-TWFjZG9uYWxkPC9uczI6TmFtZUVsZW1lbnQ-PC9uczI6UGVyc29uTmFtZT48L25zMTpQYXJ0eU5hbWU-PG5z
MTpQZXJzb25JbmZvIG5zMTpHZW5kZXI9IkYiLz48bnMxOkJpcnRoSW5mbz48bnMxOkJpcnRoSW5mb0VsZW1lbnQgbnM
xOlR5cGU9IkJpcnRoWWVhciI-MTk4NTwvbnMxOkJpcnRoSW5mb0VsZW1lbnQ-PG5zMTpCaXJ0aEluZm9FbGVtZW50IG
5zMTpUeXBlPSJCaXJ0aE1vbnRoIj4wNjwvbnMxOkJpcnRoSW5mb0VsZW1lbnQ-PG5zMTpCaXJ0aEluZm9FbGVtZW50I
G5zMTpUeXBlPSJCaXJ0aERheSI-MTQ8L25zMTpCaXJ0aEluZm9FbGVtZW50PjxuczE6QmlydGhQbGFjZURldGFpbHM-
PG5zNTpDb3VudHJ5PjxuczU6TmFtZUVsZW1lbnQgbnM1Ok5hbWVUeXBlPSJOYW1lIj5OZXcgWmVhbGFuZDwvbnM1Ok5
hbWVFbGVtZW50PjwvbnM1OkNvdW50cnk-PG5zNTpMb2NhbGl0eT48bnM1Ok5hbWVFbGVtZW50IG5zNTpOYW1lVHlwZT
0iTmFtZSI-V2VsbGluZ3Rvbjwvbnm1Ok5hbWVFbGVtZW50PjwvbnM1OkxvY2FsaXR5PjwvbnMxOkJpcnRoUGxhY2VEZ
XRhaWxzPjwvbnMxOkJpcnRoSW5mbz48L25zMTpQYXJ0eT4=
              </saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="urn:nzl:govt:ict:stds:authn:attribute:igovt:IVS:
Assertion:FIT"
              NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
              <saml:AttributeValue>
                  <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent" NameQualifier="https://identity.qa.i.govt.nz/ivs/ivs-
idp">WQADF124DE6BD32C4BCE0401CAC451542B5</saml:NameID>
              </saml:AttributeValue>
```

```
            </saml:Attribute>


          </saml:AttributeStatement>
        </saml:Assertion>
      </samlp:Response>
    </samlp:ArtifactResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note: The signature value above is sample only.

The table below will call out the elements of the message that are specific or notable to RealMe. Unless otherwise specified the other message elements will be as per [saml-core-2.0-os].

Container	Container / Element	RealMe Constrained Behaviour
<ArtifactResponse>	<Status>	RealMe SHALL return this element.
<ArtifactResponse>	<Response>	RealMe SHALL return this element.
<Response>	<Status>	RealMe SHALL return this element.
<Response>	<Assertion>	RealMe SHALL return this element.
<Assertion>	<Subject>	RealMe SHALL return this element for Logon Assertions. RealMe SHALL NOT return this element for Identity Assertions.
<Subject>	<NameID>	For identity assertions, the NameID SHALL always be returned with Format attribute: `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` For the usage of RealMe it does not store or generate these types of federated identifiers for Customers. Instead these are passed though RealMe directly from the IAPs only via the <Assertion>'s <AttributeStatement>'s A <Subject> SHALL be passed in order to retain SAML product interoperability, as most products expect this key element in the <Assertion>. The 'transient' <NameID> Format SHALL be used to indicate that the value is strictly temporary and should not be expected to be repeated. Integrated Client SP's SHOULD ignore the value.
<Subject>	<Subject Confirmation>	RealMe SHALL return this element. Any information conveyed by RealMe in the SubjectConfirmation element MUST be honoured by the SP.
<Assertion>	<Conditions>	RealMe SHALL return this element. Any information conveyed by RealMe in the Conditions element MUST be honoured by the SP.
<Assertion>	<Attribute Statement>	The RealMe Assertion Service uses the attribute statements to assert further identity information from IAPs to the SP. The following attributes in 4.2.3.2 MAY be passed.
<Assertion>	<AuthnStatement>	RealMe SHALL return this element.
<AuthnStatement>	<AuthnContext>	RealMe SHALL return this element.
<AuthnContext>	<AuthnContext ClassRef>	The AuthnContextClassRef returned by RealMe SHALL be specific to RealMe handling. This SHALL be as

Container	Container / Element	RealMe Constrained Behaviour
		referenced in section 3.2.
<AuthnContext>	<AuthnContext DeclRef>	RealMe SHALL NOT return this element. As per section 3.2 this element MUST NOT be sent in the <AuthnRequest> and consequently SHALL NOT be returned by RealMe.

Table 18 – Artifact Response element usage

4.2.3.1 Artifact Resolution Processing Rules

As per 3.5.3 of [saml-core-2.0-os] if RealMe does not recognise the artifact sent by the SP as valid then RealMe will respond with an <ArtifactResponse> element with no embedded message.

RealMe SHALL enforce a one-time-use property on the artifact by ensuring that any subsequent request with the same artifact by any requester results in an empty response as described above.

As detailed in section 2.3.6 the SP has a finite amount of time to resolve the artifact before it expires on the IdP.

4.2.3.2 SAML Attributes for the RealMe Assertion Service

The RealMe Assertion Service SHALL use the <Attribute Statement> element as follows.

Attribute	Description
`urn:nzl:govt:ict: stds:authn:safeb64: attribute:igovt:IVS: Assertion:Identity`	OPTIONAL. This SHALL contain the Customer's igovtID identity attributes in a XML format that adheres to the document reference [nzciq]. It is encoded into a XML safe string. The top level element is a <Party> element, that is not signed.
`urn:nzl:govt:ict: stds:authn: attribute:igovt:IVS: Assertion:FIT`	OPTIONAL. This SHALL contain the Customer's IVS Federated Identity Tag. It SHALL NOT be encoded as the other XML attribute types.
`urn:nzl:govt:ict: stds:authn:safeb64: attribute:NZPost:AVS: Assertion:Address`	OPTIONAL. This SHALL contain the Customer's released set of NZ Post verified address attributes in a XML format that adheres to the document reference [nzciq]. It is encoded into a XML safe string. The top level element is a <Party> element, that is not signed.
`urn:nzl:govt:ict: stds:authn:safeb64:`	OPTIONAL.

Attribute	Description
attribute:opaque_token	This SHALL contain the Customer's opaque token from the iCMS to support Client's that have integration settings into RealMe for the Assert Then Logon use case. Clients that do not use the Assert Then Logon use case SHALL NOT receive this attribute.

Table 19 – SAML Attribute usage for Assertion Service

4.2.3.2.1 XML Encoding for SAML Attributes

Expressing XML content in a SAML v2.0 <Assertion> as an <AttributeStatement>requires special handling. Including the un-encoded XML string inside the SAML <Assertion> XML will have the potential to create XML parsing exceptions in a SAML product or implementation. To avoid this problem an encoding is recommended in Appendix D of the document reference [NZ SAMS]. This encoding adopts the following approach.

An <AttributeStatement> will contain an <Attribute> element which:

- Has a Name attribute that begins with: "urn:nzl:govt:ict:stds:authn:safeb64"
- Has a NameFormat that is: `urn:oasis:name:tc:SAML:2.0:attrname-format:uri`
- Has the content of the <AttributeValue> that MUST be a 'Safe Base64' encoding of a well formed XML document. The Safe Base64 encoding is described in RFC3548, section 4. For the purposes of this specification it shall adhere to the XML format prescribed by [nzciq]. Any leading or trailing whitespace is ignored. The Safe Base64 encoding of the data MUST NOT contain whitespace (such as line breaks).

4.2.3.2.2 urn:nzl:govt:ict: stds:authn:safeb64:attributeigovt:IVS:Assertion:Identity

The RealMe Assertion Service SHALL use an <Attribute> inside the <AttributeStatement> to pass the entire IVS identity response as a XML document.

The IVS IAP SHALL employ the New Zealand Government OASIS CIQ Profile [nzciq] to convey the Customers identity attributes. This is a standard that references the "Customer Information Quality v3.0 Specifications" from OASIS which is a XML based standard to define a vocabulary to represent customer data, including identity related attributes.

The following XML document is an example .

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:Party xmlns:ns1="urn:oasis:names:tc:ciq:xpil:3"
           xmlns:ns2="urn:oasis:names:tc:ciq:xnl:3"
           xmlns:ns3="urn:oasis:names:tc:ciq:ct:3"
           xmlns:ns4="http://www.w3.org/1999/xlink"
           xmlns:ns5="urn:oasis:names:tc:ciq:xal:3">
  <ns1:PartyName>
    <ns2:PersonName>
      <ns2:NameElement ns2:ElementType="FirstName">Amelia</ns2:NameElement>
      <ns2:NameElement ns2:ElementType="MiddleName">Lucy</ns2:NameElement>
      <ns2:NameElement ns2:ElementType="LastName">Macdonald</ns2:NameElement>
    </ns2:PersonName>
```

```
  </ns1:PartyName>
  <ns1:PersonInfo ns1:Gender="F"/>
  <ns1:BirthInfo>
    <ns1:BirthInfoElement ns1:Type="BirthYear">1985</ns1:BirthInfoElement>
    <ns1:BirthInfoElement ns1:Type="BirthMonth">06</ns1:BirthInfoElement>
    <ns1:BirthInfoElement ns1:Type="BirthDay">14</ns1:BirthInfoElement>
    <ns1:BirthPlaceDetails>
      <ns5:Country>
        <ns5:NameElement ns5:NameType="Name">New Zealand</ns5:NameElement>
      </ns5:Country>
      <ns5:Locality>
        <ns5:NameElement ns5:NameType="Name">Wellington</ns5:NameElement>
      </ns5:Locality>
    </ns1:BirthPlaceDetails>
  </ns1:BirthInfo>
</ns1:Party>
```

The XML document is constructed as per document reference [nzciq] for the <PartyName> element.

The <PersonInfo> and <BirthInfo> elements are not specified in [nzciq], but the parent [ciq-3.0] specification.

The XML elements utilised in [nzciq] and [ciq-3.0] to convey identity related data SHALL be constrained as follows.

Container	Container / Element	RealMe Constrained Behaviour
<Party>	<PartyName>	As per [nzciq] & [ciq-3.0].
<Party>	<PersonInfo>	As per [nzciq] & [ciq-3.0].
<Party>	<BirthInfo>	As per [nzciq] & [ciq-3.0].
<PartyName>	<PersonName>	As per [nzciq] & [ciq-3.0].
<PersonName	<NameElement>	As per [nzciq] & [ciq-3.0] with the following restrictions: SHALL contain one and only one <NameElement> with ElementType =“LastName”. MAY contain one and only one <NameElement> with ElementType = “FirstName”. MAY contain one and only one <NameElement> with ElementType = “MiddleName”. If present, SHALL NOT be empty or blank.
<PersonInfo>	Gender	As per [ciq-3.0].
<BirthInfo>	<BirthInfoElement>	As per [ciq-3.0] with the following restrictions: SHALL NOT contain a <BirthInfoElement> with

Container	Container / Element	RealMe Constrained Behaviour
		Type= “MothersName” SHALL contain one and only one <BirthInfoElement> with Type = “BirthYear”. SHALL contain one and only one <BirthInfoElement> with Type = “BirthMonth”. SHALL contain one and only one <BirthInfoElement> with Type = “BirthDay”. SHALL NOT contain a <BirthInfoElement> with Type=”BirthTime”.
<BirthInfo>	<BirthPlaceDetails>	As per [ciq-3.0] with the following restrictions: SHALL contain one <Country> and/or one <Locality> container.
<BirthPlaceDetails>	<Country>	As per [ciq-3.0] with the following restrictions: If present, SHALL contain a NameElement of NameType=”Name”
<BirthPlaceDetails>	<Locality>	As per [ciq-3.0] with the following restrictions: If present, SHALL contain a NameElement of NameType=”Name” If present, SHALL NOT contain a NameElement of NameType=”Type”

Table 20 – IVS Constraints of NZCIQ for identity information

4.2.3.2.3 urn:nzl:govt:ict:stds:authn:attributeigovt:IVS:Assertion:FIT

The RealMe Assertion Service SHALL use an <Attribute> inside the <AttributeStatement> to pass the IVS Federated Identity Tag.

It is a <saml:NameID> and it SHALL NOT be subject to the Safe Base64 XML encoding.

The following XML document is an example.

```
<saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
NameQualifier="https://identity.qa.i.govt.nz/ivs/ivs-
idp">WQADF124DE6BD32C4BCE0401CAC451542B5</saml:NameID>
```

4.2.3.2.4 urn:nzl:govt:ict:stds:authn:safeb64:attribute:NZPost:AVS:Assertion:Address

The RealMe Assertion Service SHALL use an <Attribute> inside the <AttributeStatement> to pass the entire NZ Post AVS address response as an XML document.

The XML document is constructed as per document reference [nzciq]. This specifies the mappings of address related data into XML elements adhering to [ciq-3.0].

The following XML documents are examples of the types of address formats supported.

Address Format Type	Address Document
New Zealand	25 High Street , Island Bay 6023

Address Format Type	Address Document
Standard Address	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <p:Party xmlns:p="urn:oasis:names:tc:ciq:xpil:3" xmlns:a="urn:oasis:names:tc:ciq:xal:3"> <a:Addresses> <a:Address Type=”NZStandard” Usage=”Residential” DataQualityType=”Valid” ValidFrom=”03/01/2013> <a:Locality> <a:NameElement NameType="NZTownCity">Island Bay</a:NameElement> </a:Locality> <a:Thoroughfare> <a:NameElement NameType="NZNumberStreet”>25 High Street</a:NameElement </a:Thoroughfare> <a:PostCode> <a:Identifier Type=”NZPostCode”>6023</a:Identifier> </a:PostCode> </a:Address> </a:Addresses> </p:Party>
New Zealand Standard Address	Flat 1, 23 King Street Newtown, Wellington 6021 <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <p:Party xmlns:p="urn:oasis:names:tc:ciq:xpil:3" xmlns:a="urn:oasis:names:tc:ciq:xal:3"> <a:Addresses> <a:Address Type=”NZStandard” Usage=”Residential” DataQualityType=”Valid” ValidFrom=”03/01/2013> <a:Locality> <a:NameElement NameType="NZTownCity">Wellington</a:NameElement> <a:NameElement NameType="NZSubburb">Newtown</a:NameElement> </a:Locality> <a:Thoroughfare> <a:NameElement NameType="NZNumberStreet”>23 King Street</a:NameElement </a:Thoroughfare> <a:Premises> <a:NameElement NameType="NZUnit”>Flat 1</a:NameElement> <a:Premises> <a:PostCode> <a:Identifier Type=”NZPostCode”>6023</a:Identifier> </a:PostCode> </a:Address> </a:Addresses> </p:Party>

Address Format Type	Address Document
Rural Delivery Address	634 Clifford Road Mangawai RD5 KAIWAKA 0582

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<p:Party xmlns:p="urn:oasis:names:tc:ciq:xpil:3"
         xmlns:a="urn:oasis:names:tc:ciq:xal:3">
  <a:Addresses>
    <a:Address Type=”NZRuralDelivery” Usage=”Residential” DataQualityType=”Valid”
ValidFrom=”01/11/2011>
      <a:Locality>
        <a:NameElement NameType="NZTownCity"> KAIWAKA </a:NameElement>
        <a:NameElement NameType="NZSubburb"> Mangawai </a:NameElement>
      </a:Locality>
      <a:Thoroughfare>
        <a:NameElement NameType="NZNumberStreet”>634 Clifford Road</a:NameElement>
      </a:Thoroughfare>
      <a:RuralDelivery>
        <a:Identifier Type=”NZRuralDelivery”>RD 5</a:Identifier>
      </a:RuralDelivery>
      <a:PostCode>
        <a:Identifier Type=”NZPostCode”>0582</a:Identifier>
      </a:PostCode>
    </a:Address>
  </a:Addresses>
</p:Party>
```

Table 21 – AVS Examples of NZCIQ for identity information

The XML elements utilised in [nzciq] to convey identity related data SHALL be constrained as follows.

Container	Container / Element	RealMe Constrained Behaviour
<Party>	<Addresses>	As per [nzciq] & [ciq-3.0].
<Addresses>	<Address>	As per [nzciq] & [ciq-3.0]. The full standard OASIS CIQ V3 Address structure of elements is used.

Table 22 – AVS Constraints of NZCIQ for identity information

4.3 Signing and Encryption

Signing MUST NOT occur on the ArtifactResolve message by the SP.

Signing SHALL NOT occur on the ArtifactResponse message by RealMe IdP.

Signing SHALL occur on the Assertion message by RealMe IdP.

XML encryption SHALL NOT occur in the Assertion element by RealMe IdP.

XML encryption SHALL NOT occur in the NameID element by RealMe IdP.

XML encryption SHALL NOT occur in the Attribute elements by RealMe IdP.

If any elements marked as SHALL NOT be signed are in fact signed then the SP is not required to verify the signature. However the SP MUST verify signatures of all the elements that SHALL be signed. Explicitly this is the Assertion message.

4.4 SOAP Back Channel

The SOAP back channel call for the artifact resolution will occur over mutual SSL. The mutual SSL certificate MUST be distinct from the SAML messaging certificate.

4.5 Error Handling from an <AuthnRequest>

Upon receipt on an <AuthnRequest> and an error condition arising during the subsequent user transaction RealMe MUST securely communicate with the initiating SP the relevant application error code. The SAML v2.0 Response message contains support for error codes in the <Status> element. See [saml-core-2.0-os] section 3.2.2.1 and the constraints imposed by NZ SAMS sections 11 and 12.

The <Status> element contains two tiers of <StatusCode> elements. The first tier is reserved for status codes within the SAML v2.0 standard. The second tier is available for system entities to use defined SAML v2.0 status codes and define more specific status codes by defining appropriate URI references[2].

4.5.1 RealMe Assertion Service Error Status Codes

The table below lists error status codes that are emitted by normal operation of the RealMe Assertion Service. They MAY be received conditional to user operation, incorrect SAML messaging or general RealMe availability. These use and extend from the second tier status codes specified in [saml-core-2.0-os] section 3.2.2.2.

A service agency MUST provide handling in their integrated applications when these error status codes are received.

Error URN	Condition
urn:oasis:names: tc:SAML:2.0:status: AuthnFailed	RealMe was unable to successfully authenticate the principal. SHALL be returned by RealMe when the logon was aborted at the RealMe Logon Service by the user. SHALL be returned by RealMe when the user declines to release their identity attributes to the SP.

[2] See [saml-core-2.0-os] line 1646.

Error URN	Condition
	SHALL be returned by RealMe when the user cancels their assertion transaction upon being informed of the unavailability of an IAP.
urn:oasis:names: tc:SAML:2.0:status: NoAvailableIDP	Used by an intermediary to indicate that none of the supported identity provider <Loc> elements in an<IDPList> can be resolved or that none of the supported identity providers are available. SHALL be returned by RealMe when an RealMe Logon Service authentication against a specific credential provider could not be performed at this time.
urn:oasis:names: tc:SAML:2.0:status: NoPassive	Indicates the responding provider cannot authenticate the principal passively, as has been requested. SHALL be returned by RealMe when the <AuthnRequest> contains the IsPassive flag equal to the XML data-type of true. IsPassive is not supported in RealMe.
urn:oasis:names: tc:SAML:2.0:status: RequestUnsupported	The SAML responder or SAML authority does not support the request. SHALL be returned by RealMe when the message contains unexpected parameters and or does not conform with this specification.
urn:nzl:govt:ict:stds: authn:deployment:RealMe: SAML:2.0:status:Timeout	SHALL be returned when during the act of identity verification the user's session times out in RealMe.
urn:nzl:govt:ict:stds: authn:deployment:RealMe: SAML:2.0:status: InternalError	SHALL be returned when there is an internal error in RealMe. Includes also when there is an error in processing a response from an IAP.
urn:oasis:names: tc:SAML:2.0:status: RequestDenied	The SAML responder or SAML authority is able to process the request but has chosen not to respond. This status code MAY be used when there is concern about the security context of the request message or the sequence of request messages received from a particular requester. SHALL be returned by RealMe when the <AuthnRequest> contains prohibited data within accepted parameters (e.g. SPNameQualifier). SHALL be returned by RealMe when the <AuthnRequest> contains an IssueInstant outside an accepted time window. SHALL be returned by RealMe during an identity assertion when an IAP has not responded with identity attributes due to the account at the IAP being not valid. SHALL be returned by RealMe when the Client integration contains invalid settings. SHALL be returned by RealMe when the Customer does not have an active integration consent for the IAP.
urn:oasis:names: tc:SAML:2.0:status: NoAuthnContext	The specified authentication context requirements cannot be met by the responder. SHALL be returned by RealMe Assertion Service when the <AuthnRequest> contains a RequestedAuthnContext element that contains no AuthnContextClassRef.
urn:oasis:names: tc:SAML:2.0:status: UnsupportedBinding	The SAML responder cannot properly fulfil the request using the protocol binding specified in the request.
urn:oasis:names: tc:SAML:2.0:status: UnknownPrincipal	The responding provider does not recognize the principal specified or implied by the request. SHALL be returned by the RealMe Assertion Service when the Customer

Error URN	Condition
	attempts to logon during an Assert type use case but does does not have a RealMe account registered with the RealMe Logon Service. The Customer must use the create a RealMe account first via registering or reusing a RealMe logon.

Table 23 - RealMe Assertion Service Error Status Codes

4.5.2 Other SAML v2.0 Status Codes

The table below lists error status codes that are not emitted by normal operation of RealMe. They are eligible to be passed as part of RealMe support for the wider SAML v2.0 specification. The receipt of these is likely due to incorrect SAML v2.0 messaging or an error in the integration with RealMe.

A Client SHOULD provide logging for analysis in their integrated applications when these error status codes are received. If received then the recommended action is contact with RealMe operations personnel for further diagnosis of the messaging.

Error URN	Condition
urn:oasis:names: tc:SAML:2.0:status: InvalidAttrNameOrValue	Unexpected or invalid content was encountered within a <saml:Attribute> or <saml:AttributeValue> element.
urn:oasis:names: tc:SAML:2.0:status: InvalidNameIDPolicy	The responding provider cannot or will not support the requested name identifier policy.
urn:oasis:names: tc:SAML:2.0:status: NoSupportedIDP	Used by an intermediary to indicate that none of the identity providers in an <IDPList> are supported by the intermediary.
urn:oasis:names: tc:SAML:2.0:status: PartialLogout	Used by a session authority to indicate to a session participant that it was not able to propagate logout to all other session participants.
urn:oasis:names: tc:SAML:2.0:status: ProxyCountExceeded	Indicates that a responding provider cannot authenticate the principal directly and is not permitted to proxy the request further.
urn:oasis:names: tc:SAML:2.0:status: RequestVersionDeprecated	The SAML responder cannot process any requests with the protocol version specified in the request.
urn:oasis:names: tc:SAML:2.0:status: RequestVersionTooHigh	The SAML responder cannot process the request because the protocol version specified in the request message is a major upgrade from the highest protocol version supported by the responder.
urn:oasis:names: tc:SAML:2.0:status: RequestVersionTooLow	The SAML responder cannot process the request because the protocol version specified in the request message is too low.
urn:oasis:names: tc:SAML:2.0:status: ResourceNotRecognized	The resource value provided in the request message is invalid or unrecognised.
urn:oasis:names: tc:SAML:2.0:status: TooManyResponses	The response message would contain more elements than the SAML responder is able to return.
urn:oasis:names: tc:SAML:2.0:status: UnknownAttrProfile	An entity that has no knowledge of a particular attribute profile has been presented with an attribute drawn from that profile.

Table 24 – Other RealMe Error Status Codes

4.5.3 RealMe Error Status Messages

In an error response the <Status> element SHALL contain a <StatusMessage> element that contains meta information about the error. RealMe MAY subject these values to change.

RealMe SHALL NOT subject the values of <StatusCode> elements to change. RealMe MAY subject the <StatusMessage> values to change and therefore they MUST NOT be used to direct SP behaviour.

4.5.4 RealMe Assertion Service Error Conditions from an <AuthnRequest>

The following table lists the error status codes that SHALL be returned due to incorrect validation of an <AuthnRequest> from the RealMe Logon Assertion based on the processing rules specified in section 3 of this specification.

<StatusCode> Element Values		**Error Condition**
Top Level	**Second Level**	
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestDenied	1. <AuthnRequest> attribute IssueInstant not within the accepted time window.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:NoPassive	2. <AuthnRequest> attribute IsPassive is provided and set to the XML data-type of true.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	3. None of the<AuthnRequest> attributes AssertionConsumerServiceIndex, ProtocolBinding, or AssertionConsumerServiceURL are provided.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	4. <AuthnRequest> attribute ProtocolBinding is provided and contains a value not equal to urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	5. The <AuthnRequest> attributes AssertionConsumerServiceURL and AssertionConsumerServiceIndex are both provided.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	6. <AuthnRequest> element <Issuer> is not in a format that identifies a privacy domain.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	7. <AuthnRequest> / <NameIDPolicy> attribute Format was provided and is not either : • urn:oasis:names:tc:SAML:2.0:nameid-format:transient • urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestDenied	8. <AuthnRequest> / <NameIDPolicy> attribute SPNameQualifier is provided and is set to another namespace other than the SP's privacy domain.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext	9. <AuthnRequest> / <RequestedAuthnContext> is provided but the sub element <AuthnContextClassRef> is not provided.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	10. <AuthnRequest> / <RequestedAuthnContext> / <AuthnContextClassRef> is provided with unsupported values.

<StatusCode> Element Values		Error Condition
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	11. <AuthnRequest> / <RequestedAuthnContext> / <AuthnContextDeclRef> is provided.
urn:oasis:names:tc:SAML:2.0:status:Responder	urn:oasis:names:tc:SAML:2.0:status:RequestDenied	12. An <AuthnRequest> is received from a Client whose SP metadata contains an <EntityDescriptor> validUntil attribute that is provided and is a date in the past.

Table 25 –RealMe Assertion Service Error Conditions

4.5.5 Translation from RealMe Logon Service error Status Codes to RealMe Status Codes

The RealMe Assertion Service uses the RealMe Logon Service as the IdP to authenticate users. When the RealMe Logon Service returns an SAML v2.0 error response to the RealMe Assertion Service, on some errors the RealMe Assertion Service SHALL translate the RealMe Logon Service error to an equivalent RealMe Assertion Service error response to the Client SP.

The following table lists the RealMe Logon Service errors that are translated to the equivalent emitted RealMe Assertion Service SAML v2.0 error response to the Client SP.

Received RealMe Logon Service Error Status Code	Returned RealMe Assertion Service Error Status Code
urn:oasis:names:tc:SAML:2.0:status:AuthnFailed	urn:oasis:names:tc:SAML:2.0:status:AuthnFailed
urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP	urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP
urn:oasis:names:tc:SAML:2.0:status:NoPassive	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError
urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError
urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:Timeout	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:Timeout
urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError
urn:oasis:names:tc:SAML:2.0:status:UnsupportedBinding	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError
urn:oasis:names: tc:SAML:2.0:status:UnknownPrincipal	urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal
All other errors not defined above.	urn:nzl:govt:ict:stds:authn:deployment:RealMe:SAML:2.0:status:InternalError

Table 26 –RealMe Logon Service to RealMe Assertion Service Error Translations

All other error codes sent by the RealMe Logon Service to the RealMe Assertion Service are handled by RealMe Assertion Service user interface.

Note that there may be RealMe Assertion Service UI handling to allow a Customer to retry an authentication against the RealMe Logon Service; this may subsequently mask an error from the RealMe Logon Service once the Customer is returned from the RealMe Assertion Service to the SP.

4.5.6 Sample Error Message

A sample response message is as follows:

```
<samlp:Response xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  Destination="https://www.sample-client.co.nz/sso/ACS"
  ID="cec17a74048a4b35511d168834520380"
  InResponseTo="a958a20e059c26d1cfb73163b1a6c4f9"
  IssueInstant="2012-05-21T00:39:45Z" Version="2.0">
  <saml:Issuer>https://realme.govt.nz/realme/assert-idp</saml:Issuer>
  <samlp:Status>
    <samlp:StatusCode
      Value="urn:oasis:names:tc:SAML:2.0:status:Responder">
      <samlp:StatusCode
        Value="urn:oasis:names:tc:SAML:2.0:status:AuthnFailed">
      </samlp:StatusCode>
    </samlp:StatusCode>
    <samlp:StatusMessage>User chose not to assert their identity.</samlp:StatusMessage>
  </samlp:Status>
</samlp:Response>
```

The error message will be sent back over the same binding as success messages, i.e. HTTP-Artifact.

5 Message Specification Versioning

Future versions of this specification MAY be released with changes to the SAML v2.0 messaging. RealMe SHALL support concurrent versions of the messaging specification. This will allow existing SP's still to retain their current SAML v2.0 messaging specification version while allowing other SP's to use more recent versions of this messaging specification version.

Each SP MUST nominate the version of this messaging specification they are implementing against. This will occur as an out-of-band process during the integration with RealMe. RealMe SHALL statically configure each distinct SP against a specification and process the messaging accordingly.

The SP specific message version SHALL be identified by the Issuer element in the SAML v2.0 <AuthnRequest>. See section 3.2.

6 Identity Privacy Domains

RealMe SHALL implement the concept of identity privacy domains. In the context of this specification a privacy domain is a SAML v2.0 NameID generation space. SP's that reside in the same identity privacy domain will be returned the same SAML v2.0 NameID in the Assertion to reference the Customer's logon or identity in the SAML v2.0 response.

The <Issuer> element in the SAML v2.0 <AuthnRequest> MUST be used by the SP to identify the identity privacy domain they reside in (see section 3.2.2). The value of the <Issuer> that the SP will use is declared by the SP in the entityID when they provide their metadata to the IdP.

RealMe SHALL understand privacy domains by the following pattern:

[protocol]://[client-domain]/[privacy-context-name]/[service-name]{-client-environment}

where:

- elements in [] are mandatory
- elements in {} are optional

e.g. https://www.sample-client.co.nz/onlineservices/service1

The adoption of the pattern for <Issuer> and entityID is REQUIRED by SP's. RealMe SHALL adopt the pattern for its IdP entityID's.

The RealMe Logon Service SHALL implement a statically configured mapping between the <Issuer> value and the identity privacy domains they MAY share with other issuers. This will occur as an out-of-band process during the integration with the RealMe Logon Service.

The following table illustrates a sample mapping that RealMe will use to map between issuers and privacy domains.

<Issuer>	Identity Privacy Domain A	Identity Privacy Domain B	Identity Privacy Domain C
https://client-1.co.nz/pd-1/service1	X		
https://client-2.co.nz/pd-1/service1		X	
https://client-2.co.nz/pd-1/service2		X	
https://client-2.co.nz/pd-2/service3			X

Table 27 – Identity Privacy Domains for Issuers

In the above example the Client SP, client-1, implements only one identity privacy domains for their enterprise, however they MUST still include a privacy context name identifier in their Issuer.

The Client SP client-2 has implemented two different identity privacy domains within its enterprise. The first, "PD-1", is shared between two applications. The second, "PD-2" is only used by one application.

<Issuer> values provided to the RealMe Assertion Service will be utilised in calls to the RealMe Logon Service to identify the initiating Client SP to that service. At the time the Client SP plans to integrate to the RealMe Assertion Service the <Issuer> must be known on the RealMe Logon Service; this may or may not be an existing entityID depending on the requirements of the Client SP. For further details refer to the [realme-int-guide].

7 Appendix A: Sample Service Provider Metadata

This appendix provides a suitable sample of the metadata that a Service Provider is expected to provide. The production and consumption of SAML v2.0 metadata is a key requirement to establishing the circle of trust when integrating with RealMe.

If the SP's chosen SAML product does not support the creation of metadata RealMe MTS provides sample SP metadata in the form of an XML file, complete with demonstration agency public certificate for SAML <AuthnRequest> signatures.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
  entityID="https://www.sample-client.co.nz/onlineservices/service1"
  validUntil="2011-01-01T00:00:00Z">
  <SPSSODescriptor AuthnRequestsSigned="true"
    WantAssertionsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            <!-- certificate omitted for clarity -->
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
    <AssertionConsumerService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
      Location="https://www.sample-sp.co.nz/sso/ACS" index="0"
      isDefault="true">
    </AssertionConsumerService>
  </SPSSODescriptor>
  <Organization>
    <OrganizationName xml:lang="en-us">
      Sample Service Provider
    </OrganizationName>
    <OrganizationDisplayName xml:lang="en-us">
      Sample Service Provider
    </OrganizationDisplayName>
    <OrganizationURL xml:lang="en-us">
      https://www.sample-sp.co.nz/realm
    </OrganizationURL>
  </Organization>
```

```
  <ContactPerson contactType="support">
    <Company>Sample Service Provider</Company>
    <GivenName></GivenName>
    <SurName></SurName>
  </ContactPerson>
</EntityDescriptor>
```

7.1 Elements in Service Provider Metadata

The following tables define the restrictions placed on SP metadata by RealMe.

7.1.1 Root Element

A SAML metadata instance describes either a single entity or multiple entities via use of <EntityDescriptor> or <EntitiesDescriptor> root elements respectively. RealMe supports only the use of single <EntityDescriptor> root elements in a SAML v2.0 metadata file.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<EntityDescriptor>	A single <EntityDescriptor> MUST be provided. ☑ NZ SAMS Constraint on NZ SAMS: Single <EntityDescriptor> allowed only.	OPTIONAL. NZ SAMS line 171. ☑ SAML v2.0	Must be a root element if <EntitiesDescriptor> is not used. Ref [saml-metadata-2.0-os] section 2.3, line 307.
<Entities Descriptor>	MUST NOT be provided. RealMe does not support the simultaneous integration of more than one SP SAML v2.0 entity in a SAML v2.0 metadata file. The individual SP SAML v2.0 entities MUST be specified as separate <EntityDescriptor> root elements in distinct SAML v2.0 metadata files. ☑ NZ SAMS Constraint on NZ SAMS: <EntitiesDescriptor> not allowed.	OPTIONAL. NZ SAMS line 171. ☑ SAML v2.0	Must be a root element if <EntityDescriptor> is not used. Ref [saml-metadata-2.0-os] section 2.3, line 307.

7.1.2 Element <EntityDescriptor>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	MAY be provided. RealMe SHALL NOT require the metadata to be signed. If provided it MUST NOT contain white spaces. ☑ NZ SAMS Constraint on NZ SAMS: Whitespace not allowed.	OPTIONAL if the <EntityDescriptor> is not signed. NZ SAMS line 210. ☑ SAML v2.0	Optional. A document-unique identifier for the element, typically used as a reference point when signing. Ref [saml-metadata-2.0-os] section 2.3.2, line 374.
entityID	MUST be provided. MUST be in an RealMe identity privacy domain format as per section 7. ☑ NZ SAMS Constraint on NZ SAMS: Format as per section 7	REQUIRED. NZ SAMS line 209. ☑ SAML v2.0	Required attribute. Specifies the unique identifier of the SAML entity whose metadata is described by the element's contents. Ref [saml-metadata-2.0-os] section 2.3.2, line 371.
validUntil	SHOULD be provided and equal to the SP SAML messaging certificate expiry date. If provided RealMe SHALL only enforce the validUntil attribute. If provided RealMe SHALL check for each <AuthnRequest> sent from the Client that the validUntil attribute has not expired. If SP metadata expiry is detected RealMe SHALL respond as shown in section 5.5 If not provided RealMe SHALL not perform any expiration time checking on SP metadata.	One of validUntil or cacheDuration SHOULD be provided. NZ SAMS line 214. ☑ SAML v2.0	Optional attribute indicates the expiration time of the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 376.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	☑ NZ SAMS		
cacheDuration	MAY be provided. If provided RealMe SHALL NOT test this value. RealMe enforces only the validUntil attribute to be in alignment with the NZ SAMS recommendation that one of validUntil or cacheDuration SHOULD be provided. ☑ NZ SAMS	One of validUntil or cacheDuration SHOULD be provided. NZ SAMS line 214. ☑ SAML v2.0	Optional attribute indicates the maximum length of time a consumer should cache the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 379.
<ds:Signature>	MUST NOT be provided. RealMe supports metadata signing only within the SPSSODescriptor element. ☑ NZ SAMS Constraint on NZ SAMS: Cannot be provided	OPTIONAL ☑ SAML v2.0	Optional. An XML signature that authenticates the containing element and its contents. Ref [saml-metadata-2.0-os] section 2.3.2, line 382.
<Extensions>	MUST NOT be provided. ☑ NZ SAMS Constraint on NZ SAMS: Cannot be provided.	NOT RECOMMENDED and if present will be ignored. NZ SAMS line 222. ☑ SAML v2.0	Optional metadata extensions. Ref [saml-metadata-2.0-os] section 2.3.2, line 385.
<SPSSODescriptor>	For a Client acting as a SP only the <EntityDescriptor> MUST contain only one <SPSSODescriptor> element. ☑ NZ SAMS	For a Client acting as a SP only the <SPSSO Descriptor> MUST be provided. NZ SAMS lines 223-231. ☑ SAML v2.0	<RoleDescriptor>, <IDPSSODescriptor>, <SPSSODescriptor>, <AuthnAuthorityDescriptor>, <AttributeAuthorityDescriptor>, <PDPDescriptor> [One or More] OR <AffiliationDescriptor> Ref [saml-metadata-2.0-os] section 2.3.2, lines 389-395.
<Organization>	MUST be provided. Only	REQUIRED.	Optional. An element

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	for NZ SAMS compliance. Format as per [saml-metadata-2.0-os] section 2.3.2.1, line 443. ☑ NZ SAMS	NZ SAMS line 232. ☑ SAML v2.0	identifying the organization responsible for the SAML entity described by the element. Ref [saml-metadata-2.0-os] section 2.3.2, line 396.
<ContactPerson>	RECOMMENDED. Only for NZ SAMS compliance. Format as per [saml-metadata-2.0-os] section 2.3.2.2, line 476. ☑ NZ SAMS	RECOMMENDED that one be included. NZ SAMS line 235. ☑ SAML v2.0	Optional sequence of elements identifying various kinds of contact personnel. Ref [saml-metadata-2.0-os] section 2.3.2, line 399.
<AdditionalMetadataLocation>	MUST NOT be provided. ☑ NZ SAMS	MUST NOT be provided. NZ SAMS line 236. ☑ SAML v2.0	Optional sequence of namespace-qualified locations where additional metadata exists for the SAML entity. This may include metadata in alternate formats or describing adherence to other non-SAML specifications. Ref [saml-metadata-2.0-os] section 2.3.2, line 401.

Table 28

7.1.3 Element <SPSSODescriptor>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	MAY be provided. RealMe SHALL NOT require the metadata to be signed. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional. A document-unique identifier for the element, typically used as a reference point when signing. Ref [saml-metadata-2.0-os] section 2.3.2, line 555.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
validUntil	MAY be provided. RealMe SHALL NOT test this value. ☑ NZ SAMS Constraint on NZ SAMS: Expiry of SP metadata not supported in the SPSSODescriptor	Per OASIS. ☑ SAML v2.0	Optional attribute indicates the expiration time of the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 557.
cacheDuration	MAY be provided. RealMe SHALL NOT test this value. ☑ NZ SAMS Constraint on NZ SAMS: Caching duration of SP metadata not supported in the SPSSODescriptor.	Per OASIS. ☑ SAML v2.0	Optional attribute indicates the maximum length of time a consumer should cache the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 560.
AuthnRequestsSigned	MUST be the XML data-type of true. Signing of <AuthnRequest> is required by section 3.4 ☑ NZ SAMS Constraint on NZ SAMS: Signed <AuthnRequest> is mandatory.	Per OASIS. ☑ SAML v2.0	Optional attribute that indicates whether the <samlp:AuthnRequest> messages sent by this service provider will be signed. If omitted, the value is assumed to be false. Ref [saml-metadata-2.0-os] section 2.3.2, line 740.
WantAssertionsSigned	MUST be the XML data-type of true. Signing of Assertions is required by section 4.2.3.2.4 ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional attribute that indicates a requirement for the <saml:Assertion> elements received by this service provider to be signed. If omitted, the value is assumed to be false.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	Constraint on NZ SAMS: Signing Assertions is mandatory on RealMe		This requirement is in addition to any requirement for signing derived from the use of a particular profile/binding combination. Ref [saml-metadata-2.0-os] section 2.3.2, line 743.
errorURL	MAY be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional URI attribute that specifies a location to direct a user for problem resolution and additional support related to this role. Ref [saml-metadata-2.0-os] section 2.3.2, line 560.
protocolSupport Enumeration	MUST be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Required. A whitespace-delimited set of URIs that identify the set of protocol specifications supported by the role element. For SAML v2.0 entities, this set MUST include the SAML protocol namespace URI `urn:oasis:names:tc:SAML:2.0:protocol`. Ref [saml-metadata-2.0-os] section 2.3.2, line 560.
<KeyDescriptor>	MUST be provided to meet the digital signing requirements in section 3.4. ☑ NZ SAMS	MUST be provided. NZ SAMS line 246. ☑ SAML v2.0	Optional sequence elements that provides information about the cryptographic keys that the entity uses when acting this role. Ref [saml-metadata-2.0-os] line 579.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<ArtifactResolution Service>	MAY be provided but SHALL not be used by RealMe. An Artifact Resolution Profile is not used by RealMe to receive messages from a SP. ☑ NZ SAMS Constraint on NZ SAMS: Not supported on RealMe.	MUST be provided if an SP supports the artifact binding. NZ SAMS line 254. ☑ SAML v2.0	Optional. Zero or more elements that describe indexed endpoints that support the Artifact Resolution Profile. Ref [saml-metadata-2.0-os] section 2.3.2, line 648.
<SingleLogoutService>	MAY be provided but an SP SingleLogoutService SHALL not be used by RealMe. If provided will be ignored. ☑ NZ SAMS Constraint on NZ SAMS: If provided ignored.	MUST be provided if an SP supports a single logout service. NZ SAMS line 256. ☑ SAML v2.0	Optional. Zero or more elements that describe indexed endpoints that support the Single Logout Profiles. Ref [saml-metadata-2.0-os] section 2.3.2, line 652.
<ManageNameID Service>	SHOULD NOT be provided. If provided will be ignored. ☑ NZ SAMS	SHOULD NOT be provided. NZ SAMS line 260. ☑ SAML v2.0	Optional. Zero or more elements that describe indexed endpoints that support the Name Identifier Management profiles. Ref [saml-metadata-2.0-os] section 2.3.2, line 655.

<NameIDFormat>	MUST be provided.
	The format urn:
	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
	MUST be listed. This is required to be consistent with 3.2.3.
	The format
	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
	MAY also be listed if intended to be used as a
		MAY be provided.
		The `urn:` **`oasis:names:tc:SAML:2.0:nameid-format:persistent`** format must be listed if the <NameIDFormat> element is used in the metadata
		NZ SAMS line 264.
			Optional.
			Zero or more elements that enumerate the name identifier formats supported by this system entity acting in this role.
			See Section 8.3 of [saml-core-2.0-os] for some possible values for this element.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	format in an <NameIDPolicy> in an <AuthnRequest>. ☑ NZ SAMS	☑ SAML v2.0	Ref [saml-metadata-2.0-os] section 2.3.2, line 658.
<AssertionConsumer Service>	At least one MUST be provided that contains the attributes with the specialised requirements described in 7.1.5. ☑ NZ SAMS	Per OASIS. NZ SAMS line 288. ☑ SAML v2.0	MUST be provided. One or more elements that describe indexed endpoints that support the profiles of the Authentication Request protocol defined in [saml-profiles-2.0-os]. All service providers support at least one such endpoint, by definition. Ref [saml-metadata-2.0-os] line 748.
<AttributeConsuming Service>	MAY be provided. If provided SHALL be ignored. ☑ NZ SAMS	OPTIONAL. If provided MAY be ignored. NZ SAMS line 283. ☑ SAML v2.0	Zero or more elements that describe an application or service provided by the service provider that requires or desires the use of SAML attributes. Ref [saml-metadata-2.0-os] line 752.
<ds:Signature>	MAY be provided by the SP to sign metadata. RealMe SHALL NOT require the metadata to be signed. If the element is present RealMe SHALL check the signature. ☑ NZ SAMS	OPTIONAL ☑ SAML v2.0	Optional. An XML signature that authenticates the containing element and its contents. Ref [saml-metadata-2.0-os] section 2.3.2, line 572.
<Extensions>	MUST NOT be provided. ☑ NZ SAMS	NOT RECOMMENDED and if present will be ignored.	Optional metadata extensions. Ref [saml-metadata-2.0-os]

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	Constraint on NZ SAMS: Cannot be provided.	NZ SAMS line 222. ☑ SAML v2.0	section 2.3.2, line 575.

Table 29

7.1.4 Element <KeyDescriptor>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
use	MUST contain the value 'signing' to meet the digital signing requirements in section 3.4. ☑ NZ SAMS	MUST be used. NZ SAMS line 246. ☑ SAML v2.0	Optional attribute specifying the purpose of the key being described. Values are drawn from the KeyTypes enumeration, and consist of the values 'encryption' and 'signing'. Ref [saml-metadata-2.0-os] line 615.
	Constraint on NZ SAMS: Must specify 'signing' only		
<ds:KeyInfo>	MUST be provided. MUST contain the Client SAML v2.0 messaging certificate. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Required. The element that identifies the SAML messaging certificates and/or keys. Ref [saml-metadata-2.0-os] line 618.
<EncryptionMethod>	MAY be provided. If provided SHALL be ignored. RealMe does not perform SAML v2.0 message encryption. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional element specifying an algorithm and algorithm-specific settings supported by the entity. The exact content varies based on the algorithm supported. Ref [saml-metadata-2.0-os] line 621.

Table 30

7.1.5 Element : <AssertionConsumerService>

An SP MAY provide several <AssertionConsumerService>'s. RealMe requires a SP MUST supply at least one with the specified attributes below.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
index	MUST be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MUST be provided. A required attribute that assigns a unique integer value to the endpoint so that it can be referenced in a protocol message. Ref [saml-metadata-2.0-os] line 264.
isDefault	MAY be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional. A boolean attribute used to designate the default endpoint among an indexed set. If omitted, the value is assumed to be false. If there are a sequence of endpoints based on this type, the default endpoint is the first such endpoint with the isDefault attribute set to true. If no such endpoints exist, the default endpoint is the first such endpoint without the isDefault attribute set to false. If no such endpoints exist, the default endpoint is the first element in the sequence. Ref [saml-metadata-2.0-os] line 272.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
Binding	MUST use the value: "urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact" This is the SP binding for RealMe to send artifacts to. As per section 4.1 this must the HTTP-Artifact binding. ☑ NZ SAMS Constraint on NZ SAMS: Only the HTTP-Artifact binding is accepted.	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-metadata-2.0-os] line 229.
Location	MUST be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MUST be provided. A required URI attribute that specifies the location of the endpoint. Ref [saml-metadata-2.0-os] line 232.
ResponseLocation	MUST NOT be provided. The attribute is unused for RealMe's profile where only single types of request and response messages are applicable. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional. Used only for protocols or profiles that has more than one type of request or response message. Ref [saml-metadata-2.0-os] line 235.

Table 31

8 Appendix B: RealMe Metadata

This appendix provides the metadata that RealMe is SHALL provide. RealMe SHALL subject this to change.

THIS CONTENT IS INDICITAIVE ONLY. IT WILL BE COMPLETED POST REALME SAML V2.0 DESIGN.

```
<EntityDescriptor entityID="https://realme.govt.nz/realme/assert-idp"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  <IDPSSODescriptor WantAuthnRequestsSigned="true"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            <!-- certificate omitted for clarity -->
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <ArtifactResolutionService index="0"
      isDefault="true"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="https://as.realme.govt.nz:443/sso/ArtifactResolver/metaAlias/assert-
idp"/>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameidformat:unspecified</NameIDFormat>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://realme.govt.nz:443/sso/SSORedirect/metaAlias/assert-idp"/>
  </IDPSSODescriptor>
  <Organization>
    <OrganizationName xml:lang="en-us">
      Department of Internal Affairs
    </OrganizationName>
    <OrganizationDisplayName xml:lang="en-us">
      Department of Internal Affairs
    </OrganizationDisplayName>
    <OrganizationURL xml:lang="en-us">
      http://www.dia.govt.nz
    </OrganizationURL>
  </Organization>
  <ContactPerson contactType="support">
    <Company>Department of Internal Affairs</Company>
    <GivenName></GivenName>
    <SurName></SurName>
  </ContactPerson>
```

```
</EntityDescriptor>
```

8.1 Elements in RealMe Identity Provider Metadata

The following table will call out the restrictions placed on RealMe IdP metadata. If not noted here the restrictions will be as per [saml-metadata-2.0-os].

The following restrictions will be placed on the IdP metadata elements and attributes:

8.1.1 Root Element

A SAML v2.0 metadata instance describes either a single entity or multiple entities via use of <EntityDescriptor> or <EntitiesDescriptor> root elements respectively. RealMe supports only the use of single <EntityDescriptor> root elements in a SAML v2.0 metadata file.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<EntityDescriptor>	A single <EntityDescriptor> SHALL be provided. ☑ NZ SAMS Constraint on NZ SAMS: Single <EntityDescriptor> allowed only.	OPTIONAL. NZ SAMS line 171. ☑ SAML v2.0	Must be a root element if <EntitiesDescriptor> is not used. Ref [saml-metadata-2.0-os] section 2.3, line 307.
<Entities Descriptor>	SHALL NOT be provided. ☑ NZ SAMS Constraint on NZ SAMS: <EntitiesDescriptor> not allowed.	OPTIONAL. NZ SAMS line 171. ☑ SAML v2.0	Must be a root element if <EntityDescriptor> is not used. Ref [saml-metadata-2.0-os] section 2.3, line 307.

Table 32

8.1.2 Element <EntityDescriptor>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	SHALL NOT be provided. RealMe SHALL NOT sign metadata. ☑ NZ SAMS Constraint on NZ SAMS: Not provided.	OPTIONAL if the <EntityDescriptor> is not signed. NZ SAMS line 210. ☑ SAML v2.0	Optional. A document-unique identifier for the element, typically used as a reference point when signing. Ref [saml-metadata-2.0-os] section 2.3.2, line 374.
entityID	SHALL be provided. Contains RealMe IdP entityID that SHALL be in an identity privacy domain format as per section 7. ☑ NZ SAMS Constraint on NZ SAMS: Format as per section 7.	REQURIED. NZ SAMS line 209. ☑ SAML v2.0	Required attribute. Specifies the unique identifier of the SAML v2.0 entity whose metadata is described by the element's contents. Ref [saml-metadata-2.0-os] section 2.3.2, line 371.
validUntil	SHALL NOT be provided. RealMe SHALL NOT mandate an expiry time of its IdP metadata to integrated Clients. ☑ NZ SAMS Constraint on NZ SAMS: Expiry of RealMe IdP metadata not stated.	SHOULD be provided for the case of RealMe where the <EntityDescriptor> is the root element. NZ SAMS line 214. ☑ SAML v2.0	Optional attribute indicates the expiration time of the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 376.
cacheDuration	SHALL NOT be provided. ☑ NZ SAMS Constraint on NZ SAMS: CacheDurtion of IdP metadata not stated.	SHOULD be provided for the case of RealMe where the <EntityDescriptor> is the root element. NZ SAMS line 214. ☑ SAML v2.0	Optional attribute indicates the maximum length of time a consumer should cache the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 379.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
<ds:Signature>	SHALL NOT be provided. RealMe SHALL NOT sign metadata. ☑ NZ SAMS Constraint on NZ SAMS: Not provided.	OPTIONAL ☑ SAML v2.0	Optional. An XML signature that authenticates the containing element and its contents. Ref [saml-metadata-2.0-os] section 2.3.2, line 382.
<Extensions>	SHALL NOT be provided. ☑ NZ SAMS	NOT RECOMMENDED and if present will be ignored. NZ SAMS line 222. ☑ SAML v2.0	Optional metadata extensions. Ref [saml-metadata-2.0-os] section 2.3.2, line 385.
<IDPSSO Descriptor>	SHALL be provided ☑ NZ SAMS	For an IdP the <IDPSSO Descriptor> MUST be provided. NZ SAMS lines 223-231. ☑ SAML v2.0	<RoleDescriptor>, <IDPSSODescriptor>, <SPSSODescriptor>, <AuthnAuthorityDescriptor>, <AttributeAuthorityDescriptor>, <PDPDescriptor> [One or More] OR <AffiliationDescriptor> Ref [saml-metadata-2.0-os] section 2.3.2, lines 389-395.
<Organization>	SHALL be provided. ☑ NZ SAMS	REQURIED. NZ SAMS line 232. ☑ SAML v2.0	Optional. An element identifying the organization responsible for the SAML entity described by the element. Ref [saml-metadata-2.0-os] section 2.3.2, line 396.
<ContactPerson>	SHALL be provided. ☑ NZ SAMS	RECOMMENDED that one be included. NZ SAMS line 235.	Optional sequence of elements identifying various kinds of contact personnel. Ref [saml-metadata-2.0-

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
		☑ SAML v2.0	os] section 2.3.2, line 399.
<AdditionalMetadataLocation>	SHALL NOT be provided. ☑ NZ SAMS	MUST NOT be provided. NZ SAMS line 236. ☑ SAML v2.0	Optional sequence of namespace-qualified locations where additional metadata exists for the SAML entity. This may include metadata in alternate formats or describing adherence to other non-SAML specifications. Ref [saml-metadata-2.0-os] section 2.3.2, line 401.

Table 33

8.1.3 Element <IDPSSODescriptor>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
ID	SHALL NOT be provided. RealMe SHALL NOT sign metadata. ☑ NZ SAMS Constraint on NZ SAMS: Not provided.	Per OASIS. ☑ SAML v2.0	Optional. A document-unique identifier for the element, typically used as a reference point when signing. Ref [saml-metadata-2.0-os] section 2.3.2, line 555.
validUntil	SHALL NOT be provided. RealMe SHALL NOT mandate an expiry time of its IdP metadata in the IDPSSODescriptor ☑ NZ SAMS Constraint on NZ SAMS: Expiry of IdP metadata not supported in the	One of validUntil or cacheDuration SHOULD be provided. NZ SAMS line 214. ☑ SAML v2.0	Optional attribute indicates the expiration time of the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 557.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	IDPSSODescriptor		
cacheDuration	SHALL NOT be provided. RealMe SHALL NOT mandate a cache duration of its IdP metadata in the IDPSSODescriptor ☑ NZ SAMS Constraint on NZ SAMS: Caching duration of IdP metadata not supported in the IDPSSODescriptor	One of validUntil or cacheDuration SHOULD be provided. NZ SAMS line 214. ☑ SAML v2.0	Optional attribute indicates the maximum length of time a consumer should cache the metadata contained in the element and any contained elements. Ref [saml-metadata-2.0-os] section 2.3.2, line 560.
WantAuthnRequests Signed	SHALL be the XML data-type of true by section 3.4. ☑ NZ SAMS Constraint on NZ SAMS: Signing <AuthnRequest> is mandatory for RealMe.	Per OASIS. ☑ SAML v2.0	Optional attribute that indicates a requirement for the <samlp:AuthnRequest> messages received by this identity provider to be signed. If omitted, the value is assumed to be false. Ref [saml-metadata-2.0-os] line 687.
protocolSupport Enumeration	SHALL be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Required. A whitespace-delimited set of URIs that identify the set of protocol specifications supported by the role element. For SAML v2.0 entities, this set MUST include the SAML protocol namespace URI `urn:oasis:names:tc:SAML:2.0:protocol`. Ref [saml-metadata-2.0-os] section 2.3.2, line 560.
errorURL	SHALL NOT be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional URI attribute that specifies a location to direct a user for problem

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	Constraint on NZ SAMS: Not provided.		resolution and additional support related to this role. Ref [saml-metadata-2.0-os] section 2.3.2, line 560.
<KeyDescriptor>	SHALL be provided to meet the digital signing requirements in section 4.3. ☑ NZ SAMS	MUST be provided. NZ SAMS line 246. ☑ SAML v2.0	Optional sequence of elements that provides information about the cryptographic keys that the entity uses when acting this role. Ref [saml-metadata-2.0-os] line 579.
<ArtifactResolution Service>	A set of ArtifactResolutionService's SHALL be provided to meet the requirement that the SAML response will be returned over the HTTP-Artifact binding as per section 4.1. These are the endpoints for the SP to resolve artifacts to as in section 4.2.2. See section 4.2.1 for further details on the SAML v2.0 specification on how an ArtifactResolutionService is chosen by the SP based on the value of the artifact. ☑ NZ SAMS	MUST be provided. NZ SAMS line 254. ☑ SAML v2.0	Optional sequence of elements of type IndexedEndpointType that describe indexed endpoints that support the Artifact Resolution profile defined in [saml-profiles-2.0-os]. Ref [saml-metadata-2.0-os] line 648.
<SingleLogoutService>	SHALL NOT be provided. RealMe is not responsible for providing a SingleLogoutService to Clients. ☑ NZ SAMS	MUST be provided if a site supports a single logout service. NZ SAMS line 256.	Optional. Zero or more elements that describe indexed endpoints that support the Single Logout Profiles. Ref [saml-metadata-2.0-os] section 2.3.2, line 652.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
		☑ SAML v2.0	
<ManageNameID Service>	SHALL NOT be provided. ☑ NZ SAMS	SHOULD NOT be provided. NZ SAMS line 260. ☑ SAML v2.0	Optional. Zero or more elements that describe indexed endpoints that support the Name Identifier Management profiles. Ref [saml-metadata-2.0-os] section 2.3.2, line 655.

<NameIDFormat>	SHALL be provided.
	The format urns:
	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
	and
	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
	will be listed.
	This is required to be consistent with 3.2.3.
	
	☑ NZ SAMS
		MAY be provided.
		The `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` format must be listed if the <NameID Format> element is used in the metadata
		
		NZ SAMS line 264.
		
		☑ SAML v2.0
			Optional.
			Zero or more elements that enumerate the name identifier formats supported by this system entity acting in this role.
			
			See Section 8.3 of [saml-core-2.0-os] for some possible values for this element.
			
			Ref [saml-metadata-2.0-os] section 2.3.2, line 658.

<SingleSignOnService>	SHALL be provided. Contains an attribute with specialised requirements, as described below. ☑ NZ SAMS	Per OASIS. NZ SAMS line 272. ☑ SAML v2.0	Mandatory sequence of elements of type EndpointType that describe endpoints that support the profiles of the Authentication Request protocol defined in [saml-profiles-2.0-os]. All identity providers support at least one such endpoint, by definition. Ref [saml-metadata-2.0-os] line 690.
<NameIDMapping Service>	SHALL NOT be provided. ☑ NZ SAMS	SHOULD NOT be provided. NZ SAMS line	Optional. Zero or more elements that describe endpoints that support the Name Identifier

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
		260. ☑ SAML v2.0	Mapping profile. Ref [saml-metadata-2.0-os] line 694.
<AssertionIDRequest Service>	SHALL NOT be provided. ☑ NZ SAMS	MUST NOT be provided. NZ SAMS line 277. ☑ SAML v2.0	Optional. Zero or more elements that describe endpoints that support the profile of the Assertion Request protocol. Ref [saml-metadata-2.0-os] line 698.
<AttributeProfile>	SHALL NOT be provided. RealMe SHALL NOT support the attribute profile. ☑ NZ SAMS Constraint on NZ SAMS: Not provided.	MAY be provided. NZ SAMS line 277. ☑ SAML v2.0	Optional. Zero or more elements that enumerate the attribute profiles supported by this identity provider. Ref [saml-metadata-2.0-os] line 702.
<saml:Attribute>	SHALL NOT be provided. ☑ NZ SAMS	MUST NOT be provided. NZ SAMS line 283. ☑ SAML v2.0	Optional. Zero or more elements that identify the SAML attributes supported by the identity provider. Ref [saml-metadata-2.0-os] line 705.
<ds:Signature>	SHALL NOT be provided. RealMe SHALL NOT sign metadata. ☑ NZ SAMS Constraint on NZ SAMS: Not provided.	OPTIONAL ☑ SAML v2.0	Optional. An XML signature that authenticates the containing element and its contents. Ref [saml-metadata-2.0-os] section 2.3.2, line 572.
<Extensions>	SHALL NOT be provided.	NOT RECOMMENDE	Optional metadata extensions.

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
	☑ NZ SAMS	D and if present will be ignored. NZ SAMS line 222. ☑ SAML v2.0	Ref [saml-metadata-2.0-os] section 2.3.2, line 575.

Table 34

8.1.4 Element <KeyDescriptor>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
use	MUST contain the value 'signing' to meet the digital signing requirements in section 4.2.3.2.4. ☑ NZ SAMS Constraint on NZ SAMS: Must specify 'signing' only	MUST be used. NZ SAMS line 246. ☑ SAML v2.0	Optional attribute specifying the purpose of the key being described. Values are drawn from the KeyTypes enumeration, and consist of the values 'encryption' and 'signing'. Ref [saml-metadata-2.0-os] line 615.
<ds:KeyInfo>	SHALL be provided. SHALL contain RealMe SAML messaging certificate. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Required. The element that identifies the SAML messaging certificates and/or keys. Ref [saml-metadata-2.0-os] line 618.
<EncryptionMethod>	SHALL NOT be provided. RealMe does not perform SAML message encryption. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional element specifying an algorithm and algorithm-specific settings supported by the entity. The exact content varies based on the algorithm supported. Ref [saml-metadata-2.0-os] line 621.

Table 35

8.1.5 Element <ArtifactResolutionService>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
Binding	SHALL use the value: `urn:oasis:names:tc:SAML:2.0:bindings:SOAP` All artifacts shall be resolved over SOAP as per section 4.1. ☑ NZ SAMS Constraint on NZ SAMS: Must specify SOAP only.	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-metadata-2.0-os] line 229.
Location	MUST be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MUST be provided. A required URI attribute that specifies the location of the endpoint. Ref [saml-metadata-2.0-os] line 232.
ResponseLocation	MUST NOT be provided. The attribute is unused for RealMe's profile where only single types of request and response messages are applicable. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional. Used only for protocols or profiles that has more than one type of request or response message. Ref [saml-metadata-2.0-os] line 235.

Table 36

8.1.6 Element <SingleSignOnService>

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement

Attribute / Element	RealMe Requirement	NZ SAMS 1.0 Requirement	OASIS SAML v2.0 Requirement
Binding	SHALL use the value `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect` All authentication requests to RealMe MUST occur over the HTTP-Redirect binding as per section 3.3. ☑ NZ SAMS Constraint on NZ SAMS: Must specify HTTP-Redirect binding only.	Per OASIS. ☑ SAML v2.0	MUST be provided. Ref [saml-metadata-2.0-os] line 229.
Location	MUST be provided. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	MUST be provided. A required URI attribute that specifies the location of the endpoint. Ref [saml-metadata-2.0-os] line 232.
ResponseLocation	MUST NOT be provided. The attribute is unused for RealMe's profile where only single types of request and response messages are applicable. ☑ NZ SAMS	Per OASIS. ☑ SAML v2.0	Optional. Used only for protocols or profiles that has more than one type of request or response message. Ref [saml-metadata-2.0-os] line 235.

Table 37

END

REALME ASSERTION SERVICE SAML V2.0 MESSAGE SPECIFICATION.