

README.zxid

Sampo Kellomäki (sampo@iki.fi)

October 13, 2009

Abstract

ZXID.org Identity Management toolkit implements standalone SAML 2.0 and Liberty ID-WSF 2.0 stacks and aims at implementing all popular federation and ID Web Services protocols. It is a C implementation with minimal external dependencies - OpenSSL, CURL, and zlib - ensuring easy deployment (no DLLhell). Due to its small footprint and efficient and accurate schema driven implementation, it is suitable for embedded and high volume applications. Language bindings to all popular highlevel languages such as PHP, Perl, and Java, are provided via SWIG. ZXID implements, as of July 07, SP, WSC, and WSP roles. IdP role will follow as the project evolves.

ZXID.org ist eine C-Bibliothek, die den vollständigen SAML 2.0-Stack implementiert und alle populären Identitätsverwaltungs-Protokolle wie Liberty ID-FF 1.2, WS-Federation, WS-Trust und ID-Webservices wie Liberty ID-WSF 1.1 und 2.0 implementieren will. Sie beruht auf Schema-basierter Code-Erzeugung, woraus eine genaue Implementation resultiert. SWIG wird verwendet, um Schnittstellen zu Skriptsprachen wie Perl, PHP und Python sowie zu Java bereitzustellen. Sie kann als SP, WSC und WSP fungieren.

A biblioteca de gestão de identidades ZXID.org é uma implementação, em C, das normas SAML 2.0 e Liberty ID-WSF 2.0 com dependências externas mínimas - OpenSSL, CURL, e zlib - facilitando uma implantação fácil sem "inferno dos DLL". Sendo económica em consumo de recursos é indicada para aplicações embutidas ou de grande volume e performance. A biblioteca é disponibilizada para todos os linguagens de programação de alto nível como, p.ex., PHP, Perl, e Java, através de interfaces SWIG. ZXID de hoje (Jul 07) pode funcionar nos papéis SP (Provedor de Serviços), WSC (Cliente de Serviços Web) e WSP (Provedor de Serviços Web), sendo o papel IdP (Provedor de Identidade) suportado na futura evolução do projecto.

La librería de gestión de identidades ZXID.org es una implementación en C de las normas SAML 2.0 y Liberty ID-WSF 2.0, con dependencias externas mínimas - OpenSSL, CURL, y zlib - que elimina el "Infierno DLL" en su implantación. Como ZXID es muy económica, es apta para aplicaciones embebidas o de gran volumen y envergadura. Los lenguajes de programación de alto nivel, como Perl, PHP, y Java, son soportados con generador de interfaces SWIG. Hoy (Feb 07) el ZXID soporta los roles SP (proveedor de servicios) y WSC (cliente de los servicios web). Los roles IdP (proveedor de identidades) y WSP (proveedor de servicios

web) serán soportados en fases futuras del proyecto.

ZXID.org on verkkohenkilöllisyyden ja -tunnisteiden hallintakirjasto joka tukee SAML 2.0 (sisäänkirjaantuminen) ja Liberty ID-WSF 2.0 (henkilöllisyyteen pohjautuvat webbipalvelut) standardeja. ZXID vaatii vain OpenSSL, CURL ja zlib kirjastot joten se välttää "DLL helvetti"-ongelman. Skemapohjaisena C toteutuksena se on tarkka ja taloudellinen ja kelpaa sulautettuihin ja erittäin kovaa suorituskykyä vaativiin sovelluksiin. Se tukee korkeantason kieliä - kuten Perliä, PHP:tä, ja Javaa - SWIG generoiduin rajapinnoin. ZXID tukee (Heinäkuu 07) SP (palveluntarjoaja), WSC (webbipalvelunkutsuja), ja WSP (webbipalveluntarjoaja) rooleja. IdP (henkilöllisyydenvarmentaja) rooli toteutetaan projektin tulevissa vaiheissa.

Contents

1 Other Documentation

This README.zxid is in process of being rewritten and restructured. A lot of the material has moved to specific files, which you should read.

- mod_auth_saml Apache module documentation: SSO without programming.
- zxid_simple() Easy API for SAML
- ZXID Raw API: Program like the pros (and fix your own problems). See also Function Reference
- ZXID ID-WSF API: Make Identity Web Services Calls using ID-WSF
- ZXID Compilation and Installation: Compile and install from source or package. See also INSTALL.zxid for quick overview.
- ZXID Configuration Reference: Nitty gritty on all options.
- ZXID Circle of Trust Reference: How to set up the Circle of Trust, i.e. the partners your web site works with.
- ZXID Logging Reference: ZXID digitally signed logging facility
- javazxid: Using ZXID from Java
- Net::SAML: Using ZXID from Perl
- php_zxid: Using ZXID from PHP
- FAQ: Frequently Asked Questions
- README.smime: Crypto and Cert Tutorial

2 ZXID Project

Web site <http://zxid.org/>

License Open source: Apache 2, see License chapter and file COPYING

Immediate goal: build a SAML 2.0 SP and ID-WSF 2.0 WSC

Goals of ZXID project include

- SOAP 1.1 support (done)
- SAML 2.0 compliance
 - SP role (done)
 - IdP role
- Liberty ID-FF 1.2 support
 - SP
 - IdP
 - SAML 1.1
- Liberty ID-WSF 1.1 support
 - Discovery bootstrap
 - Discovery WSC
 - ID-DAP WSC
 - ID-DAP WSP
- Liberty ID-WSF 2.0 support
 - Discovery bootstrap (done)
 - Discovery WSC (done)
 - ID-DAP WSC (done)
 - ID-DAP WSP

Table 1: ZXID Platform Support

Platform	Native	Cross Compile	Notes
Linux-ix86	gcc-3.4.6	n/a	Development platform
Solaris 8-sparc	gcc-3.4.6	Linux gcc-3.4.6	Fully functional
Windows 2000	-	Linux gcc-3.4.6	Poorly tested
xBSD/Unix	gcc-3.4.6	n/a	C core tested, language bindings not tested

Table 2: ZXID Feature and Language Support (version number indicates last testing)

Feature	C	mod_perl	mod_php	Python	Java/Tomcat	Apache	Shell
Geo Location	Alpha						
ID-MM7	Alpha						
ID-DAP	Beta						
ID-HR-XML	Beta						
Contact Book	Alpha						
People Service	Alpha						
Discovery	Yes						
Web Services (ID-WSF)	Yes						
SSO	0.17	0.17	0.17	Plan	0.17	Plan	0.17

Table 3: ZXID Enabled Application Packages

Application	Language	Notes
DokuWiki	PHP	Patch available, in process of submitting to DokuWiki authors

2.1 Project Layout

Following directory layout is used by the project. Many of the specified directories are used by intermediate outputs that are not distributed in tarball releases, but may or may not be present in CVS checkouts.

```

zxid-0.xx
|
+-- Net      The Net::SAML perl module (also mod_perl)
+-- php      PHP / mod_php integration
+-- zxidjava The Java JNI interface to ZXID
+-- servlet  Apache Tomcat integration
+-- c        C code generated from the Schema Grammar descriptions
+-- sg       Schema Grammar (.sg) descriptions of protocols
+-- xsd      XML schema descriptions of protocols (not distributed)
+-- tex      Temporary files for document generation using PlainDoc (not distributed)
+-- html     HTML documentation generated using PlainDoc
+-- review   Publicly released announcements and documents (not distributed)
+-- t        Test scripts and expected test outputs
`-- tmp      Temporary files, such as actual test outputs

```

The Manifest file, which follows, explains each file in more detail.

```

# zxid/Manifest
# $Id: Manifest,v 1.57 2009-09-07 16:13:02 sampo Exp $
# Packing list for distribution and explanation of files

Manifest      - This file. Describes contents of the distribution.
Changes       - Change log and revision history

```

```

INSTALL.zxid      - Quick installation instructions for the impatient
README.zxid       - How to build and operate ZXID, API documentation
README.zxid-win32 - Windows build notes (preliminary Jan 2007)
zxid-install.pd   - Instructions for Installation from Package or Compilation
zxid-conf.pd      - Instructions for configuration, reference to configuratio\
n options
zxid-cot.pd       - Instructions for creating Circle-of-Trust and manipulin\
g certificates
zxid-simple.pd    - Documentation for ZXID Simple API
zxid-raw.pd       - Documentation for ZXID Raw API
zxid-wsf.pd       - Documentation for ZXID ID-WSF Support
zxid-log.pd       - Documentation on encrypting and signing logging API and A\
udit trail
mod_auth_saml.pd - Apache mod_auth_saml specific instructions
apache.pd         - Apache compilation configuration for mod_php, mod_perl
mediawiki-zxid.pd - Documentation on MediaWiki integration
zxid-java.pd      - Documentation on Java support
zxid-perl.pd      - Documentation on Perl support
zxid-php.pd       - Documentation on PHP support
schemata.pd       - Document summarizing schemata and examples
testplan.pd       - Testing plan
zxid-faq.pd       - Frequently Asked Questions
zxid-license.pd   - Licensing and legal terms chapter for ZXID and dependency\
libraries
zxid-book.pd      - Top level file that pulls together all chapters of ZXID B\
ook
zxid-ref.pd       - Comprehensive list of literature references for ZXID proj\
ect
doc-end.pd        - Formatting code include for documents
doc-inc.pd        - Navigation code include for documents
ref-inc.pd        - Navigation code include for generated reference
LICENSE-2.0.txt   - Apache License v2.0
ca.crt            - Certification Authority certificate for zxid.pem
zxid.pem          - Certificate and privatekey combo for testing (not secure)
favicon.ico       - A ZX/SP favicon for use in demo SPs
gen-consts-from-gperf-output.pl - Used in build process
gen-cot-links.pl - Handy tool for creating documentative symlinks in CoT dir\
ectory
gen-conf-ref.pl   - Generate configuration reference from zxidconf.h
pulverize.pl      - A build tool for generating pulverized libraries for dead\
function elimination
call-anal.pl      - Call graph analysis tool (see make callgraph)
xml-pretty.pl     - XML Pretty Printer

# Handwritten code

Makefile          - Used to build ZXID (needs GNU make)
BSDmakefile       - Trigger gmake on BSD systems
ermac.h           - Error reporting and utility macros
platform.h        - Platform support kludges

```

saml2.h	- SAML related constants
wsf.h	- Liberty ID-WSF related constants
zx.h	- General data structures and prototypes used by generated \ code
zxid.h	- Specific data structures and prototypes for handwritten c \ ode
zxidconf.h	- Configuration parameters and default configuration of ZXID
zxidnoswig.h	- Prototypes that give indigestion to SWIG
zxwsc.h	- Specific data structures and prototypes for Web Services \ Client
aux-templ.c	- Code generation template for auxiliary functions
dec-templ.c	- Code generation template for decoders
enc-templ.c	- Code generation template for encoders
ds-templ.c	- Code generation template for DS script API
getput-templ.c	- Code generation template for accessor functions
zxlog.c	- Logging routines with encryption and signing
zxlogview.c	- Log viewing tool with decryption and sig verification
zxsig.c	- XML DSIG support
zxcrypto.c	- Cryptographical functions
zxlib.c	- Functions used to capture commonalities in generated code
zxns.c	- Namespace manipulation functions for generated code
zxutil.c	- Common library functions used by zx system
zxidcgi.c	- SP specific CGI parsing (see zxid.h)
zxidconf.c	- Configuration file and option parsing (see zxid.h)
zxidpool.c	- Attribute pool management
zxidses.c	- SP session creation, parsing, and destruction (see zxid.h)
zxiduser.c	- Local user account management (see zxid.h)
zxidecp.c	- Enhanced Client Proxy check and functionality (see zxid.h)
zxidcdc.c	- Common Domain Cookie check (see zxid.h)
zxidloc.c	- Service Locator: compute from metadata and input the end \ point to use
zxidlib.c	- Common library functions for SSO (see zxid.h)
zxiddec.c	- Decoding redirect and POST bindings
zxidsp.c	- SP dispatch functions
zxididpx.c	- IdP dispatch functions
zxidmeta.c	- Metadata generation, parsing, and cache
zxidcurl.c	- Glue to libcurl
zxidmk.c	- Handwritten constructors for SSO
zxidmni.c	- NameID Management
zxidslo.c	- Single Logout and other management functions
zxidpep.c	- Policy Enforcement Point functions
zxidpsso.c	- Single Sign-On functions for IdP: Generate A7N
zxidsso.c	- Single Sign-On functions for SP: Consume A7N
zxida7n.c	- Functions for querying assertions
zxidepr.c	- End Point Reference (EPR) and bootstrap handling
zxidwsc.c	- Handwritten ID-WSF Web Services Client engine
zxidmkwsf.c	- Handwritten constructors for WSF
zxidhlowsf.c	- Demonstration of calling ID-WSF services (DS and DAP)
zxidxmltool.c	- Testing tool for parsing XML
zxbench.c	- A benchmarking tool

```
zxencdectest.c - An XML encoding and decoding testing tool
zxidssofinalizetest.c - Test zxid_sso_finalize()
zxdecode.c - SAML redirect and post message decoding tool
zxcot.c - CoT (Circle-of-Trust) management tool: list CoT, add meta\
data to CoT

# Full API demos

zxid.c - Main ZXID SP program (a CGI script)
zxid.pl - SAML 2.0 SP example written in perl
zxid.php - SAML 2.0 SP example written in php
zxid.java - SAML 2.0 SP example written in java (as CGI script)
zxid-java.sh - Shell script for wrapping zxid.java with correct paths
zxidwsctool.c - Command line Web Services Client, a tool for making ID-WS\
F calls

# Simple and Hello World demos

zxidsimp.c - Simple API main definition
zxidhlo.c - Hello World SSO using simple API
zxidsimple.c - Simple API helper program for shell scripts
zxidhlo.sh - Hello World SSO as a shell script
zxidhlo.php - Hello World SSO as a PHP script to run under mod_php
zxidhlocgi.php - Hello World SSO as a PHP script to run as stand alone CGI\
script
zxidhlo.pl - Hello World SSO as a PHP script
zxidhlo-java.sh - Script to set Java environment
zxidhlo.java - Hello World SSO using Java JNI
servlet/WEB-INF/web.xml - Hello World servlet definition
zxidhrxmlwsc.c - Example of ID-SIS HR-XML Web Services Client
zxidhrxmlwsp.c - Example of ID-SIS HR-XML Web Services Provider
zxididp.c - A rudimentary IdP (WIP Nov 2008)
zxidsp.c - A slightly more configurable use of simple API

# S/MIME Utility for Certificate Manipulations, Signing, and Encryption

README.smime - Tutorial on use of smime tool
smimeutil.h
smime-enc.c - Encryption (assymmetric and symmetric) and signing
smime-qry.c - Get string representations of various certificate paramet\
ers
smime-vfy.c - Decryption and signature verification
smime.c - main() of smime command line interface
smimemime.c - Wrap stuff in mime entities
smimeutil.c
certauth.c - Certification authority functions
keygen.c - Key generation functions
logprint.h - Logging macros
macglue.h - Macintosh specific kludges (very old)
pkcs12.c - Import and export PKCS12
```

```

test-smime.pl      - Tests the command line tool
test2-smime.pl    - Tests the SMIMEUtil perl module
filex.pm          - Locking file operations (of generic utility)
tcpcat.pm         - Send and receive data over TCP connections (like http)
send.pl          - Send mail
pass-password.pl  - Demonstrates passing passwords securely
hash-certs.pl    - Hash certs for SSLeay/OpenSSL type certificate directory
smimeutil.i      - SWIG input file to generate SMIMEUtil perl module

# Default Circle of Trust partner IdP's metadata

default-cot/OKCy5mMaXMJUnKQ1wVJCcT00AA8 - auth-int.orange.fr
default-cot/ZLIYSwzbsSQdzIWHISwoWtdrx6JI - auth.orange.fr
default-cot/_CBGcFVVbIEmt5oh3jUx4GEfHLM - idp.symdemo.com
default-cot/s36Te-rgbzReSjVc8vDDGy89tT8 - http://idp.ssocircle.com

# Module generation facilities

phpzxid.i         - SWIG input file for php_zxid.so PHP extension
pyzxid.i         - SWIG input file for py_zxid.so Python extension
rubyzxid.i       - SWIG input file for ruby_zxid.so Ruby extension
csharpzxid.i     - SWIG input file for csharp_zxid.so C# extension
javafxid.i       - SWIG input file for libzxidjni.so Java JNI extension
wsc.i           - SWIG input file for Net::WSF::WSC perl module
wsfraw.i        - SWIG input file for Net::WSF::Raw perl module
zxid.i          - SWIG input file for Net::SAML perl module
zxidmd.i        - SWIG input file for Net::SAML::Metadata perl module
zxidraw.i       - SWIG input file for Net::SAML::Raw perl module

# Schema grammar descriptions (used as input to code generation)

sg/liberty-authentication-context-v2.0.sg
sg/liberty-idff-protocols-schema-1.2-errata-v2.0.sg
sg/liberty-idff-utility-v1.0.sg
sg/liberty-idwsf-disco-svc-v1.2.sg
sg/liberty-idwsf-disco-svc-v2.0.sg
sg/liberty-idwsf-interaction-svc-v1.1.sg
sg/liberty-idwsf-interaction-svc-v2.0.sg
sg/liberty-idwsf-security-mechanisms-v1.2.sg
sg/liberty-idwsf-security-mechanisms-v2.0.sg
sg/liberty-idwsf-soap-binding-v1.2.sg
sg/liberty-idwsf-soap-binding-v2.0.sg
sg/liberty-idwsf-soap-binding.sg          - Framework SOAP header
sg/liberty-idwsf-utility-1.0-errata-v1.0.sg
sg/liberty-idwsf-utility-v1.1.sg
sg/liberty-idwsf-utility-v2.0.sg
sg/liberty-metadata-v2.0.sg
sg/liberty-utility-v2.0.sg
sg/oasis-sstc-saml-schema-assertion-1.1.sg
sg/oasis-sstc-saml-schema-protocol-1.1.sg

```

```

sg/saml-schema-assertion-2.0.sg
sg/saml-schema-metadata-2.0.sg
sg/saml-schema-protocol-2.0.sg
sg/saml-schema-ecp-2.0.sg
sg/liberty-paos-v2.0.sg
sg/ws-addr-1.0.sg
sg/wsf-soap11.sg      - Mega SOAP parser for SAML and ID-WSF messages
sg/wss-secext-1.0.sg
sg/wss-util-1.0.sg
sg/xenc-schema.sg
sg/xmldsig-core.sg
sg/ec.sg              - IncludedNamespaces from Exclusive Canonicalization
sg/xml.sg
sg/xsi.sg
sg/xs.sg
sg/id-dap.sg          - ID Directory Access Protocol
sg/lib-id-sis-cb-proto.sg - Contact Book Protocol
sg/lib-id-sis-cb-cdm.sg - Contact Book Conceptual Data Model
sg/liberty-id-sis-gl-v1.0-14.sg - Geo Location Service
sg/id-mm7-R6-1-4.sg
sg/liberty-idwsf-dst-v2.0.sg      - DST 2.0
sg/liberty-idwsf-dst-dt-v2.0.sg - DST 2.0 data types
sg/liberty-idwsf-subst-ref-v1.0.sg
sg/liberty-idwsf-subst-v1.0.sg
sg/liberty-idwsf-dst-v2.1.sg
sg/liberty-idwsf-idmapping-svc-v2.0.sg
sg/liberty-idwsf-people-service-v1.0.sg
sg/liberty-idwsf-authn-svc-v2.0.sg
sg/access_control-xacml-2.0-context-schema-os.sg
sg/access_control-xacml-2.0-policy-schema-os.sg
sg/access_control-xacml-2.0-saml-assertion-schema-os.sg
sg/access_control-xacml-2.0-saml-protocol-schema-os.sg
sg/ws-trust-1.3.sg
sg/ws-policy.sg
sg/ws-secureconversation-1.3.sg

# Schema generated C code (see also Makefile if you add files)

c/license.c - Generated file: License string
c/zxidvers.h - Generated file: version string

c/zx-attrs.c - Generated: Mapping of a string to attribute token
c/zx-aux.c - Generated from aux-templ.c and various .sg files
c/zx-const.h - Generated: Token value constants
c/zx-data.h - Generated: Data structures reflecting schemata (.sg files). \
Root object.
c/zx-dec.c - Generated from dec-templ.c and various .sg files. The root d\
ecoder.
c/zx-elems.c - Generated: Mapping of a string to element token
c/zx-enc.c - Generated from enc-templ.c and various .sg files. The root e\

```

ncoder.

c/zx-getput.c - Generated from getput-templ.c and various .sg files
 c/zx-ns.c - Generated: initializations of namespace tables
 c/zx-ns.h - Generated: namespace constant and macro definitions

c/zx-a-data.h - Generated: Web Services Addressing data structures
 c/zx-a-aux.c - Generated from aux-templ.c: WS-Addr aux functions
 c/zx-a-dec.c - Generated from dec-templ.c: WS-Addr decoders
 c/zx-a-enc.c - Generated from enc-templ.c: WS-Addr encoders
 c/zx-a-getput.c - Generated from getput-templ.c

c/zx-ac-data.h;	c/zx-ac-aux.c;	c/zx-ac-dec.c;	c/zx-ac-enc.c;	c/z\
x-ac-getput.c				
c/zx-b-data.h;	c/zx-b-aux.c;	c/zx-b-dec.c;	c/zx-b-enc.c;	c/z\
x-b-getput.c				
c/zx-b12-data.h;	c/zx-b12-aux.c;	c/zx-b12-dec.c;	c/zx-b12-enc.c;	c/z\
x-b12-getput.c				
c/zx-di-data.h;	c/zx-di-aux.c;	c/zx-di-dec.c;	c/zx-di-enc.c;	c/z\
x-di-getput.c				
c/zx-di12-data.h;	c/zx-di12-aux.c;	c/zx-di12-dec.c;	c/zx-di12-enc.c;	c/z\
x-di12-getput.c				
c/zx-ds-data.h;	c/zx-ds-aux.c;	c/zx-ds-dec.c;	c/zx-ds-enc.c;	c/z\
x-ds-getput.c				
c/zx-e-data.h;	c/zx-e-aux.c;	c/zx-e-dec.c;	c/zx-e-enc.c;	c/z\
x-e-getput.c				
c/zx-ff12-data.h;	c/zx-ff12-aux.c;	c/zx-ff12-dec.c;	c/zx-ff12-enc.c;	c/z\
x-ff12-getput.c				
c/zx-is-data.h;	c/zx-is-aux.c;	c/zx-is-dec.c;	c/zx-is-enc.c;	c/z\
x-is-getput.c				
c/zx-is12-data.h;	c/zx-is12-aux.c;	c/zx-is12-dec.c;	c/zx-is12-enc.c;	c/z\
x-is12-getput.c				
c/zx-lu-data.h;	c/zx-lu-aux.c;	c/zx-lu-dec.c;	c/zx-lu-enc.c;	c/z\
x-lu-getput.c				
c/zx-m20-data.h;	c/zx-m20-aux.c;	c/zx-m20-dec.c;	c/zx-m20-enc.c;	c/z\
x-m20-getput.c				
c/zx-md-data.h;	c/zx-md-aux.c;	c/zx-md-dec.c;	c/zx-md-enc.c;	c/z\
x-md-getput.c				
c/zx-ecp-data.h;	c/zx-ecp-aux.c;	c/zx-ecp-dec.c;	c/zx-ecp-enc.c;	c/z\
x-ecp-getput.c				
c/zx-paos-data.h;	c/zx-paos-aux.c;	c/zx-paos-dec.c;	c/zx-paos-enc.c;	c/z\
x-paos-getput.c				
c/zx-sa-data.h;	c/zx-sa-aux.c;	c/zx-sa-dec.c;	c/zx-sa-enc.c;	c/z\
x-sa-getput.c				
c/zx-sa11-data.h;	c/zx-sa11-aux.c;	c/zx-sa11-dec.c;	c/zx-sa11-enc.c;	c/z\
x-sa11-getput.c				
c/zx-sbf-data.h;	c/zx-sbf-aux.c;	c/zx-sbf-dec.c;	c/zx-sbf-enc.c;	c/z\
x-sbf-getput.c				
c/zx-sec-data.h;	c/zx-sec-aux.c;	c/zx-sec-dec.c;	c/zx-sec-enc.c;	c/z\
x-sec-getput.c				
c/zx-sec12-data.h;	c/zx-sec12-aux.c;	c/zx-sec12-dec.c;	c/zx-sec12-enc.c;	c/z\

x-sec12-getput.c				
c/zx-sp-data.h;	c/zx-sp-aux.c;	c/zx-sp-dec.c;	c/zx-sp-enc.c;	c/z\
x-sp-getput.c				
c/zx-sp11-data.h;	c/zx-sp11-aux.c;	c/zx-sp11-dec.c;	c/zx-sp11-enc.c;	c/z\
x-sp11-getput.c				
c/zx-wsse-data.h;	c/zx-wsse-aux.c;	c/zx-wsse-dec.c;	c/zx-wsse-enc.c;	c/z\
x-wsse-getput.c				
c/zx-wsu-data.h;	c/zx-wsu-aux.c;	c/zx-wsu-dec.c;	c/zx-wsu-enc.c;	c/z\
x-wsu-getput.c				
c/zx-xenc-data.h;	c/zx-xenc-aux.c;	c/zx-xenc-dec.c;	c/zx-xenc-enc.c;	c/z\
x-xenc-getput.c				
c/zx-exca-data.h;	c/zx-exca-aux.c;	c/zx-exca-dec.c;	c/zx-exca-enc.c;	c/z\
x-exca-getput.c				
c/zx-xsi-data.h;	c/zx-xsi-aux.c;	c/zx-xsi-dec.c;	c/zx-xsi-enc.c;	c/z\
x-xsi-getput.c				
c/zx-xs-data.h;	c/zx-xs-aux.c;	c/zx-xs-dec.c;	c/zx-xs-enc.c;	c/z\
x-xs-getput.c				
c/zx-xml-data.h;	c/zx-xml-aux.c;	c/zx-xml-dec.c;	c/zx-xml-enc.c;	c/z\
x-xml-getput.c				
c/zx-dap-data.h;	c/zx-dap-aux.c;	c/zx-dap-dec.c;	c/zx-dap-enc.c;	c/z\
x-dap-getput.c				
c/zx-ps-data.h;	c/zx-ps-aux.c;	c/zx-ps-dec.c;	c/zx-ps-enc.c;	c/z\
x-ps-getput.c				
c/zx-im-data.h;	c/zx-im-aux.c;	c/zx-im-dec.c;	c/zx-im-enc.c;	c/z\
x-im-getput.c				
c/zx-as-data.h;	c/zx-as-aux.c;	c/zx-as-dec.c;	c/zx-as-enc.c;	c/z\
x-as-getput.c				
c/zx-subst-data.h;	c/zx-subst-aux.c;	c/zx-subst-dec.c;	c/zx-subst-enc.c;	c/z\
x-subst-getput.c				
c/zx-dst-data.h;	c/zx-dst-aux.c;	c/zx-dst-dec.c;	c/zx-dst-enc.c;	c/z\
x-dst-getput.c				
c/zx-cb-data.h;	c/zx-cb-aux.c;	c/zx-cb-dec.c;	c/zx-cb-enc.c;	c/z\
x-cb-getput.c				
c/zx-cdm-data.h;	c/zx-cdm-aux.c;	c/zx-cdm-dec.c;	c/zx-cdm-enc.c;	c/z\
x-cdm-getput.c				
c/zx-gl-data.h;	c/zx-gl-aux.c;	c/zx-gl-dec.c;	c/zx-gl-enc.c;	c/z\
x-gl-getput.c				
c/zx-mm7-data.h;	c/zx-mm7-aux.c;	c/zx-mm7-dec.c;	c/zx-mm7-enc.c;	c/z\
x-mm7-getput.c				
c/zx-xa-data.h;	c/zx-xa-aux.c;	c/zx-xa-dec.c;	c/zx-xa-enc.c;	c/z\
x-xa-getput.c				
c/zx-xac-data.h;	c/zx-xac-aux.c;	c/zx-xac-dec.c;	c/zx-xac-enc.c;	c/z\
x-xac-getput.c				
c/zx-xasa-data.h;	c/zx-xasa-aux.c;	c/zx-xasa-dec.c;	c/zx-xasa-enc.c;	c/z\
x-xasa-getput.c				
c/zx-xasp-data.h;	c/zx-xasp-aux.c;	c/zx-xasp-dec.c;	c/zx-xasp-enc.c;	c/z\
x-xasp-getput.c				
c/zx-wst-data.h;	c/zx-wst-aux.c;	c/zx-wst-dec.c;	c/zx-wst-enc.c;	c/z\

```

x-wst-getput.c
c/zx-wsp-data.h;   c/zx-wsp-aux.c;   c/zx-wsp-dec.c;   c/zx-wsp-enc.c;   c/z\
x-wsp-getput.c
c/zx-wsc-data.h;   c/zx-wsc-aux.c;   c/zx-wsc-dec.c;   c/zx-wsc-enc.c;   c/z\
x-wsc-getput.c

# Advanced Client

c/zx-dp-data.h;   c/zx-dp-aux.c;   c/zx-dp-dec.c;   c/zx-dp-enc.c;   c/z\
x-dp-getput.c
c/zx-pmm-data.h;   c/zx-pmm-aux.c;   c/zx-pmm-dec.c;   c/zx-pmm-enc.c;   c/z\
x-pmm-getput.c
c/zx-prov-data.h;   c/zx-prov-aux.c;   c/zx-prov-dec.c;   c/zx-prov-enc.c;   c/z\
x-prov-getput.c
c/zx-idp-data.h;   c/zx-idp-aux.c;   c/zx-idp-dec.c;   c/zx-idp-enc.c;   c/z\
x-idp-getput.c
c/zx-shps-data.h;   c/zx-shps-aux.c;   c/zx-shps-dec.c;   c/zx-shps-enc.c;   c/z\
x-shps-getput.c

# Unofficial stuff

c/zx-hrxml-data.h; c/zx-hrxml-aux.c; c/zx-hrxml-dec.c; c/zx-hrxml-enc.c; c/z\
x-hrxml-getput.c
c/zx-idhrxml-data.h; c/zx-idhrxml-aux.c; c/zx-idhrxml-dec.c; c/zx-idhrxml-en\
c.c; c/zx-idhrxml-getput.c
c/zx-demomed-data.h; c/zx-demomed-aux.c; c/zx-demomed-dec.c; c/zx-demomed-en\
c.c; c/zx-demomed-getput.c

# Expected output for various test cases

t/authnreq.xml
t/se-req.xml
t/se-req2.xml
t/se-resp.xml
t/se-artif-resp.xml - Example response to artifact resolution. Shows SSO \
assertion.
t/sso-w-bootstraps.xml - Example response to artifact resolution. Shows SSO \
assertion w/bootstraps
t/x509.xml - Example of ID-WSF SOAP call using x509 sec mech
t/bin-bearer.xml - Example of ID-WSF SOAP call using bearer token (bin\
ary) sec mech
t/saml-bearer.xml - Example of ID-WSF SOAP call using bearer token (SAM\
L2) sec mech

# Apache authentication module

mod_auth_saml.c - Apache auth module to SAML protect web pages
protected/content.txt - Test content for mod_auth_saml
protected/saml - Test content for mod_auth_saml
protected/orange.cgi - Demonstration of using Orange Personal APIs

```

```
protected/protected.html
pers/personalized.html
pers/env.cgi
intra/intranet.html
intra/env.cgi
strong/strong.html
other/other.html
idpsel.cgi          - Custom IdP selection script for the strong area.

# Net::SAML module (zxid.i)

Net/README.zxid-perl
Net/Makefile.PL    - How to build the module
Net/SAML.pod       - Bare bones documentation
Net/SAML.pm        - Generated with SWIG from zxid.i and headers
Net/SAML_wrap.c    - Generated with SWIG from zxid.i and headers
Net/test.pl        - Unit tests

# php_zxid.so PHP extension (phpzxid.i)

php/README.zxid-php
php/php_zxid.h     - Generated
php/zxid.php       - Generated
php/zxid_wrap.c    - Generated
php/zxid.ini

# py_zxid.so Python extension (pyzxid.i)

py/README.zxid-py
py/zxid.py         - Generated
py/zxid_wrap.c    - Generated

# ruby_zxid.so Ruby extension (rubyzxid.i)

ruby/README.zxid-ruby
#ruby/zxid.ruby   - Generated
ruby/zxid_wrap.c  - Generated

# csharp_zxid.so C# extension (csharpzxid.i)

csharp/README.zxid-csharp
csharp/zxid.cs    - Generated
csharp/zxid_wrap.c - Generated

csharp/SWIGTYPE_p_X509.cs
csharp/SWIGTYPE_p_f_p_void__void.cs
csharp/SWIGTYPE_p_f_p_void_size_t__p_void.cs
csharp/SWIGTYPE_p_f_size_t__p_void.cs
csharp/SWIGTYPE_p_int.cs
csharp/SWIGTYPE_p_p_char.cs
```

csharp/SWIGTYPE_p_p_void.cs
csharp/SWIGTYPE_p_p_zx_ns_s.cs
csharp/SWIGTYPE_p_p_zx_xenc_EncryptedKey_s.cs
csharp/SWIGTYPE_p_time_t.cs
csharp/SWIGTYPE_p_timeval.cs
csharp/SWIGTYPE_p_unsigned_char.cs
csharp/SWIGTYPE_p_void.cs
csharp/SWIGTYPE_p_zx_a_Address_s.cs
csharp/SWIGTYPE_p_zx_a_EndpointReference_s.cs
csharp/SWIGTYPE_p_zx_any_attr_s.cs
csharp/SWIGTYPE_p_zx_any_elem_s.cs
csharp/SWIGTYPE_p_zx_ctx.cs
csharp/SWIGTYPE_p_zx_dap_QueryItem_s.cs
csharp/SWIGTYPE_p_zx_dap_Query_s.cs
csharp/SWIGTYPE_p_zx_dap_ResultQuery_s.cs
csharp/SWIGTYPE_p_zx_dap_Select_s.cs
csharp/SWIGTYPE_p_zx_dap_Subscription_s.cs
csharp/SWIGTYPE_p_zx_dap_TestItem_s.cs
csharp/SWIGTYPE_p_zx_dap_TestOp_s.cs
csharp/SWIGTYPE_p_zx_di_Query_s.cs
csharp/SWIGTYPE_p_zx_ds_KeyInfo_s.cs
csharp/SWIGTYPE_p_zx_ds_Reference_s.cs
csharp/SWIGTYPE_p_zx_ds_Signature_s.cs
csharp/SWIGTYPE_p_zx_e_Body_s.cs
csharp/SWIGTYPE_p_zx_e_Envelope_s.cs
csharp/SWIGTYPE_p_zx_e_Header_s.cs
csharp/SWIGTYPE_p_zx_elem_s.cs
csharp/SWIGTYPE_p_zx_ff12_Assertion_s.cs
csharp/SWIGTYPE_p_zx_md_AssertionConsumerService_s.cs
csharp/SWIGTYPE_p_zx_md_EntityDescriptor_s.cs
csharp/SWIGTYPE_p_zx_md_KeyDescriptor_s.cs
csharp/SWIGTYPE_p_zx_md_ManageNameIDService_s.cs
csharp/SWIGTYPE_p_zx_md_SPSSODescriptor_s.cs
csharp/SWIGTYPE_p_zx_md_SingleLogoutService_s.cs
csharp/SWIGTYPE_p_zx_node_s.cs
csharp/SWIGTYPE_p_zx_ns_s.cs
csharp/SWIGTYPE_p_zx_root_s.cs
csharp/SWIGTYPE_p_zx_sall_Assertion_s.cs
csharp/SWIGTYPE_p_zx_sa_Assertion_s.cs
csharp/SWIGTYPE_p_zx_sa_Attribute_s.cs
csharp/SWIGTYPE_p_zx_sa_EncryptedID_s.cs
csharp/SWIGTYPE_p_zx_sa_Issuer_s.cs
csharp/SWIGTYPE_p_zx_sa_NameID_s.cs
csharp/SWIGTYPE_p_zx_sp_ArtifactResolve_s.cs
csharp/SWIGTYPE_p_zx_sp_AuthnRequest_s.cs
csharp/SWIGTYPE_p_zx_sp_LogoutRequest_s.cs
csharp/SWIGTYPE_p_zx_sp_LogoutResponse_s.cs
csharp/SWIGTYPE_p_zx_sp_ManageNameIDRequest_s.cs
csharp/SWIGTYPE_p_zx_sp_ManageNameIDResponse_s.cs
csharp/SWIGTYPE_p_zx_sp_NewEncryptedID_s.cs

```
csharp/SWIGTYPE_p_zx_sp_Status_s.cs
csharp/SWIGTYPE_p_zx_str.cs
csharp/SWIGTYPE_p_zx_tok.cs
csharp/SWIGTYPE_p_zx_xenc_EncryptedData_s.cs
csharp/SWIGTYPE_p_zx_xenc_EncryptedKey_s.cs
csharp/SWIGTYPE_p_zxid_cgi.cs
csharp/SWIGTYPE_p_zxid_conf.cs
csharp/SWIGTYPE_p_zxid_curl_ctx.cs
csharp/SWIGTYPE_p_zxid_entity.cs
csharp/SWIGTYPE_p_zxid_ses.cs
csharp/SWIGTYPE_p_zxsig_ref.cs
csharp/zxidPINVOKE.cs

# libzxidjni.so Java JNI extension (javazxid.i)

zxidjava/README.zxid-java
zxidjava/zxid_wrap.c

zxidjava/SWIGTYPE_p_X509.java
zxidjava/SWIGTYPE_p_f_p_void__void.java
zxidjava/SWIGTYPE_p_f_p_void_size_t__p_void.java
zxidjava/SWIGTYPE_p_f_size_t__p_void.java
zxidjava/SWIGTYPE_p_fdtype.java
zxidjava/SWIGTYPE_p_int.java
zxidjava/SWIGTYPE_p_p_char.java
zxidjava/SWIGTYPE_p_p_void.java
zxidjava/SWIGTYPE_p_p_zx_ns_s.java
zxidjava/SWIGTYPE_p_p_zx_xenc_EncryptedKey_s.java
zxidjava/SWIGTYPE_p_time_t.java
zxidjava/SWIGTYPE_p_timeval.java
zxidjava/SWIGTYPE_p_unsigned_char.java
zxidjava/SWIGTYPE_p_void.java
zxidjava/SWIGTYPE_p_zx_a_Address_s.java
zxidjava/SWIGTYPE_p_zx_a_EndpointReference_s.java
zxidjava/SWIGTYPE_p_zx_dap_QueryItem_s.java
zxidjava/SWIGTYPE_p_zx_dap_Query_s.java
zxidjava/SWIGTYPE_p_zx_dap_ResultQuery_s.java
zxidjava/SWIGTYPE_p_zx_dap_Select_s.java
zxidjava/SWIGTYPE_p_zx_dap_Subscription_s.java
zxidjava/SWIGTYPE_p_zx_dap_TestItem_s.java
zxidjava/SWIGTYPE_p_zx_dap_TestOp_s.java
zxidjava/SWIGTYPE_p_zx_di_Query_s.java
zxidjava/SWIGTYPE_p_zx_ds_KeyInfo_s.java
zxidjava/SWIGTYPE_p_zx_ds_Reference_s.java
zxidjava/SWIGTYPE_p_zx_ds_Signature_s.java
zxidjava/SWIGTYPE_p_zx_e_Body_s.java
zxidjava/SWIGTYPE_p_zx_e_Envelope_s.java
zxidjava/SWIGTYPE_p_zx_e_Header_s.java
zxidjava/SWIGTYPE_p_zx_ff12_Assertion_s.java
zxidjava/SWIGTYPE_p_zx_md_ArtifactResolutionService_s.java
```

zxidjava/SWIGTYPE_p_zx_md_AssertionConsumerService_s.java
zxidjava/SWIGTYPE_p_zx_md_EntityDescriptor_s.java
zxidjava/SWIGTYPE_p_zx_md_IDPSSODescriptor_s.java
zxidjava/SWIGTYPE_p_zx_md_KeyDescriptor_s.java
zxidjava/SWIGTYPE_p_zx_md_ManageNameIDService_s.java
zxidjava/SWIGTYPE_p_zx_md_SPSSODescriptor_s.java
zxidjava/SWIGTYPE_p_zx_md_SingleLogoutService_s.java
zxidjava/SWIGTYPE_p_zx_md_SingleSignOnService_s.java
zxidjava/SWIGTYPE_p_zx_root_s.java
zxidjava/SWIGTYPE_p_zx_sall_Assertion_s.java
zxidjava/SWIGTYPE_p_zx_sa_Assertion_s.java
zxidjava/SWIGTYPE_p_zx_sa_AttributeStatement_s.java
zxidjava/SWIGTYPE_p_zx_sa_Attribute_s.java
zxidjava/SWIGTYPE_p_zx_sa_AuthnStatement_s.java
zxidjava/SWIGTYPE_p_zx_sa_EncryptedAssertion_s.java
zxidjava/SWIGTYPE_p_zx_sa_EncryptedID_s.java
zxidjava/SWIGTYPE_p_zx_sa_Issuer_s.java
zxidjava/SWIGTYPE_p_zx_sa_NameID_s.java
zxidjava/SWIGTYPE_p_zx_sa_Subject_s.java
zxidjava/SWIGTYPE_p_zx_sp_ArtifactResolve_s.java
zxidjava/SWIGTYPE_p_zx_sp_AuthnRequest_s.java
zxidjava/SWIGTYPE_p_zx_sp_LogoutRequest_s.java
zxidjava/SWIGTYPE_p_zx_sp_LogoutResponse_s.java
zxidjava/SWIGTYPE_p_zx_sp_ManageNameIDRequest_s.java
zxidjava/SWIGTYPE_p_zx_sp_ManageNameIDResponse_s.java
zxidjava/SWIGTYPE_p_zx_sp_NewEncryptedID_s.java
zxidjava/SWIGTYPE_p_zx_sp_Response_s.java
zxidjava/SWIGTYPE_p_zx_sp_Status_s.java
zxidjava/SWIGTYPE_p_zx_xac_Attribute_s.java
zxidjava/SWIGTYPE_p_zx_xac_Response_s.java
zxidjava/SWIGTYPE_p_zx_xasa_XACMLAuthzDecisionStatement_s.java
zxidjava/SWIGTYPE_p_zx_xasp_XACMLAuthzDecisionQuery_s.java
zxidjava/SWIGTYPE_p_zx_xenc_EncryptedData_s.java
zxidjava/SWIGTYPE_p_zx_xenc_EncryptedKey_s.java
zxidjava/zx_any_attr_s.java
zxidjava/zx_any_elem_s.java
zxidjava/zx_ctx.java
zxidjava/zx_elem_s.java
zxidjava/zx_node_s.java
zxidjava/zx_ns_s.java
zxidjava/zx_str.java
zxidjava/zx_tok.java
zxidjava/zxid_atsrc.java
zxidjava/zxid_attr.java
zxidjava/zxid_cgi.java
zxidjava/zxid_conf.java
zxidjava/zxid_cstr_list.java
zxidjava/zxid_curl_ctx.java
zxidjava/zxid_entity.java
zxidjava/zxid_map.java

```

zxidjava/zxid_need.java
zxidjava/zxid_ses.java
zxidjava/zxid_wrap.c
zxidjava/zxidjni.java
zxidjava/zxidjniConstants.java
zxidjava/zxidjniJNI.java
zxidjava/zxsig_ref.java

# Precheck. These are build time tests for dependency libraries.

precheck/chk-zlib.c
precheck/chk-openssl.c
precheck/chk-curl.c
precheck/chk-apache.c

#EOF

```

2.2 Protocol Encoders and Decoders

The protocol encoders and decoders are generated automatically from the schema grammar (.sg) descriptions. This ensures accurate protocol implementation. While the output is strictly schema driven and correct, the decoders have some provisions to accept some deviations from strict spec (e.g. out of order elements are tolerated). However, one should note that XMLDSIG does not tolerate very much deviation, thus even if decoder accepts a slightly illformed message, it is likely to fail in signature verification.

There are three outputs from generation

1. Data structures describing the data (xx.h)
2. Encoder that linearizes the data structure to wire protocol (xx-enc.c)
3. Decoder that converts wire protocol byte stream to a data structure (xx-dec.c)

2.3 Standards and Namespaces

ZXID uses consistently the same namespace prefixes throughout the project. The generated encoders and decoders support following schemata

Table 4: ZXID Namespace Convention

Prefix	URI	Description
sa	urn:oasis:names:tc:SAML:2.0:assertion	SAML 2.0
sp	urn:oasis:names:tc:SAML:2.0:protocol	
md	urn:oasis:names:tc:SAML:2.0:metadata	
ecp	urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp	

Table 4: (continuation)

paos	urn:liberty:paos:2006-08	
sa11	urn:oasis:names:tc:SAML:1.0:assertion	SAML 1.1
sp11	urn:oasis:names:tc:SAML:1.0:protocol	
ff12	urn:liberty:iff:2003-08	ID-FF 1.2
m20	urn:liberty:metadata:2004-12	v2.0 (almost same as 1.2)
ac	urn:liberty:ac:2004-12	v2.0 (almost same as 1.2)
b12	urn:liberty:sb:2003-08	ID-WSF 1.1 SOAP Binding
sec12	urn:liberty:sec:2003-08	ID-WSF 1.1 Security Mechanisms
di12	urn:liberty:disco:2003-08	ID-WSF 1.1 Discovery Service
is12	urn:liberty:is:2003-08	ID-WSF 1.1 Interaction Service
lu	urn:liberty:util:2006-08	ID-WSF 2.0 Utility Schema
sbf	urn:liberty:sb	Framework header
b	urn:liberty:sb:2006-08	ID-WSF 2.0 SOAP Binding
sec	urn:liberty:security:2006-08	ID-WSF 2.0 Security Mechanisms
di	urn:liberty:disco:2006-08	ID-WSF 2.0 Discovery Service
is	urn:liberty:is:2006-08	ID-WSF 2.0 Interaction Service
dap	urn:liberty:id-sis-dap:2006-08:dst-2.1	ID Directory Access Protocol
dst	urn:liberty:dst:2006-08	Data Services Template 2.1
subs	urn:liberty:ssos:2006-08	Subscription and Notification
ps	urn:liberty:ps:2006-08	People Service
im	urn:liberty:ims:2006-08	Identity Mapping svc (aka Token Map)
as	urn:liberty:sa:2006-08	ID-WSF 2.0 Authentication Service
cb	urn:liberty:id-sis-cb:2004-10	Contact Book Protocol (DST 2.0 based)
cdm	urn:liberty:cb:conceptual-data-model:2004-10	Contact Book Common Data Model
gl	urn:liberty:id-sis-gl:2005-07	Geolocation Service
mm7	http://www.3gpp.org/ftp/Specs/archive/23_series/23.110/ID-MM7-1-4	ID-MM7 (aka ID-SIS-CSM) MM7-1-4
dp	urn:liberty:dp:2006-12	ID-WSF 2.0 Design Patterns
idp	urn:liberty:idp:2006-12	ID-WSF 2.0 IdP as web svc
pmm	urn:liberty:pmm:2006-12	ID-WSF 2.0 Prov Mod Mgr
prov	urn:liberty:prov:2006-12	ID-WSF 2.0 TM Provisioning
shps	urn:liberty:shps:2006-12	ID-WSF 2.0 Svc Handling and Proxying
e	http://schemas.xmlsoap.org/soap/envelope/	SOAP 1.1, with SAML and WSF
xa	urn:oasis:names:tc:xacml:2.0:policy:schema:os	XACML 2.0
xac	urn:oasis:names:tc:xacml:2.0:context:schema:os	
xasp	urn:oasis:xacml:2.0:saml:protocol:schema:os	
xasa	urn:oasis:xacml:2.0:saml:assertion:schema:os	
wst	http://docs.oasis-open.org/ws-sx/ws-trust/200512/	WS-Trust 1.3 CD-01
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	*** Newer version? http://www.w3.org/ns/ws-policy/
wsc	http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/	WS-SecureConversation CD-01
ds	http://www.w3.org/2000/09/xmldsig#	XML Signatures
xenc	http://www.w3.org/2001/04/xmlenc#	XML Encryption
exca	http://www.w3.org/2001/10/xml-exc-c14n#	Exclusive Canonicalization
a	http://www.w3.org/2005/08/addressing	WSA 1.0
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-security-exct-1.0.xsd	WS-SecuritySecExt-1.0
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-wss-security-utility-1.0.xsd	WS-SecurityUtility-1.0

Table 4: (continuation)

xml	http://www.w3.org/XML/1998/namespace	http://www.w3.org/2001/xml.xsd
xsi	http://www.w3.org/2001/XMLSchema-instance	
xs	http://www.w3.org/2001/XMLSchema	Namespace only, no code
igf0 urn:LibertyAlliance:igf:0.3:carml Early draft 01, WIP	carml0 urn:LibertyAlliance:igf:0.3:carml Early draft03,WIP:igf:0.3:core	

3 License

Copyright (c) 2006-2009 Symlabs (symlabs@symlabs.com), All Rights Reserved. Author: Sampo Kellomäki (sampo@iki.fi)

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement number 216287 (TAS3 - Trusted Architecture for Securely Shared Services - www.tas3.eu).

While the source distribution of ZXID does not contain SSLeay or OpenSSL code, if you use this code you will use OpenSSL library. Please give Eric Young and OpenSSL team credit (as required by their licenses).

Binary distribution of this product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). See LICENSE.openssl for further information.

Binary distribution of this product includes cryptographic software written by Eric Young (eay@cryptsoft.com). Binary distribution of this product includes software written by Tim Hudson (tjh@cryptsoft.com). See LICENSE.ssleay for further information.

And remember, you, and nobody else but you, are responsible for auditing ZXID and OpenSSL library for security problems, back-doors, and general suitability for your application.

3.1 Dependency Library Licenses

ZXID strives to maintain IPR hygiene and avoid both non-free and GPL license contamination. All the dependency libraries have BSD style licenses

- OpenSSL under BSDish (with "advertising" clause)
- libcurl under BSDish
- zlib under BSDish
- libc available as part of the operating system

Please see each library package for the exact details of their licenses.

3.2 Specification IPR

ZXID is based on open SAML and Liberty specifications. The parties that have developed these specifications, including Symlabs, have made Royalty Free (RF) licensing commitment. Please ask OASIS and Liberty Alliance for the specifics of their IPR policies and IPR disclosures.

Some protocols, such as WS-Trust and WS-Federation enjoy Microsoft's pledge¹ that they will not sue you even if you implement these specifications. You should evaluate yourself whether this is good enough for your situation.

3.3 Further Warranties

If you need the author or Symlabs to further disclaim IPR interest or make warranties of non-infringement, such declarations are available for a fee. Please contact sales@symlabs.com

Legal queries and clarifications will be answered at then-current Symlabs Professional Services rate, please contact sales@symlabs.com.

4 Testing

ZXID test suite is still in tatters. Some things that should be tested

1. Will generated HTTP redirect sig validate at IdP?
2. Does IdP issued A7N validate?
3. Validation of EncryptedAssertion?
4. Will generated SOAP binding sig validate at IdP?
5. Does IdP issued SOAP sig validate?

Metadata related

1. IBM metadata (can we parse)
2. Sun metadata (can we parse)

¹If you have a reference to where this pledge can be found, please let me know so it can be included here.

XML related

1. Fully qualified XML parses?
2. Unknown ns prefix that refers to known namespace URI
3. Known ns prefix, referring to wrong URI
4. Known prefix refers to aliased URI
5. Use of default namespaces working?
6. Unknown prefix and URI as long as it is never used
7. Unknown prefix and URI, used
8. Known NS (prefix or URI), unknown element

5 Integration of Other Libraries with ZXID

5.1 Conor Cahill's C++ Library for ID-WSF

Conor P. Cahill, of AOL and Intel fame, has developed and maintains a C++ library for ID-WSF 2.0 Web Service Client functionality for selected application protocols, including the ID-WSF 2.0 Discovery and some application protocols. Conor also provides a server side package that implements the corresponding WSP roles in Java. These libraries are valuable resources and come with extensive test suites - in fact, passing Conor's test suites has become the gold standard for validity and interoperability of any ID-WSF implementations (this is not to detract from formal IOP events and the Liberty certification program, but passing Conor's test suite is a good predictor of getting certified).

Install Recipe

Conor's libraries have certain dependencies. Following is my best understanding of how to get them installed.²

```
mkdir conor
cd conor
tar xvf /t/LibertyIDWSFServices-v0.8.2.tgz
cd ..
mkdir conor-cli
cd conor-cli/
tar xvf /t/LibertyClientToolkit-v1.0.1.tgz
```

²As of May 2007, Conor's packages explode in the current working directory. I recommend creating a wrapper directory first. Also, the client and server functionality can not be unpacked in same directory without creating conflict and overwriting some files.

5.2 Pat Patterson's php module

Pat Patterson of Sun distributes a pure PHP module (not to be confused with Sun's OpenSSO open source effort, with which Pat has some contact) that implements some aspects of SAML 2.0. As of May 2007, his library provides functionality that, by and large, parallels that of the `php_zxid` module. A major advantage of his module is that it does not have C shared library dependency, but beware that he still depends on XML parsing and popular crypto libraries (openssl) to be available. These assumptions are not onerous, but you should be aware of them in case your system differs from main stream deployments.

Overall, Pat's PHP implementation, as of May 2007, is still lacking in metadata generation and loading (it does not implement Auto-CoT or Well Known Location) and has some rough edges around less frequently used parts of the SAML specification. No doubt matters will improve over the time.

Pat's library handles only SSO and not ID Web Services. It would be possible to extract the discovery bootstrap from SSO using his library after which you can use ZXID WSC API to actually call the services.

5.3 Sun OpenSSO

Sun Microsystems distributes an open source implementation of SAML 2.0. Their implementation is of primary interest as it provides a freely available IdP implementation (as of May 2007 IMNSHO the ZXID SP interface is superior to the OpenSSO SP - and since both implement an open standard, you can mix ZXID SP with OpenSSO IdP).

Thus, the ZXID to OpenSSO integration reduces to each one acting in its role using standard wire protocol - SAML 2.0.

6 Appendix: Schema Grammars

Large parts of ZXID code are generated from *schema grammars* which are a convenient notation for describing XML schmata. This chapter gives a sampling of some schema grammars that are currently implemented and distributed in the ZXID package. For fuller list, see `sg` subdirectory of the distribution or `schemata.pd` file.

6.1 SAML 2.0

6.1.1 saml-schema-assertion-2.0 (sa)

```
# zxid/sg/saml-schema-assertion-2.0.sg
# $Id: saml-schema-assertion-2.0.sg,v 1.9 2009-08-25 16:22:45 sampo Exp $
#
# N.B. This file is not a direct conversion. Instead it has been manually edited to
# make it simpler and to facilitate code generation.
# 15.10.2006, extended AttributeValue schema to cater for bootstrap, Sampo Kellomaki (sampo@iki.fi)
```

Table 5: Schema grammar syntax

Construct	Description
ee	Bareword signifies an XML element
@aa	At (@) prefix signifies an XML attribute
%tt	Percent (%) prefix signifies a complexType
&gg	Ampersand (&) prefix signifies group
&@ag	Ampersand and at (&@) prefix signifies attributeGroup
xx -> %tt	Arrow (->) signifies reference to type that defines element or attribute
xx: ... ;	Colon (:) means that the definition of type follows immediately
ee	An element or attribute by itself means exactly one occurrence is expected
ee?	Question mark (?) means the element or attribute is optional
ee*	Asterisk (*) means the element may appear from zero to infinite number of times (same as * in regular expressions)
ee+	Plus (+) means the element must appear at least once, but may appear an infinite number of times (same as + in regular expressions)
ee{x,y}	The element must appear between x and y times (same as in regex)
ee ee	The pipe symbol () means elements are mutually exclusive choices.
ee ee	Concatenation of elements or attributes means sequence
<i>base(t)</i>	Introduce Extension base type (derive a type)
<i>redef(..)</i>	Redefine a type (using <xs:redefine> construct)
<i>mixed(l)</i>	Mark a complex type as having mixed content type, i.e. strings and elements alternate
<i>enum(...)</i>	Introduce enumeration of xs:strings
any	xs:any, the XML arbitrary element extension mechanism
@any	xs:anyAttribute, the XML arbitrary attribute extension mechanism
target(...)	Define target namespace described by the schema
import(...)	Bring in other schemata and namespaces
ns(...)	Declare existence of another namespace (without importing it)

```
# 10.2.2007, added other types of assertions as potential Advice content --S\
ampo
# 3.3.2007, added XACML support --Sampo
# 24.8.2009, modified sa:Statement to be able to carry xac:Response --Sampo
```

```
target(sa, urn:oasis:names:tc:SAML:2.0:assertion)
ns(xs,http://www.w3.org/2001/XMLSchema)
import(ds,http://www.w3.org/2000/09/xmldsig#,http://www.w3.org/TR/2002/REC-x\
mldsig-core-20020212/xmldsig-core-schema.xsd)
import(xenc,http://www.w3.org/2001/04/xmlenc#,http://www.w3.org/TR/2002/REC-\
xmlenc-core-20021210/xenc-schema.xsd)
ns(dil2, urn:liberty:disco:2003-08)
ns(a, http://www.w3.org/2005/08/addressing)
ns(sall, urn:oasis:names:tc:SAML:1.0:assertion)
ns(ffl2, urn:liberty:iff:2003-08)
ns(xasa, urn:oasis:xacml:2.0:saml:assertion:schema:os)
ns(xac, urn:oasis:names:tc:xacml:2.0:context:schema:os)
ns(xsi, http://www.w3.org/2001/XMLSchema-instance)
ns(idp, urn:liberty:idp:2006-12)
```

```

&@IDNameQualifiers:
  @NameQualifier? -> %xs:string
  @SPNameQualifier? -> %xs:string
  ;

BaseID -> %sa:BaseIDAbstractType
%BaseIDAbstractType:
  &@sa:IDNameQualifiers
  ;

NameID -> %sa:NameIDType
%NameIDType: base(xs:string)
  @Format? -> %xs:anyURI
  &@sa:IDNameQualifiers
  @SPProvidedID? -> %xs:string
  ;

%EncryptedElementType:
  xenc:EncryptedData
  xenc:EncryptedKey*
  ;

EncryptedID -> %sa:EncryptedElementType
Issuer -> %sa:NameIDType
AssertionIDRef -> %xs:NCName
AssertionURIRef -> %xs:anyURI

Assertion -> %sa:AssertionType
%AssertionType:
  sa:Issuer
  ds:Signature?
  sa:Subject?
  sa:Conditions?
  sa:Advice?
  sa:Statement* # *** how to express * for choice
  sa:AuthnStatement*
  sa:AuthzDecisionStatement*
  sa:AttributeStatement*
  xasa:XACMLAuthzDecisionStatement*
  xasa:XACMLPolicyStatement*
  @ID -> %xs:ID
  @IssueInstant -> %xs:dateTime
  @Version -> %xs:string
  ;

Subject -> %sa:SubjectType
%SubjectType:
  sa:BaseID? # Only one of the IDs should occur
  sa:NameID?
  sa:EncryptedID?
  sa:SubjectConfirmation* # This is more lax than SAML spec
  ;

SubjectConfirmation -> %sa:SubjectConfirmationType

```

```

%SubjectConfirmationType:
  sa:BaseID?                # Only one of the IDs should occur
  sa:NameID?
  sa:EncryptedID?
  sa:SubjectConfirmationData?
  @Method -> %xs:anyURI
  ;

SubjectConfirmationData -> %sa:SubjectConfirmationDataType
%SubjectConfirmationDataType: base(anyType)
  ds:KeyInfo+
  @Address? -> %xs:string
  @InResponseTo? -> %xs:NCName
  @NotBefore? -> %xs:dateTime
  @NotOnOrAfter? -> %xs:dateTime
  @Recipient? -> %xs:anyURI
  @xsi:type?
  @any
  ;

%KeyInfoConfirmationDataType: base(sa:SubjectConfirmationDataType)
  ds:KeyInfo+
  ;

Conditions -> %sa:ConditionsType
%ConditionsType:
  sa:Condition*             # *** Stated differently in XSD
  sa:AudienceRestriction*
  sa:OneTimeUse*
  sa:ProxyRestriction*
  idp:SubjectRestriction*
  @NotBefore? -> %xs:dateTime
  @NotOnOrAfter? -> %xs:dateTime
  ;

Condition -> %sa:ConditionAbstractType

AudienceRestriction -> %sa:AudienceRestrictionType
%AudienceRestrictionType: base(sa:ConditionAbstractType)
  sa:Audience+
  ;

Audience -> %xs:anyURI

OneTimeUse -> %sa:OneTimeUseType
%OneTimeUseType: base(sa:ConditionAbstractType) ;

ProxyRestriction -> %sa:ProxyRestrictionType
%ProxyRestrictionType: base(sa:ConditionAbstractType)
  sa:Audience*
  @Count? -> %xs:nonNegativeInteger
  ;

Advice -> %sa:AdviceType

```

```

%AdviceType:
  sa:AssertionIDRef*   # *** really a choice, but maxOccurs="unbounded"
  sa:AssertionURIRef*
  sa:Assertion*
  sa:EncryptedAssertion*
  sa11:Assertion*
  ff12:Assertion*
  any* ns(##other) processContents(lax)
;

EncryptedAssertion -> %sa:EncryptedElementType

#Statement -> %sa:StatementAbstractType

Statement -> %sa:StatementType

%StatementType: base(sa:StatementAbstractType)
  xac:Response*
  any* ns(##other) processContents(lax)
  @xsi:type? -> %xs:string
;

AuthnStatement -> %sa:AuthnStatementType
%AuthnStatementType: base(sa:StatementAbstractType)
  sa:SubjectLocality?
  sa:AuthnContext
  @AuthnInstant -> %xs:dateTime
  @SessionIndex? -> %xs:string
  @SessionNotOnOrAfter? -> %xs:dateTime
;

SubjectLocality -> %sa:SubjectLocalityType
%SubjectLocalityType:
  @Address? -> %xs:string
  @DNSName? -> %xs:string
;

AuthnContext -> %sa:AuthnContextType
%AuthnContextType:
  sa:AuthnContextClassRef? # N.B. We diverge from canonical XSD
  sa:AuthnContextDecl?
  sa:AuthnContextDeclRef?
  sa:AuthenticatingAuthority*
;

AuthnContextClassRef -> %xs:anyURI
AuthnContextDeclRef -> %xs:anyURI
AuthnContextDecl -> %xs:anyType
AuthenticatingAuthority -> %xs:anyURI

AuthzDecisionStatement -> %sa:AuthzDecisionStatementType
%AuthzDecisionStatementType: base(sa:StatementAbstractType)
  sa:Action+
  sa:Evidence?

```

```

    @Decision -> %sa:DecisionType
    @Resource -> %xs:anyURI
    ;

%DecisionType: enum( Permit Deny Indeterminate ) ;

Action -> %sa:ActionType
%ActionType: base(string)
    @Namespace -> %xs:anyURI
    ;

Evidence -> %sa:EvidenceType
%EvidenceType:
    sa:AssertionIDRef*      # XSD has choice maxOccurs="unbounded"
    sa:AssertionURIRef*
    sa:Assertion*
    sa:EncryptedAssertion*
    ;

AttributeStatement -> %sa:AttributeStatementType
%AttributeStatementType: base(sa:StatementAbstractType)
    sa:Attribute*          # XSD has choice maxOccurs="unbounded"
    sa:EncryptedAttribute*
    ;

Attribute -> %sa:AttributeType
%AttributeType:
    sa:AttributeValue*
    @FriendlyName? -> %xs:string
    @Name           -> %xs:string
    @NameFormat?   -> %xs:anyURI
    @any
    ;

# To cater for discovery bootstraps we add them to schema here
#AttributeValue -> %xs:anyType

AttributeValue -> %sa:AttributeValueType
%AttributeValueType:
    dil2:ResourceOffering*
    a:EndpointReference*
    ;

EncryptedAttribute -> %sa:EncryptedElementType

TestElem:
    sa:AttributeValue*
    ;

#EOF

```

6.1.2 saml-schema-protocol-2.0 (sp)

```
# zxid/sg/saml-schema-protocol-2.0.sg
# $Id: saml-schema-protocol-2.0.sg,v 1.5 2008-02-23 03:59:31 sampo Exp $
#
# N.B. This file is not a direct conversion. Instead it has been manually
# edited to make it simpler and to facilitate code generation.

target (sp,urn:oasis:names:tc:SAML:2.0:protocol)
import (sa,urn:oasis:names:tc:SAML:2.0:assertion,saml-schema-assertion-2.0.xsd\
d)
import (ds,http://www.w3.org/2000/09/xmldsig#,http://www.w3.org/TR/2002/REC-x\
mldsig-core-20020212/xmldsig-core-schema.xsd)
ns (xs, http://www.w3.org/2001/XMLSchema)

%RequestAbstractType:
  sa:Issuer?
  ds:Signature?
  sp:Extensions?
  @ID -> %xs:ID
  @Version -> %xs:string
  @IssueInstant -> %xs:dateTime
  @Destination? -> %xs:anyURI
  @Consent? -> %xs:anyURI
  ;

Extensions -> %sp:ExtensionsType
%ExtensionsType:
  any+
  ;

%StatusResponseType:
  sa:Issuer?
  ds:Signature?
  sp:Extensions?
  sp:Status
  @ID -> %xs:ID
  @InResponseTo? -> %xs:NCName
  @Version -> %xs:string
  @IssueInstant -> %xs:dateTime
  @Destination? -> %xs:anyURI
  @Consent? -> %xs:anyURI
  ;

Status -> %sp:StatusType
%StatusType:
  sp:StatusCode
  sp:StatusMessage?
  sp:StatusDetail?
  ;

StatusCode -> %sp:StatusCodeType
%StatusCodeType:
  sp:StatusCode?
```

```
@Value -> %xs:anyURI
;

StatusMessage -> %xs:string

StatusDetail -> %sp:StatusDetailType
%StatusDetailType:
  any*
;

AssertionIDRequest -> %sp:AssertionIDRequestType
%AssertionIDRequestType: base(sp:RequestAbstractType)
  sa:AssertionIDRef+
;

SubjectQuery -> %sp:SubjectQueryAbstractType
%SubjectQueryAbstractType: base(sp:RequestAbstractType)
  sa:Subject
;

AuthnQuery -> %sp:AuthnQueryType
%AuthnQueryType: base(sp:SubjectQueryAbstractType)
  sp:RequestedAuthnContext?
  @SessionIndex? -> %xs:string
;

RequestedAuthnContext -> %sp:RequestedAuthnContextType
%RequestedAuthnContextType:
  sa:AuthnContextClassRef*
  sa:AuthnContextDeclRef*
  @Comparison? -> %sp:AuthnContextComparisonType
;

%AuthnContextComparisonType: enum( exact minimum maximum better ) ;

AttributeQuery -> %sp:AttributeQueryType
%AttributeQueryType: base(sp:SubjectQueryAbstractType)
  sa:Attribute*
;

AuthzDecisionQuery -> %sp:AuthzDecisionQueryType
%AuthzDecisionQueryType: base(sp:SubjectQueryAbstractType)
  sa:Action+
  sa:Evidence?
  @Resource -> %xs:anyURI
;

AuthnRequest -> %sp:AuthnRequestType
%AuthnRequestType: base(sp:RequestAbstractType)
  sa:Subject?
  sp:NameIDPolicy?
  sa:Conditions?
  sp:RequestedAuthnContext?
  sp:Scoping?
```

```
@ForceAuthn? -> %xs:boolean
@IsPassive? -> %xs:boolean
@ProtocolBinding? -> %xs:anyURI
@AssertionConsumerServiceIndex? -> %xs:unsignedShort
@AssertionConsumerServiceURL? -> %xs:anyURI
@AttributeConsumingServiceIndex? -> %xs:unsignedShort
@ProviderName? -> %xs:string
;

NameIDPolicy -> %sp:NameIDPolicyType
%NameIDPolicyType:
  @Format? -> %xs:anyURI
  @SPNameQualifier? -> %xs:string
  @AllowCreate? -> %xs:boolean
;

Scoping -> %sp:ScopingType
%ScopingType:
  sp:IDPList?
  sp:RequesterID*
  @ProxyCount? -> %xs:nonNegativeInteger
;

RequesterID -> %xs:anyURI

IDPList -> %sp:IDPListType
%IDPListType:
  sp:IDPEntry+
  sp:GetComplete?
;

IDPEntry -> %sp:IDPEntryType
%IDPEntryType:
  @ProviderID -> %xs:anyURI
  @Name? -> %xs:string
  @Loc? -> %xs:anyURI
;

GetComplete -> %xs:anyURI

Response -> %sp:ResponseType
%ResponseType: base(sp:StatusResponseType)
  sa:Assertion?
  sa:EncryptedAssertion?
;

ArtifactResolve -> %sp:ArtifactResolveType
%ArtifactResolveType: base(sp:RequestAbstractType)
  sp:Artifact
;

Artifact -> %xs:string

ArtifactResponse -> %sp:ArtifactResponseType
```

```
%ArtifactResponseType: base(sp:StatusResponseType)
  sp:Response?
  any?
  ;

ManageNameIDRequest -> %sp:ManageNameIDRequestType
%ManageNameIDRequestType: base(sp:RequestAbstractType)
  sa:NameID?
  sa:EncryptedID?
  sp:NewID?
  sp:NewEncryptedID?
  sp:Terminate?
  ;

NewID -> %xs:string

NewEncryptedID -> %sa:EncryptedElementType

Terminate -> %sp:TerminateType

ManageNameIDResponse -> %sp:StatusResponseType

LogoutRequest -> %sp:LogoutRequestType
%LogoutRequestType: base(sp:RequestAbstractType)
  sa:BaseID?
  sa:NameID?
  sa:EncryptedID?
  sp:SessionIndex*
  @Reason? -> %xs:string
  @NotOnOrAfter? -> %xs:dateTime
  ;

SessionIndex -> %xs:string

LogoutResponse -> %sp:StatusResponseType

NameIDMappingRequest -> %sp:NameIDMappingRequestType
%NameIDMappingRequestType: base(sp:RequestAbstractType)
  sa:BaseID?
  sa:NameID?
  sa:EncryptedID?
  sp:NameIDPolicy
  ;

NameIDMappingResponse -> %sp:NameIDMappingResponseType
%NameIDMappingResponseType: base(sp:StatusResponseType)
  sa:NameID?
  sa:EncryptedID?
  ;

#EOF
```

6.1.3 saml-schema-metadata-2.0 (md)

```

# zxid/sg/saml-schema-metadata-2.0.sh .sg
# Slightly edited, 27.5.2006, Sampo Kellomaki (sampo@iki.fi)
# $Id: saml-schema-metadata-2.0.sg,v 1.3 2007-10-09 16:56:13 sampo Exp $

target(md,urn:oasis:names:tc:SAML:2.0:metadata)
import(ds,http://www.w3.org/2000/09/xmldsig#,http://www.w3.org/TR/2002/REC-x\
mldsig-core-20020212/xmldsig-core-schema.xsd)
import(xenc,http://www.w3.org/2001/04/xmlenc#,http://www.w3.org/TR/2002/REC-\
xmlenc-core-20021210/xenc-schema.xsd)
import(sa,urn:oasis:names:tc:SAML:2.0:assertion,saml-schema-assertion-2.0.xs\
d)
# import(xml,http://www.w3.org/XML/1998/namespace,http://www.w3.org/2001/xml\
.xsd)
ns(xs, http://www.w3.org/2001/XMLSchema)
ns(xml, http://www.w3.org/XML/1998/namespace)

%entityIDType: base(xs:anyURI) ;

%localizedNameType: base(xs:string)
  @xml:lang? -> %xs:string #@xml:lang vs. @lang ***
  #@lang? -> %xs:string
  ;

%localizedURIType: base(xs:anyURI)
  @xml:lang? -> %xs:string #@xml:lang vs. @lang ***
  #@lang? -> %xs:string
  ;

Extensions -> %md:ExtensionsType
%ExtensionsType:
  any+
  ;

%EndpointType:
  any*
  @Binding -> %xs:anyURI
  @Location -> %xs:anyURI
  @ResponseLocation? -> %xs:anyURI
  @index? -> %xs:unsignedShort
  @isDefault? -> %xs:boolean
  @any
  ;

EntitiesDescriptor -> %md:EntitiesDescriptorType
%EntitiesDescriptorType:
  ds:Signature?
  md:Extensions?
  md:EntityDescriptor* # these were originally choice unbounded
  md:EntitiesDescriptor*
  @validUntil? -> %dateTime
  @cacheDuration? -> %duration
  @ID? -> %xs:ID

```

```

    @Name? -> %xs:string
    ;

EntityDescriptor -> %md:EntityDescriptorType
%EntityDescriptorType:
    ds:Signature?
    md:Extensions?
    md:RoleDescriptor*           # following were originally choice unbo\
unded
    md:IDPSSODescriptor*
    md:SPSSODescriptor*
    md:AuthnAuthorityDescriptor*
    md:AttributeAuthorityDescriptor*
    md:PDPDescriptor*
    md:AffiliationDescriptor*
    md:Organization?
    md:ContactPerson*
    md:AdditionalMetadataLocation*
    @entityID -> %md:entityIDType
    @validUntil? -> %dateTime
    @cacheDuration? -> %duration
    @ID? -> %xs:ID
    @any
    ;

Organization -> %md:OrganizationType
%OrganizationType:
    md:Extensions?
    md:OrganizationName+
    md:OrganizationDisplayName+
    md:OrganizationURL+
    @any
    ;

OrganizationName -> %md:localizedNameType
OrganizationDisplayName -> %md:localizedNameType
OrganizationURL -> %md:localizedURIType

ContactPerson -> %md:ContactType
%ContactType:
    md:Extensions?
    md:Company?
    md:GivenName?
    md:SurName?
    md:EmailAddress*
    md:TelephoneNumber*
    @contactType -> %md:ContactTypeType
    @any
    ;

Company -> %xs:string
GivenName -> %xs:string
SurName -> %xs:string
EmailAddress -> %xs:anyURI

```

```

TelephoneNumber -> %xs:string

%ContactTypeType: enum( technical support administrative billing other ) ;

AdditionalMetadataLocation -> %md:AdditionalMetadataLocationType
%AdditionalMetadataLocationType: base(xs:anyURI)
  @namespace -> %xs:anyURI
  ;

RoleDescriptor -> %md:RoleDescriptorType
%RoleDescriptorType:
  ds:Signature?
  md:Extensions?
  md:KeyDescriptor*
  md:Organization?
  md:ContactPerson*
  @ID? -> %xs:ID
  @validUntil? -> %dateTime
  @cacheDuration? -> %duration
  @protocolSupportEnumeration -> %xs:anyURI
  @errorURL? -> %xs:anyURI
  @any
  ;

KeyDescriptor -> %md:KeyDescriptorType
%KeyDescriptorType:
  ds:KeyInfo
  md:EncryptionMethod*
  @use? -> %md:KeyTypes
  ;

%KeyTypes: enum( encryption signing ) ;
EncryptionMethod -> %xenc:EncryptionMethodType
%SSODescriptorType: base(md:RoleDescriptorType)
  md:ArtifactResolutionService*
  md:SingleLogoutService*
  md:ManageNameIDService*
  md:NameIDFormat*
  ;

ArtifactResolutionService -> %md:EndpointType
SingleLogoutService -> %md:EndpointType
ManageNameIDService -> %md:EndpointType
NameIDFormat -> %xs:anyURI

IDPSSODescriptor -> %md:IDPSSODescriptorType
%IDPSSODescriptorType: base(md:SSODescriptorType)
  md:SingleSignOnService+
  md:NameIDMappingService*
  md:AssertionIDRequestService*
  md:AttributeProfile*
  sa:Attribute*
  @WantAuthnRequestsSigned? -> %xs:boolean
  ;

```

```
SingleSignOnService -> %md:EndpointType
NameIDMappingService -> %md:EndpointType
AssertionIDRequestService -> %md:EndpointType
AttributeProfile -> %xs:anyURI

SPSSODescriptor -> %md:SPSSODescriptorType
%SPSSODescriptorType: base(md:SSODescriptorType)
  md:AssertionConsumerService+
  md:AttributeConsumingService*
  @AuthnRequestsSigned? -> %xs:boolean
  @WantAssertionsSigned? -> %xs:boolean
  ;

AssertionConsumerService -> %md:EndpointType

AttributeConsumingService -> %md:AttributeConsumingServiceType
%AttributeConsumingServiceType:
  md:ServiceName+
  md:ServiceDescription*
  md:RequestedAttribute+
  @index -> %xs:unsignedShort
  @isDefault? -> %xs:boolean
  ;

ServiceName -> %md:localizedNameType
ServiceDescription -> %md:localizedNameType

RequestedAttribute -> %md:RequestedAttributeType
%RequestedAttributeType: base(sa:AttributeType)
  @isRequired? -> %xs:boolean
  ;

AuthnAuthorityDescriptor -> %md:AuthnAuthorityDescriptorType
%AuthnAuthorityDescriptorType: base(md:RoleDescriptorType)
  md:AuthnQueryService+
  md:AssertionIDRequestService*
  md:NameIDFormat*
  ;

AuthnQueryService -> %md:EndpointType

PDPDescriptor -> %md:PDPDescriptorType
%PDPDescriptorType: base(md:RoleDescriptorType)
  md:AuthzService+
  md:AssertionIDRequestService*
  md:NameIDFormat*
  ;

AuthzService -> %md:EndpointType

AttributeAuthorityDescriptor -> %md:AttributeAuthorityDescriptorType
%AttributeAuthorityDescriptorType: base(md:RoleDescriptorType)
  md:AttributeService+
```

```

    md:AssertionIDRequestService*
    md:NameIDFormat*
    md:AttributeProfile*
    sa:Attribute*
    ;

AttributeService -> %md:EndpointType

AffiliationDescriptor -> %md:AffiliationDescriptorType
%AffiliationDescriptorType:
    ds:Signature?
    md:Extensions?
    md:AffiliateMember+
    md:KeyDescriptor*
    @affiliationOwnerID -> %md:entityIDType
    @validUntil? -> %dateTime
    @cacheDuration? -> %duration
    @ID? -> %xs:ID
    @any
    ;

AffiliateMember -> %md:entityIDType

#EOF

```

6.2 Liberty ID-WSF 2.0

6.2.1 liberty-idwsf-utility-v2.0 (lu)

```

# zxid/sg/liberty-idwsf-utility-v2.0.sg
# Slightly edited, 18.9.2006, Sampo Kellomäki (sampo@iki.fi)
# $Id: liberty-idwsf-utility-v2.0.sg,v 1.3 2009-09-05 02:23:41 sampo Exp $

target(lu, urn:liberty:util:2006-08)

%IDType:    base(xs:string) ;
%IDReferenceType: base(xs:string) ;
@itemID    -> %lu:IDType
@itemIDRef -> %lu:IDReferenceType

%StatusType:
    lu:Status*
    @code    -> %xs:string
    @ref?    -> %lu:IDReferenceType
    @comment? -> %xs:string
    ;
Status     -> %lu:StatusType

%ResponseType:
    lu:Status
    lu:Extension*
    @itemIDRef? -> %lu:IDReferenceType

```

```

    @any
    ;

    TestResult      -> %lu:TestResultType
    %TestResultType: base(xs:boolean)
    @itemIDRef     -> %lu:IDReferenceType
    ;

    %EmptyType:    base(xs:anyType) ;

    Extension -> %lu:extensionType
    %extensionType:
    any+ ns(##other) processContents(lax)
    ;

    #EOF

```

6.2.2 liberty-idwsf-soap-binding-v2.0 (b)

```

# zxid/sg/liberty-idwsf-soap-binding-v2.0.sg
# Slightly edited, 5.9.2006, Sampo Kellomäki (sampo@iki.fi)
# $Id: liberty-idwsf-soap-binding-v2.0.sg,v 1.7 2007-09-30 05:10:03 sampo Ex\
p $

target(b,      urn:liberty:sb:2006-08)
import(sp,     urn:oasis:names:tc:SAML:2.0:protocol)
import(wsu,    http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecu\
ity-utility-1.0.xsd,wss-util-1.0.xsd)
import(a,      http://www.w3.org/2005/08/addressing,ws-addr-1.0.xsd)
import(lu,     urn:liberty:util:2006-08,liberty-idwsf-utility-v2.0.xsd)
import(e,      http://schemas.xmlsoap.org/soap/envelope/)
import(sa11,   urn:oasis:names:tc:SAML:1.0:assertion)
import(sa,     urn:oasis:names:tc:SAML:2.0:assertion)
import(ff12,   urn:liberty:iff:2003-08)

&@hdr:
  @wsu:Id?
  @e:mustUnderstand?
  @e:actor?
  @id? -> %xs:anyURI
  ;

Framework -> %b:FrameworkType
%FrameworkType:
  any* processContents(lax)
  @version -> %xs:string
  &@b:hdr      # Added by Sampo
  @any
  ;

Sender -> %b:SenderType
%SenderType:

```

```

    @providerID      -> %xs:anyURI
    @affiliationID?  -> %xs:anyURI
    &@b:hdr          # Added by Sampo
    @any
    ;

TargetIdentity -> %b:TargetIdentityType
%TargetIdentityType:
    sa:Assertion?
    sa11:Assertion?
    ff12:Assertion?
    any* processContents(lax)
    &@b:hdr          # Added by Sampo
    @any
    ;

CredentialsContext -> %b:CredentialsContextType
%CredentialsContextType:
    sp:RequestedAuthnContext?
    b:SecurityMechID* -> %xs:anyURI
    &@b:hdr          # Added by Sampo
    @any
    ;

EndpointUpdate -> %b:EndpointUpdateType
%EndpointUpdateType: base(a:EndpointReferenceType)
    @updateType? -> %xs:anyURI
    ;

Timeout -> %b:TimeoutType
%TimeoutType:
    @maxProcessingTime -> %xs:integer
    &@b:hdr          # Added by Sampo
    @any
    ;

ProcessingContext -> %b:ProcessingContextType
%ProcessingContextType: base(xs:anyURI)
    &@b:hdr          # Added by Sampo
    @any
    ;

Consent -> %b:ConsentType
%ConsentType:
    @uri -> %xs:anyURI
    @timestamp? -> %xs:dateTime
    &@b:hdr          # Added by Sampo
    @any
    ;

UsageDirective -> %b:UsageDirectiveType
%UsageDirectiveType:
    any+ ns(##other) processContents(lax)
    @ref -> %xs:IDREF

```

```

    &@b:hdr          # Added by Sampo
    @any
    ;

ApplicationEPR -> %a:EndpointReferenceType

UserInteraction -> %b:UserInteractionHeaderType
%UserInteractionHeaderType:
    b:InteractionService* -> %a:EndpointReferenceType
    @interact? -> %xs:string default (interactIfNeeded)
    @language? -> %xs:NMTOKENS
    @redirect? -> %xs:boolean default (0)
    @maxInteractTime? -> %xs:integer
    &@b:hdr          # Added by Sampo
    @any
    ;

RedirectRequest -> %b:RedirectRequestType
%RedirectRequestType:
    @redirectURL -> %xs:anyURI
    &@b:hdr          # Added by Sampo
    ;

#EOF

```

6.2.3 liberty-idwsf-security-mechanisms-v2.0 (sec)

```

# zxid/sg/liberty-idwsf-security-mechanisms-v2.0.sg
# Slightly edited, 5.9.2006, Sampo Kellomäki (sampo@iki.fi)
# 10.2.2007, added sa:Assertion as potential security token type --Sampo
# $Id: liberty-idwsf-security-mechanisms-v2.0.sg,v 1.7 2009-08-25 16:22:45 s\
# ampo Exp $

target(sec, urn:liberty:security:2006-08)
ns(sa, urn:oasis:names:tc:SAML:2.0:assertion)
ns(sall, urn:oasis:names:tc:SAML:1.0:assertion)
ns(ff12, urn:liberty:iff:2003-08)

TokenPolicy -> %sec:TokenPolicyType
%TokenPolicyType:
    any* processContents(lax)
    @validUntil? -> %xs:dateTime
    @issueTo? -> %xs:anyURI
    @type? -> %xs:anyURI
    @wantDSEPR? -> %xs:boolean
    ;

# @any*

TransitedProvider -> %sec:TransitedProviderType
%TransitedProviderType: base(xs:anyURI)
    @timeStamp? -> %xs:dateTime

```

```

    @confirmationURI? -> %xs:anyURI
    ;

TransitedProviderPath -> %sec:TransitedProviderPathType
%TransitedProviderPathType:
    sec:TransitedProvider+
    ;

Token -> %sec:TokenType
%TokenType:
    sa:Assertion?
    sa:EncryptedAssertion?
    saml:Assertion?
    ffl2:Assertion?
    any* processContents(lax)
    @id? -> %xs:ID
    @ref? -> %xs:anyURI
    @usage? -> %xs:anyURI
    ;

#EOF

```

6.2.4 liberty-idwsf-disco-svc-v2.0 (di)

```

# zxid/sg/liberty-idwsf-disco-svc-v2.0.sg
# Slightly edited, 18.9.2006, Sampo Kellomäki (sampo@iki.fi)
# $Id: liberty-idwsf-disco-svc-v2.0.sg,v 1.2 2009-09-05 02:23:41 sampo Exp $

target(di, urn:liberty:disco:2006-08)
import(md, urn:oasis:names:tc:SAML:2.0:metadata, saml-schema-metadata-2.0.x\
sd)
import(b, urn:liberty:sb:2006-08, liberty-idwsf-soap-binding-v2.0.xsd)
import(sbf, urn:liberty:sb, liberty-idwsf-soap-binding.xsd)
import(a, http://www.w3.org/2005/08/addressing, ws-addr-1.0.xsd)
import(lu, urn:liberty:util:2006-08, liberty-idwsf-utility-v2.0.xsd)
import(sec, urn:liberty:security:2006-08, liberty-idwsf-security-mechanisms-\
v2.0.xsd)

Abstract -> %xs:string
ProviderID -> %xs:anyURI
ServiceType -> %xs:anyURI
Framework -> %sbf:FrameworkType
@NotOnOrAfter -> %xs:dateTime

SecurityContext:
    di:SecurityMechID+
    sec:Token*
    ;
SecurityMechID -> %xs:anyURI

Options -> %di:OptionsType
Option -> %xs:anyURI

```

```

%OptionsType:
  di:Option*
  ;

Address -> %xs:anyURI
Action -> %xs:anyURI

Keys -> %di:KeyType
%KeyType:
  md:KeyDescriptor+
  ;

SvcMD -> %di:SvcMetadataType
%SvcMetadataType:
  di:Abstract
  di:ProviderID
  di:ServiceContext+
  @svcMDID? -> %xs:string
  ;

ServiceContext -> %di:ServiceContextType
%ServiceContextType:
  di:ServiceType+
  di:Options*
  di:EndpointContext+
  ;

EndpointContext -> %di:EndpointContextType
%EndpointContextType:
  di:Address+
  sbf:Framework+
  di:SecurityMechID+
  di:Action*
  ;

SvcMDID -> %xs:string

Query -> %di:QueryType
%QueryType:
  di:RequestedService* -> %di:RequestedServiceType
  @any
  ;

%RequestedServiceType:
  di:ServiceType*
  di:ProviderID*
  di:Options*
  di:SecurityMechID*
  di:Framework*
  di:Action*
  any* ns(##other) processContents(lax)
  @reqID? -> %xs:string
  @resultsType? -> %xs:string
  ;

```

```
QueryResponse -> %di:QueryResponseType
%QueryResponseType:
  lu:Status
  a:EndpointReference*
  @any
  ;

SvcMDAssociationAdd -> %di:SvcMDAssociationAddType
%SvcMDAssociationAddType:
  di:SvcMDID+
  @any
  ;

SvcMDAssociationAddResponse -> %di:SvcMDAssociationAddResponseType
%SvcMDAssociationAddResponseType:
  lu:Status
  @any
  ;

SvcMDAssociationDelete -> %di:SvcMDAssociationDeleteType
%SvcMDAssociationDeleteType:
  di:SvcMDID+
  @any
  ;

SvcMDAssociationDeleteResponse -> %di:SvcMDAssociationDeleteResponseType
%SvcMDAssociationDeleteResponseType:
  lu:Status
  @any
  ;

SvcMDAssociationQuery -> %di:SvcMDAssociationQueryType
%SvcMDAssociationQueryType:
  di:SvcMDID*
  @any
  ;

SvcMDAssociationQueryResponse -> %di:SvcMDAssociationQueryResponseType
%SvcMDAssociationQueryResponseType:
  lu:Status
  di:SvcMDID*
  @any
  ;

SvcMDRegister -> %di:SvcMDRegisterType
%SvcMDRegisterType:
  di:SvcMD+
  @any
  ;

SvcMDRegisterResponse -> %di:SvcMDRegisterResponseType
%SvcMDRegisterResponseType:
  lu:Status
```

```

    di:SvcMDID*
    di:Keys*
    @any
    ;

SvcMDDelete -> %di:SvcMDDeleteType
%SvcMDDeleteType:
    di:SvcMDID+
    @any
    ;

SvcMDDeleteResponse -> %di:SvcMDDeleteResponseType
%SvcMDDeleteResponseType:
    lu:Status
    @any
    ;

SvcMDQuery -> %di:SvcMDQueryType
%SvcMDQueryType:
    di:SvcMDID*
    @any
    ;

SvcMDQueryResponse -> %di:SvcMDQueryResponseType
%SvcMDQueryResponseType:
    lu:Status
    di:SvcMD*
    @any
    ;

SvcMDReplace -> %di:SvcMDReplaceType
%SvcMDReplaceType:
    di:SvcMD+
    @any
    ;

SvcMDReplaceResponse -> %di:SvcMDReplaceResponseType
%SvcMDReplaceResponseType:
    lu:Status
    @any
    ;

#EOF

```

6.2.5 id-dap (dap)

```

# id-dap.sg -- Authorative ID-DAP 1.0 Service Schema
# Author: Sampo Kellomaki (sampo@symlabs.com)
# http://www.w3.org/2001/03/webdata/xsv
# $Id: id-dap.sg,v 1.2 2007-06-19 15:17:04 sampo Exp $
# This schema reflects Liberty ID Directory Access Protocol,
# version 1.0-07 of 11.10.2006

```

```

target (dap,      urn:liberty:id-sis-dap:2006-08:dst-2.1)
import (dst,     urn:liberty:dst:2006-08,      liberty-idwsf-dst-v2.1.xsd)
import (subs,   urn:liberty:ssos:2006-08,      liberty-idwsf-subsv1.0.xsd)
import (lu,     urn:liberty:util:2006-08,    liberty-idwsf-utility-v2.0.xsd)

Create          -> %dap:CreateType
CreateResponse -> %dap:CreateResponseType
Query           -> %dap:QueryType
QueryResponse  -> %dap:QueryResponseType
Modify          -> %dap:ModifyType
ModifyResponse -> %dap:ModifyResponseType
Delete          -> %dap>DeleteType
DeleteResponse -> %dap>DeleteResponseType
Notify         -> %dap:NotifyType
NotifyResponse -> %dap:NotifyResponseType

%SelectType:
  dap:dn?          -> %xs:string
  dap:filter?     -> %xs:string
  @scope?         -> %xs:integer  default (0)
  @sizelimit?    -> %xs:integer  default (0)
  @timelimit?    -> %xs:integer  default (0)
  @attributes?   -> %xs:string
  @typesonly?    -> %xs:boolean  default (false)
  @dereferences? -> %xs:integer  default (0)
  ;

%TestOpType:      base (dap:SelectType) ;
%SortType:        base (xs:string) ;
%TriggerType:     base (xs:string) ;
%AggregationType: base (xs:string) ;

%AppDataType:
  dap:LDIF?
  dap:Subscription?
  ;

LDIF: base (xs:string)
  &@dst:localizedLeafAttributes
  ;

%CreateType:      base (dst:RequestType)
  dap:Subscription*
  dap:CreateItem+
  dap:ResultQuery*
  ;

CreateItem        -> %dap:CreateItemType
%CreateItemType:
  dap:NewData?
  &@dst:CreateItemAttributeGroup
  ;

```

```

NewData                -> %dap:AppDataType

%CreateResponseType:  base (dap:DataResponseType) ;
%DataResponseType:   base (dst:DataResponseBaseType)
  dap:ItemData*
  ;

%QueryType:          base (dst:RequestType)
  dap:TestItem*
  dap:QueryItem*
  dap:Subscription*
  ;

TestItem              -> %dap:TestItemType
%TestItemType:       base (dst:TestItemBaseType)
  dap:TestOp?       -> %dap:TestOpType
  ;

QueryItem            -> %dap:QueryItemType
%QueryItemType:     base (dap:ResultQueryType)
  &@dst:PaginationAttributeGroup
  ;

%QueryResponseType:  base (dst:DataResponseBaseType)
  dst:TestResult*
  dap:Data*
  ;

Data                 -> %dap:DataType
%DataType:           base (dap:ItemDataType)
  &@dst:PaginationResponseAttributeGroup
  ;

%ModifyType:         base (dst:RequestType)
  dap:Subscription*
  dap:ModifyItem+
  dap:ResultQuery*
  ;

ModifyItem           -> %dap:ModifyItemType
%ModifyItemType:
  dap:Select?
  dap:NewData?
  &@dst:ModifyItemAttributeGroup
  ;

%ModifyResponseType: base (dap:DataResponseType) ;

%DeleteType:         base (dst:RequestType)
  dap:DeleteItem+
  ;

DeleteItem           -> %dap>DeleteItemType
%DeleteItemType:     base (dst>DeleteItemBaseType)

```

```

    dap:Select?
    ;

%DeleteResponseType: base(lu:ResponseType) ;

Select                -> %dap:SelectType

ResultQuery           -> %dap:ResultQueryType
%ResultQueryType:    base(dst:ResultQueryBaseType)
    dap:Select?
    dap:Sort?         -> %dap:SortType
    ;

ItemData              -> %dap:ItemDataType
%ItemDataType:       base(dap:AppDataType)
    &@dst:ItemDataAttributeGroup
    ;

Subscription          -> %dap:SubscriptionType
%SubscriptionType:   base(subs:SubscriptionType)
    dap:ResultQuery*
    dap:Aggregation? -> %dap:AggregationType
    dap:Trigger?     -> %dap:TriggerType
    ;

%NotifyType:         base(dst:RequestType)
    dap:Notification*
    &@subs:NotifyAttributeGroup
    ;

Notification          -> %dap:NotificationType
%NotificationType:   base(subs:NotificationType)
    dap:ItemData*
    ;

%NotifyResponseType: base(subs:NotifyResponseType) ;

#EOF

```

6.2.6 liberty-idwsf-subsv1.0 (subs)

```

# zxid/sg/liberty-idwsf-subsv1.0.sg
# Slightly edited, 1.3.2007, Sampo Kellomäki (sampo@iki.fi)
# $Id: liberty-idwsf-subsv1.0.sg,v 1.2 2009-09-05 02:23:41 sampo Exp $

target(subs, urn:liberty:ssos:2006-08)
import(lu, urn:liberty:util:2006-08, liberty-idwsf-utility-v2.0.xsd)

%SubscriptionType:
    subs:RefItem*
    lu:Extension*
    @subscriptionID -> %lu:IDType

```

```

    @notifyToRef      -> %xs:anyURI
    @adminNotifyToRef? -> %xs:anyURI
    @starts?         -> %xs:dateTime
    @expires?        -> %xs:dateTime
    @id?             -> %xs:ID
    @includeData?:   enum( Yes No YesWithCommonAttributes ) ;
;

RefItem -> %subs:RefItemType
%RefItemType:
    @subscriptionID? -> %lu:IDType
    @lu:itemIDRef
;

%NotifyAttributeGroup:
    @timeStamp?      -> %xs:dateTime
;

%NotificationType:
    lu:TestResult*
    @id?             -> %xs:ID
    @subscriptionID -> %lu:IDType
    @expires?        -> %xs:dateTime
    @endReason?      -> %xs:anyURI
;

%NotifyResponseType: base(lu:ResponseType) ;

#EOF

```

6.2.7 liberty-idwsf-dst-v2.1 (dst)

```

# zxid/sg/liberty-idwsf-dst-v2.1.sg
# Slightly edited, 1.3.2007, Sampo Kellomaki (sampo@iki.fi)
# $Id: liberty-idwsf-dst-v2.1.sg,v 1.2 2009-09-05 02:23:41 sampo Exp $

target(dst, urn:liberty:dst:2006-08)
import(lu, urn:liberty:util:2006-08, liberty-idwsf-utility-v2.0.xsd)
import(xml, http://www.w3.org/XML/1998/namespace, http://www.w3.org/2001/xml\
.xsd)

@id -> %lu:IDType
@modificationTime -> %xs:dateTime
%commonAttributes:
    @dst:id?
    @dst:modificationTime?
;

@ACC -> %xs:anyURI
@ACCTime -> %xs:dateTime
@modifier -> %xs:string

%leafAttributes:

```

```
&@dst:commonAttributes
@dst:ACC?
@dst:ACCTime?
@dst:modifier?
;

@script -> %xs:anyURI

&@localizedLeafAttributes:
  &@dst:leafAttributes
  @xml:lang
  @dst:script?
;

@refreshOnOrAfter -> %xs:dateTime
@destroyOnOrAfter -> %xs:dateTime

%DSTLocalizedString: base(xs:string)
  &@dst:localizedLeafAttributes
;

%DSTString: base(xs:string)
  &@dst:leafAttributes
;

%DSTInteger: base(xs:integer)
  &@dst:leafAttributes
;

%DSTURI: base(xs:anyURI)
  &@dst:leafAttributes
;

%DSTDate: base(xs:date)
  &@dst:leafAttributes
;

%DSTMonthDay: base(xs:gMonthDay)
  &@dst:leafAttributes
;

@itemID -> %lu:IDType
@itemIDRef -> %lu:IDReferenceType

%RequestType:
  lu:Extension*
  @dst:itemID?
  @any
;

%ResponseType:
  lu:Status
  lu:Extension*
  @dst:itemIDRef?
```

```

    @any
    ;

%DataResponseBaseType: base(dst:ResponseType)
    @timeStamp? -> %xs:dateTime
    ;

ChangeFormat: enum( ChangedElements CurrentElements ) ;
@changeFormat: enum( ChangedElements CurrentElements All ) ;
@objectType -> %xs:NCName
@predefined -> %xs:string

&@selectQualif:
    @dst:objectType?
    @dst:predefined?
    ;

%ResultQueryBaseType:
    dst:ChangeFormat{0,2}
    &@dst:selectQualif
    @dst:itemIDRef?
    @contingency? -> %xs:boolean
    @includeCommonAttributes? -> %xs:boolean default (0)
    @changedSince? -> %xs:dateTime
    @dst:itemID?
    ;

&@ItemDataAttributeGroup:
    @dst:itemIDRef?
    @notSorted?: enum( Now Never ) ;
    @dst:changeFormat?
    ;

%TestItemBaseType:
    &@dst:selectQualif
    @id? -> %xs:ID
    @dst:itemID?
    ;

TestResult -> %dst:TestResultType
%TestResultType: base(xs:boolean)
    @dst:itemIDRef
    ;

&@PaginationAttributeGroup:
    @count? -> %xs:nonNegativeInteger
    @offset? -> %xs:nonNegativeInteger default (0)
    @setID? -> %lu:IDType
    @setReq?: enum( Static DeleteSet ) ;
    ;

&@PaginationResponseAttributeGroup:
    @remaining? -> %xs:integer
    @nextOffset? -> %xs:nonNegativeInteger default (0)

```

```

    @setID? -> %lu:IDType
    ;

    &@CreateItemAttributeGroup:
        @dst:objectType?
        @id? -> %xs:ID
        @dst:itemID?
    ;

    &@ModifyItemAttributeGroup:
        &@dst:selectQualif
        @notChangedSince? -> %xs:dateTime
        @overrideAllowed? -> %xs:boolean default (0)
        @id? -> %xs:ID
        @dst:itemID?
    ;

    %DeleteItemBaseType:
        &@dst:selectQualif
        @notChangedSince? -> %xs:dateTime
        @id? -> %xs:ID
        @dst:itemID?
    ;
    %DeleteResponseType: base(dst:ResponseType) ;

#EOF

```

6.3 SOAP 1.1 Processor wsf-soap11 (e)

```

# zxid/sg/wsf-soap11.sg
# $Id: wsf-soap11.sg,v 1.12 2009-08-25 16:22:45 sampo Exp $
# Heavily edited, 27.5.2006, Sampo Kellomäki (sampo@iki.fi)
# 26.2.2007, merged saml20-soap11.sg and di-soap11.sg to only
#     one SOAP processor. --Sampo
# 3.3.2007, added XACML support --Sampo
#
# Mega SOAP processor for Web Services and SSO Frameworks
#
# Main purpose of this schema is to permit direct, one pass, parsing of
# of SAML and WSF content in SOAP envelope. Thus relevant SOAP extension
# points have been replaced with actual SAML and WSF elements.
#
# When you add new SOAP messages, you need to add them here, to the body.
# See also zxid/c/zx-e-data.h, which is generated.

target(e, http://schemas.xmlsoap.org/soap/envelope/)
ns(xs, http://www.w3.org/2001/XMLSchema)
ns(a, http://www.w3.org/2005/08/addressing)
ns(sbf, urn:liberty:sb)
ns(b, urn:liberty:sb:2006-08)
ns(b12, urn:liberty:sb:2003-08)
ns(di, urn:liberty:disco:2006-08)

```

6.3 SOAP 1.1 Processor *wsf-soap11* (e) 6 APPENDIX: SCHEMA GRAMMARS

```
ns(dil2, urn:liberty:disco:2003-08)
ns(dap, urn:liberty:id-sis-dap:2006-08:dst-2.1)
ns(ps, urn:liberty:ps:2006-08)
ns(im, urn:liberty:ims:2006-08)
ns(as, urn:liberty:sa:2006-08)
ns(wsse, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity\
-secext-1.0.xsd)
ns(xasp, urn:oasis:xacml:2.0:saml:protocol:schema:os)
ns(mm7, http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-\
6-MM7-1-4)
ns(cb, urn:liberty:id-sis-cb:2004-10)
ns(gl, urn:liberty:id-sis-gl:2005-07)
ns(dp, urn:liberty:dp:2006-12)
ns(pmm, urn:liberty:pmm:2006-12)
ns(prov, urn:liberty:prov:2006-12)
ns(shps, urn:liberty:shps:2006-12)
ns(idp, urn:liberty:idp:2006-12)
ns(idhrxml, urn:id-sis-idhrxml:2007-06:dst-2.1)
ns(demomed, urn:x-demo:me:2006-01)
```

Envelope -> %e:Envelope

```
%Envelope:
  e:Header?
  e:Body
  @id? -> %xs:ID
  any*
  @any?
  ;
```

Header -> %e:Header

```
%Header:
  paos:Request?
  paos:Response?
  ecp:Request?
  ecp:Response?
  ecp:RelayState?
  a:MessageID?
  a:RelatesTo?
  a:ReplyTo?
  a:From?
  a:FaultTo?
  a:To?
  a:Action?
  a:ReferenceParameters?
  sbf:Framework?
  b:Framework?
  b:Sender?
  b:TargetIdentity?
  b:CredentialsContext?
  b:EndpointUpdate?
  b:Timeout?
  b:ProcessingContext?
  b:Consent?
  b:UsageDirective?
```

```

b:ApplicationEPR?
b:UserInteraction?
b:RedirectRequest?
b12:Correlation?
b12:Provider?
b12:ProcessingContext?
b12:Consent?
b12:UsageDirective?
mm7:TransactionID?
wsse:Security?
@id? -> %xs:ID
any*
@any?
;

```

Body -> %e:Body

```

%Body:
  sp:ArtifactResolve?
  sp:ArtifactResponse?
  sp:ManageNameIDRequest?
  sp:ManageNameIDResponse?
  sp:LogoutRequest?
  sp:LogoutResponse?
  sp:NameIDMappingRequest?
  sp:NameIDMappingResponse?
  sp:AttributeQuery?
  sp:AuthnQuery?
  sp:AuthzDecisionQuery?
  sp:AssertionIDRequest?
  sp:Response?
  sp:AuthnRequest?
  sp11:Request?
  sp11:Response?
  ff12:RegisterNameIdentifierRequest?
  ff12:RegisterNameIdentifierResponse?
  ff12:FederationTerminationNotification?
  ff12:LogoutRequest?
  ff12:LogoutResponse?
  ff12:NameIdentifierMappingRequest?
  ff12:NameIdentifierMappingResponse?
  xasp:XACMLAuthzDecisionQuery?
  xasp:XACMLPolicyQuery?
  di:Query?
  di:QueryResponse?
  di12:Query?
  di12:QueryResponse?
  di12:Modify?
  di12:ModifyResponse?
  e:Fault?
  di:SvcMDAssociationAdd?
  di:SvcMDAssociationAddResponse?
  di:SvcMDAssociationDelete?
  di:SvcMDAssociationDeleteResponse?
  di:SvcMDAssociationQuery?

```

di:SvcMDAssociationQueryResponse?
di:SvcMDRegister?
di:SvcMDRegisterResponse?
di:SvcMDDelete?
di:SvcMDDeleteResponse?
di:SvcMDQuery?
di:SvcMDQueryResponse?
di:SvcMDReplace?
di:SvcMDReplaceResponse?
dap:Create?
dap:CreateResponse?
dap:Query?
dap:QueryResponse?
dap:Modify?
dap:ModifyResponse?
dap:Delete?
dap:DeleteResponse?
dap:Notify?
dap:NotifyResponse?
ps:AddEntityRequest?
ps:AddEntityResponse?
ps:AddKnownEntityRequest?
ps:AddKnownEntityResponse?
ps:AddCollectionRequest?
ps:AddCollectionResponse?
ps:AddToCollectionRequest?
ps:AddToCollectionResponse?
ps:RemoveEntityRequest?
ps:RemoveEntityResponse?
ps:RemoveCollectionRequest?
ps:RemoveCollectionResponse?
ps:RemoveFromCollectionRequest?
ps:RemoveFromCollectionResponse?
ps:ListMembersRequest?
ps:ListMembersResponse?
ps:QueryObjectsRequest?
ps:QueryObjectsResponse?
ps:GetObjectInfoRequest?
ps:GetObjectInfoResponse?
ps:SetObjectInfoRequest?
ps:SetObjectInfoResponse?
ps:TestMembershipRequest?
ps:TestMembershipResponse?
ps:ResolveIdentifierRequest?
ps:ResolveIdentifierResponse?
ps:Notify?
ps:NotifyResponse?
im:IdentityMappingRequest?
im:IdentityMappingResponse?
as:SASLRequest?
as:SASLResponse?
mm7:SubmitReq?
mm7:SubmitRsp?
mm7:DeliverReq?

mm7:DeliverRsp?
mm7:CancelReq?
mm7:CancelRsp?
mm7:ReplaceReq?
mm7:ReplaceRsp?
mm7:extendedCancelReq?
mm7:extendedCancelRsp?
mm7:extendedReplaceReq?
mm7:extendedReplaceRsp?
mm7:DeliveryReportReq?
mm7:DeliveryReportRsp?
mm7:ReadReplyReq?
mm7:ReadReplyRsp?
mm7:RSErrorRsp?
mm7:VASPErrrorRsp?
mm7:QueryStatusReq?
mm7:QueryStatusRsp?
cb:Query?
cb:QueryResponse?
cb:Create?
cb:CreateResponse?
cb>Delete?
cb>DeleteResponse?
cb:Modify?
cb:ModifyResponse?
cb:Notify?
cb:NotifyResponse?
cb:ReportUsage?
cb:ReportUsageResponse?
gl:Query?
gl:QueryResponse?
gl:Create?
gl:CreateResponse?
gl>Delete?
gl>DeleteResponse?
gl:Modify?
gl:ModifyResponse?
gl:Notify?
gl:NotifyResponse?
demomed:StoreObjectRequest?
demomed:StoreObjectResponse?
demomed:GetObjectListRequest?
demomed:GetObjectListResponse?
demomed:GetObjectRequest?
demomed:GetObjectResponse?
demomed>DeleteObjectRequest?
demomed>DeleteObjectResponse?
pmm:Provision?
pmm:ProvisionResponse?
pmm:PMActivate?
pmm:PMActivateResponse?
pmm:PMDeactivate?
pmm:PMDeactivateResponse?
pmm:PMDelete?

pmm:PMDeleteResponse?
pmm:PMUpdate?
pmm:PMUpdateResponse?
pmm:PMGetStatus?
pmm:PMGetStatusResponse?
pmm:PMSetStatus?
pmm:PMSetStatusResponse?
prov:PMERegister?
prov:PMERegisterResponse?
prov:PMEUpload?
prov:PMEUploadResponse?
prov:PMEDownload?
prov:PMEDownloadResponse?
prov:PMEEEnable?
prov:PMEEEnableResponse?
prov:PMEDisable?
prov:PMEDisableResponse?
prov:PMEDelete?
prov:PMEDeleteResponse?
prov:PMEGetInfo?
prov:PMEGetInfoResponse?
prov:PMGetStatus?
prov:PMGetStatusResponse?
prov:PMSetStatus?
prov:PMSetStatusResponse?
prov:PMGetDescriptor?
prov:PMGetDescriptorResponse?
prov:PMActivate?
prov:PMActivateResponse?
prov:PMDeactivate?
prov:PMDeactivateResponse?
prov:PMRegisterDescriptor?
prov:PMRegisterDescriptorResponse?
prov:PMUpdate?
prov:PMUpdateResponse?
prov:PMDelete?
prov:PMDeleteResponse?
prov:Poll?
prov:PollResponse?
prov:UpdateEPR?
prov:UpdateEPRResponse?
idp:GetAssertion?
idp:GetAssertionResponse?
idp:GetProviderInfo?
idp:GetProviderInfoResponse?
idp:CreatedStatus?
idp:CreatedStatusResponse?
shps>Delete?
shps>DeleteResponse?
shps:GetStatus?
shps:GetStatusResponse?
shps:Query?
shps:QueryResponse?
shps:Invoke?

```
shps:InvokeResponse?
shps:QueryRegistered?
shps:QueryRegisteredResponse?
shps:Register?
shps:RegisterResponse?
shps:SetStatus?
shps:SetStatusResponse?
shps:Update?
shps:UpdateResponse?
shps:Poll?
shps:PollResponse?
shps:ProxyInvoke?
shps:ProxyInvokeResponse?
idhrxml:Create?
idhrxml:CreateResponse?
idhrxml:Query?
idhrxml:QueryResponse?
idhrxml:Modify?
idhrxml:ModifyResponse?
idhrxml>Delete?
idhrxml>DeleteResponse?
idhrxml:Notify?
idhrxml:NotifyResponse?
@id? -> %xs:ID
;

@mustUnderstand -> %xs:boolean
@actor          -> %xs:anyURI
@encodingStyle  -> %xs:anyURI
&@encodingStyle:
  @e:encodingStyle?
;

Fault -> %e:Fault
%Fault:
  e:faultcode -> %xs:QName
  e:faultstring -> %xs:string
  e:faultactor? -> %xs:anyURI
  e:detail? -> %e:detail
;

%detail:
  any*
  @any
;

#EOF
```

6.4 XML and Web Services Infrastructure

6.4.1 xmldsig-core (ds)

```

# xmldsig-core.sg -- Slightly edited after generation
# $Id: xmldsig-core.sg,v 1.3 2007-09-24 02:34:34 sampo Exp $

target(ds, http://www.w3.org/2000/09/xmldsig#)
ns(xs, http://www.w3.org/2001/XMLSchema)
ns(exca, http://www.w3.org/2001/10/xml-exc-cl4n#)
ns(xenc, http://www.w3.org/2001/04/xmlenc#)

%CryptoBinary: base(xs:base64Binary) ;

Signature -> %ds:SignatureType
%SignatureType:
  ds:SignedInfo
  ds:SignatureValue
  ds:KeyInfo?
  ds:Object*
  @Id? -> %xs:ID
  ;

SignatureValue -> %ds:SignatureValueType
%SignatureValueType: base(xs:base64Binary)
  @Id? -> %xs:ID
  ;

SignedInfo -> %ds:SignedInfoType
%SignedInfoType:
  ds:CanonicalizationMethod
  ds:SignatureMethod
  ds:Reference+
  @Id? -> %xs:ID
  ;

CanonicalizationMethod -> %ds:CanonicalizationMethodType
%CanonicalizationMethodType:
  any*
  @Algorithm -> %xs:anyURI
  ;

SignatureMethod -> %ds:SignatureMethodType
%SignatureMethodType:
  ds:HMACOutputLength? -> %ds:HMACOutputLengthType
  any*
  @Algorithm -> %xs:anyURI
  ;

Reference -> %ds:ReferenceType
%ReferenceType:
  ds:Transforms?
  ds:DigestMethod
  ds:DigestValue

```

```

@Id? -> %xs:ID
@URI? -> %xs:anyURI
@Type? -> %xs:anyURI
;

Transforms -> %ds:TransformsType
%TransformsType:
  ds:Transform+
;

Transform -> %ds:TransformType
%TransformType:
  ds:XPath* -> %xs:string
  exca:InclusiveNamespaces?
  any*
  @Algorithm -> %xs:anyURI
;

DigestMethod -> %ds:DigestMethodType
%DigestMethodType:
  any*
  @Algorithm -> %xs:anyURI
;

DigestValue -> %ds:DigestValueType
%DigestValueType: base(xs:base64Binary) ;

KeyInfo -> %ds:KeyInfoType
%KeyInfoType:
  ds:KeyName*
  ds:KeyValue*
  ds:RetrievalMethod*
  ds:X509Data*
  ds:PGPData*
  ds:SPKIData*
  ds:MgmtData*
  xenc:EncryptedKey*
  any*
  @Id? -> %xs:ID
;

KeyName -> %xs:string

MgmtData -> %xs:string

KeyValue -> %ds:KeyValueTypes
%KeyValueTypes:
  ds:DSAKeyValue?
  ds:RSAKeyValue?
  any?
;

RetrievalMethod -> %ds:RetrievalMethodType
%RetrievalMethodType:

```

```

ds:Transforms?
@URI? -> %xs:anyURI
@Type? -> %xs:anyURI
;

X509Data -> %ds:X509DataType
%X509DataType:
  ds:X509IssuerSerial* -> %ds:X509IssuerSerialType
  ds:X509SKI* -> %xs:base64Binary
  ds:X509SubjectName* -> %xs:string
  ds:X509Certificate* -> %xs:base64Binary
  ds:X509CRL* -> %xs:base64Binary
  any*
;

%X509IssuerSerialType:
  ds:X509IssuerName -> %xs:string
  ds:X509SerialNumber -> %xs:integer
;

PGPData -> %ds:PGPDataType
%PGPDataType:
  ds:PGPKeyID? -> %xs:base64Binary
  ds:PGPKeyPacket? -> %xs:base64Binary
  any*
;

SPKIData -> %ds:SPKIDDataType
%SPKIDDataType:
  ds:SPKISexp -> %xs:base64Binary
  any?
;

Object -> %ds:ObjectType
%ObjectType:
  any* processContents(lax)
  @Id? -> %xs:ID
  @MimeType? -> %xs:string
  @Encoding? -> %xs:anyURI
;

Manifest -> %ds:ManifestType
%ManifestType:
  ds:Reference+
  @Id? -> %xs:ID
;

SignatureProperties -> %ds:SignaturePropertiesType
%SignaturePropertiesType:
  ds:SignatureProperty+
  @Id? -> %xs:ID
;

SignatureProperty -> %ds:SignaturePropertyType

```

```

%SignaturePropertyType:
  any+
  @Target -> %xs:anyURI
  @Id? -> %xs:ID
  ;

%HMACOutputLengthType: base(xs:integer) ;

DSAKeyValue -> %ds:DSAKeyValue
%DSAKeyValue:
  ds:P? -> %ds:CryptoBinary
  ds:Q? -> %ds:CryptoBinary
  ds:G? -> %ds:CryptoBinary
  ds:Y -> %ds:CryptoBinary
  ds:J? -> %ds:CryptoBinary
  ds:Seed? -> %ds:CryptoBinary
  ds:PgenCounter? -> %ds:CryptoBinary
  ;

RSAKeyValue -> %ds:RSAKeyValue
%RSAKeyValue:
  ds:Modulus -> %ds:CryptoBinary
  ds:Exponent -> %ds:CryptoBinary
  ;

#EOF

```

6.4.2 xenc-schema (xenc)

```

# xenc-schema.sg -- Slightly edited after generation
# $Id: xenc-schema.sg,v 1.2 2007-09-24 02:34:34 sampo Exp $

target(xenc,http://www.w3.org/2001/04/xmlenc#)
ns(xs,http://www.w3.org/2001/XMLSchema)
import(ds,http://www.w3.org/2000/09/xmldsig#,http://www.w3.org/TR/2002/REC-x\
mldsig-core-20020212/xmldsig-core-schema.xsd)

%EncryptedType:
  xenc:EncryptionMethod? -> %xenc:EncryptionMethodType
  ds:KeyInfo?
  xenc:CipherData
  xenc:EncryptionProperties?
  @Id? -> %xs:ID
  @Type? -> %xs:anyURI
  @MimeType? -> %xs:string
  @Encoding? -> %xs:anyURI
  ;

%EncryptionMethodType:
  xenc:KeySize? -> %xenc:KeySizeType
  xenc:OAEPparams? -> %xs:base64Binary
  any*

```

6.4 XML and Web Services Infrastructure6 APPENDIX: SCHEMA GRAMMARS

```
@Algorithm -> %xs:anyURI
;

%KeySizeType: base(xs:integer) ;

CipherData -> %xenc:CipherDataType
%CipherDataType:
    xenc:CipherValue? -> %xs:base64Binary
    xenc:CipherReference?
;

CipherReference -> %xenc:CipherReferenceType
%CipherReferenceType:
    xenc:Transforms? -> %xenc:TransformsType
    @URI -> %xs:anyURI
;

%TransformsType:
    ds:Transform+
;

EncryptedData -> %xenc:EncryptedDataType
%EncryptedDataType: base(xenc:EncryptedType) ;

EncryptedKey -> %xenc:EncryptedKeyType
%EncryptedKeyType: base(xenc:EncryptedType)
    xenc:ReferenceList?
    xenc:CarriedKeyName? -> %xs:string
    @Recipient? -> %xs:string
;

AgreementMethod -> %xenc:AgreementMethodType
%AgreementMethodType:
    xenc:KA-Nonce? -> %xs:base64Binary
    any*
    xenc:OriginatorKeyInfo? -> %ds:KeyInfoType
    xenc:RecipientKeyInfo? -> %ds:KeyInfoType
    @Algorithm -> %xs:anyURI
;

ReferenceList:
    xenc:DataReference? -> %xenc:ReferenceType
    xenc:KeyReference? -> %xenc:ReferenceType
;

%ReferenceType:
    any*
    @URI -> %xs:anyURI
;

EncryptionProperties -> %xenc:EncryptionPropertiesType
%EncryptionPropertiesType:
    xenc:EncryptionProperty+
    @Id? -> %xs:ID
```

```

;

EncryptionProperty -> %xenc:EncryptionPropertyType
%EncryptionPropertyType:
  any*
  @Target? -> %xs:anyURI
  @Id? -> %xs:ID
  @any?
;

#EOF

```

6.4.3 ws-addr-1.0 (a)

```

# zxid/sg/ws-addr-1.0.sg
# Slightly edited, 5.9.2006, Sampo Kellomaki (sampo@iki.fi)
# 6.2.2007, Added Discovery specifics to the Metadata --Sampo
# $Id: ws-addr-1.0.sg,v 1.9 2007-09-30 05:10:03 sampo Exp $

target(a, http://www.w3.org/2005/08/addressing)
#t arget(a, http://schemas.xmlsoap.org/ws/2004/08/addressing) # used by WS \
Federation?
import(di, urn:liberty:disco:2006-08, liberty-idwsf-disco-svc-v2.0.xsd)
import(e, http://schemas.xmlsoap.org/soap/envelope/)
import(wsu, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecuri\
ty-utility-1.0.xsd,wss-util-1.0.xsd)
ns(sbf, urn:liberty:sb)
ns(b, urn:liberty:sb:2006-08)

&@hdrs:
  @wsu:Id?
  @e:mustUnderstand?
  @e:actor?
  @id? -> %xs:anyURI
  @ID? -> %xs:anyURI
;

EndpointReference -> %a:EndpointReferenceType
%EndpointReferenceType:
  a:Address -> %a:AttributedURIType
  a:ReferenceParameters?
  a:Metadata?
  @notOnOrAfter? -> %xs:dateTime # Added by Sampo
  &@a:hdrs # Added by Sampo
  any* ns(##other) processContents(lax)
  @any
;

ReferenceParameters -> %a:ReferenceParametersType
%ReferenceParametersType:
  &@a:hdrs # Added by Sampo
  b:TargetIdentity*

```

6.4 XML and Web Services Infrastructure6 APPENDIX: SCHEMA GRAMMARS

```
    any* processContents (lax)
    @any
    ;

Metadata -> %a:MetadataType
%MetadataType:
    sbf:Framework?
    di:Abstract?
    di:ProviderID?
    di:ServiceType?
    di:SecurityContext?
    any* processContents (lax)
    @any
    ;

MessageID -> %a:AttributedURIType

RelatesTo -> %a:RelatesToType
%RelatesToType: base (xs:anyURI)
    @RelationshipType? -> %a:RelationshipTypeOpenEnum # default (http://www.\
w3.org/2005/08/addressing/reply)
    &a:hdrs # Added by Sampo
    @any
    ;

%RelationshipTypeOpenEnum: union(a:RelationshipType xs:anyURI) ;
%RelationshipType: enum( http://www.w3.org/2005/08/addressing/reply ) ;

ReplyTo -> %a:EndpointReferenceType
From -> %a:EndpointReferenceType
FaultTo -> %a:EndpointReferenceType
To -> %a:AttributedURIType
Action -> %a:AttributedURIType

%AttributedURIType: base (xs:anyURI)
    &a:hdrs # Added by Sampo
    @any
    ;

@IsReferenceParameter -> %xs:boolean

%FaultCodesOpenEnumType: union(a:FaultCodesType xs:QName)
;

%FaultCodesType: enum( a:InvalidAddressingHeader a:InvalidAddress a:Invalid\
EPR a:InvalidCardinality a:MissingAddressInEPR a:DuplicateMessageID a:Action\
Mismatch a:MessageAddressingHeaderRequired a:DestinationUnreachable a:Action\
NotSupported a:EndpointUnavailable ) ;

RetryAfter -> %a:AttributedUnsignedLongType

%AttributedUnsignedLongType: base (xs:unsignedLong)
    &a:hdrs # Added by Sampo
    @any
```

```

;

ProblemHeaderQName -> %a:AttributedQNameType

%AttributedQNameType:  base(xs:QName)
    &a:hdrs             # Added by Sampo
    @any
;

ProblemHeader -> %a:AttributedAnyType

%AttributedAnyType:
    any* processContents(lax)
    &a:hdrs             # Added by Sampo
    @any
;

ProblemURI -> %a:AttributedURIType

ProblemAction -> %a:ProblemActionType
%ProblemActionType:
    a:Action?
    a:SoapAction? -> %xs:anyURI
    &a:hdrs             # Added by Sampo
    @any
;

#EOF
```

7 Appendix: Some Example XML Blobs

These XML blobs are for reference. They have been pretty printed. Indentation indicates nesting level and closing tags have been abbreviated as "</>". The actual XML on wire generally does not have any whitespace.

7.1 SAML 2.0 Artifact Response with SAML 2.0 SSO Assertion and Two Bootstraps

This example corresponds to `t/sso-w-bootstraps.xml` in the distribution.

Both bootstraps illustrate SAML assertion as bearer token.

```
<soap:Envelope
  xmlns:lib="urn:liberty:iff:2003-08"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soap:Body>

    <sp:ArtifactResponse
      xmlns:sp="urn:oasis:names:tc:SAML:2.0:protocol"
```

```
ID="REvgoIilkzTmk-aIX6tKE"
InResponseTo="RfAsltVf2"
IssueInstant="2007-02-10T05:38:15Z"
Version="2.0">
<sa:Issuer
  xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
  Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
  https://a-idp.liberty-iop.org:8881/idp.xml</>
<sp:Status>
  <sp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></>

<sp:Response
  xmlns:sp="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="RCCzul3z77SiSXqsFplu1"
  InResponseTo="NojFIihxw"
  IssueInstant="2007-02-10T05:37:42Z"
  Version="2.0">
<sa:Issuer
  xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
  Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
  https://a-idp.liberty-iop.org:8881/idp.xml</>
<sp:Status>
  <sp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></>

<sa:Assertion
  xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="ASSE6bgfaV-sapQsAilXOvBu"
  IssueInstant="2007-02-10T05:37:42Z"
  Version="2.0">
<sa:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
  https://a-idp.liberty-iop.org:8881/idp.xml</>

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <ds:Reference URI="#ASSE6bgfaV-sapQsAilXOvBu">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /></>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <ds:DigestValue>r8OvtNmqsLkYwCNg6bsRZAdT4NE=</></>
    <ds:SignatureValue>GtWVZzHYW54ioHk/C7zjDRThohrpwC4=</></>

<sa:Subject>
  <sa:NameID
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
    NameQualifier="https://a-idp.liberty-iop.org:8881/idp.xml">PB5fLIA41RU2bH4HkQsn9</>
  <sa:SubjectConfirmation
    Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <sa:SubjectConfirmationData
      NotOnOrAfter="2007-02-10T06:37:41Z"
      Recipient="https://sp1.zxidsp.org:8443/zxidhlo?o=B"/></></>
```

```
<sa:Conditions
  NotBefore="2007-02-10T05:32:42Z"
  NotOnOrAfter="2007-02-10T06:37:42Z">
<sa:AudienceRestriction>
  <sa:Audience>https://spl.zxidsp.org:8443/zxidhlo?o=B</></></>

<sa:Advice>

<!-- This assertion is the credential for the ID-WSF 1.1 bootstrap (below). -->

<sa:Assertion
  ID="CREDOTGakvhNoPlaiTq4bXBg"
  IssueInstant="2007-02-10T05:37:42Z"
  Version="2.0">
<sa:Issuer
  Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
  https://a-idp.liberty-iop.org:8881/idp.xml</>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <ds:Reference URI="#CREDOTGakvhNoPlaiTq4bXBg">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /></>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <ds:DigestValue>dqg/28hw5eEv+ceFyiLImeJ1P8w=</></></>
      <ds:SignatureValue>UK1EgHKQwuoCE=</></>
    <sa:Subject>
      <sa:NameID/> <!-- *** Bug here!!! -->
      <sa:SubjectConfirmation
        Method="urn:oasis:names:tc:SAML:2.0:cm:bearer" /></>
    <sa:Conditions
      NotBefore="2007-02-10T05:32:42Z"
      NotOnOrAfter="2007-02-10T06:37:42Z">
    <sa:AudienceRestriction>
      <sa:Audience>https://spl.zxidsp.org:8443/zxidhlo?o=B</></></></></>

<sa:AuthnStatement
  AuthnInstant="2007-02-10T05:37:42Z"
  SessionIndex="1171085858-4">
<sa:AuthnContext>
  <sa:AuthnContextClassRef>
    urn:oasis:names:tc:SAML:2.0:ac:classes:Password</></></>

<sa:AttributeStatement>

<!-- Regular attribute -->

<sa:Attribute
  Name="cn"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
  <sa:AttributeValue>Sue</></>
```

```
<!-- ID-WSF 1.1 Bootstrap for discovery. See also the Advice, above. -->
```

```
<sa:Attribute
  Name="DiscoveryResourceOffering"
  NameFormat="urn:liberty:disco:2003-08">
  <sa:AttributeValue>
    <di12:ResourceOffering
      xmlns:di12="urn:liberty:disco:2003-08"
      entryID="2">
      <di12:ResourceID>
        https://a-idp.liberty-iop.org/profiles/WSF1.1/RID-DISCO-sue</>
      <di12:ServiceInstance>
        <di12:ServiceType>urn:liberty:disco:2003-08</>
        <di12:ProviderID>https://a-idp.liberty-iop.org:8881/idp.xml</>
        <di12:Description>
          <di12:SecurityMechID>urn:liberty:security:2005-02:TLS:Bearer</>
          <di12:CredentialRef>CREDOTGAKvhNoPlaiTq4bXBg</>
          <di12:Endpoint>https://a-idp.liberty-iop.org:8881/DISCO-S</></></>
        <di12:Abstract>Symlabs Discovery Service Team G</></></></>
```

```
<!-- ID-WSF 2.0 Bootstrap for Discovery. The credential (bearer token) is inline. -->
```

```
<sa:Attribute
  Name="urn:liberty:disco:2006-08:DiscoveryEPR"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
  <sa:AttributeValue>
    <wsa:EndpointReference
      xmlns:wsa="http://www.w3.org/2005/08/addressing"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0"
      notOnOrAfter="2007-02-10T07:37:42Z"
      wsu:Id="EPRIDcjP80b09In47SDj09b37">
      <wsa:Address>https://a-idp.liberty-iop.org:8881/DISCO-S</>
      <wsa:Metadata xmlns:di="urn:liberty:disco:2006-08">
        <di:Abstract>SYMfiam Discovery Service</>
        <sbef:Framework xmlns:sbef="urn:liberty:sb" version="2.0"/>
        <di:ProviderID>https://a-idp.liberty-iop.org:8881/idp.xml</>
        <di:ServiceType>urn:liberty:disco:2006-08</>
        <di:SecurityContext>
          <di:SecurityMechID>urn:liberty:security:2005-02:TLS:Bearer</>
        </di:SecurityContext>
        <sec:Token
          xmlns:sec="urn:liberty:security:2006-08"
          usage="urn:liberty:security:tokenusage:2006-08:SecurityToken">
          <sa:Assertion
            ID="CREDV6ZBMyicmyvDq9pLIoSR"
            IssueInstant="2007-02-10T05:37:42Z"
            Version="2.0">
            <sa:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
              https://a-idp.liberty-iop.org:8881/idp.xml</>
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
              <ds:SignedInfo>
                <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14#"></ds:CanonicalizationMethod>
                <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"></ds:SignatureMethod>
```


7.3 ID-WSF 2.0 Call with Bearer (Binary) Sec Mech

```

<ds:Reference URI="#TO">...</>
<ds:Reference URI="#ACT">...</>
<ds:Reference URI="#TS">...</>
<ds:Reference URI="#X509">
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>Ru4cAfeBAB</></>
<ds:Reference URI="#BDY">
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>YgGfS0pi56p</></></>
<ds:KeyInfo><wsse:SecurityTokenReference><wsse:Reference URI="#X509"/></></>
<ds:SignatureValue>HJJWbvqW9E84vJVQkjDElqscSXZ5Ekw==</></></>
<e:Body wsu:Id="BDY">
  <xx:Query/></></>

```

The salient features of the above XML blob are

- Signature that covers relevant SOAP headers and Body
- Absence of any explicit identity token.

Absence of identity token means that from the headers it is not possible to identify the target identity. The signature generally covers the Invoker identity (the WSC that is calling the service). Since one WSC typically serves many principals, knowing which principal is impossible. For this reason X509 security mechanism is seldom used in ID-WSF 2.0 world (with ID-WSF 1.1 the ResourceID provides an alternative way of identifying the principal, thus making X509 a viable option).

7.3 ID-WSF 2.0 Call with Bearer (Binary) Sec Mech

```

<e:Envelope
  xmlns:e="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:b="urn:liberty:sb:2005-11"
  xmlns:sec="urn:liberty:security:2005-11"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsa="http://www.w3.org/2005/03/addressing">
  <e:Header>
    <wsa:MessageID wsu:Id="MID">...</>
    <wsa:To wsu:Id="TO">...</>
    <wsa:Action wsu:Id="ACT">urn:xx:Query</>
    <wsse:Security mustUnderstand="1">
      <wsu:Timestamp wsu:Id="TS">
        <wsu:Created>2005-06-17T04:49:17Z</></>
      <wsse:BinarySecurityToken
        ValueType="anyNSPrefix:ServiceSessionContext"
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.xsd"
        wsu:Id="BST">
        mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSr1v4
        YqUc70MiJcBtKBp3+jlD4HPUaurIqHA0vrmdMpm+sF2BnpND118f/mXCv3XbWhiL
        VT4r9ytfpXBluelOV93X8RUz4ecZcDm9e+IEG+pQjnvgrSgac1NrW5K/CJEOUUh
        oGTrym0Ziutezhrw/gOeLVtkywsMgDr77gWZxRvw01wlogtUdTceurBIDANj+KVZ
        vLKlTCaGAUNIjkiDDgti=</>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig #">

```

7.4 ID-WSF 2.0 Call with Bearer (SAML) Sec Mech

```
<ds:SignedInfo>
  <ds:Reference URI="#MID">...</>
  <ds:Reference URI="#TO">...</>
  <ds:Reference URI="#ACT">...</>
  <ds:Reference URI="#TS">...</>
  <ds:Reference URI="#BST">...</>
  <ds:Reference URI="#BDY">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>YgGfS0pi56pu</></></>
  ...</></></>
<e:Body wsu:Id="BDY">
  <xx:Query/></></>
```

7.4 ID-WSF 2.0 Call with Bearer (SAML) Sec Mech

```
<e:Envelope
  xmlns:e="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:sb="urn:liberty:sb:2005-11"
  xmlns:sec="urn:liberty:security:2005-11"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmenc#">
  <e:Header>
    <sbf:Framework version="2.0-simple" e:mustUnderstand="1"
      e:actor="http://schemas.../next"
      wsu:Id="SBF"/>
    <wsa:MessageID wsu:Id="MID">...</>
    <wsa:To wsu:Id="TO">...</>
    <wsa:Action wsu:Id="ACT">urn:xx:Query</>
    <wsse:Security mustUnderstand="1">
      <wsu:Timestamp wsu:Id="TS">
        <wsu:Created>2005-06-17T04:49:17Z</></>
      </>
    </>
    <sa:Assertion
      xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
      Version="2.0"
      ID="A7N123"
      IssueInstant="2005-04-01T16:58:33.173Z">
      <sa:Issuer>http://idp.syndemo.com/idp.xml</>
      <ds:Signature>...</>
      <sa:Subject>
        <sa:EncryptedID>
          <xenc:EncryptedData>U2XTCNvRX7B11NK182nmY00TEk==</>
          <xenc:EncryptedKey>...</></>
        </>
        <sa:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/></>
      </>
      <sa:Conditions
        NotBefore="2005-04-01T16:57:20Z"
        NotOnOrAfter="2005-04-01T21:42:4 3Z">
        <sa:AudienceRestrictionCondition>
          <sa:Audience>http://wsp.zxidsp.org</></></>
        </>
      </>
    </>
  </>
  <sa:AuthnStatement
```

```

    AuthnInstant="2005-04-01T16:57:30.000Z"
    SessionIndex="6345789">
  <sa:AuthnContext>
    <sa:AuthnContextClassRef>
      urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</></></>
  <sa:AttributeStatement>
    <sa:EncryptedAttribute>
      <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
        mQEMAzRniWkAAAEH9RbzqXdgcX8fpEqSrlv4=</>
      <xenc:EncryptedKey>...</></></></>

  <wsse:SecurityTokenReference
    xmlns:wsse1="..."
    wsu:Id="STR1"
    wsse1:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0">
  <wsse:KeyIdentifier
    ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
    A7N123</></>

  <ds:Signature>
    <ds:SignedInfo>
      <ds:Reference URI="#MID">...</>
      <ds:Reference URI="#TO">...</>
      <ds:Reference URI="#ACT">...</>
      <ds:Reference URI="#TS">...</>
      <ds:Reference URI="#STR1">
        <ds:Transform Algorithm="...#STR-Transform">
          <wsse:TransformationParameters>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315">
              <ds:Reference URI="#BDY"/></>
            ...</></></>
          </>
        </>
      </>
    </>
  </>
<e:Body wsu:Id="BDY">
  <xx:Query/></></>

```

*** is the reference above to wsse1:TokenType really correct?

Note how the <Subject> and the attributes are encrypted such that only the WSP can open them. This protects against WSC gaining knowledge of the NameID at the WSP.

References

- [SAML11core] SAML 1.1 Core, OASIS, 2003
- [SAML11bind] "Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1", Oasis Standard, 2.9.2003, oasis-sstc-saml-bindings-1.1
- [IDFF12] <http://www.projectliberty.org/resources/specifications.php>
- [IDFF12meta] Peted Davis, Ed., "Liberty Metadata Description and Discovery Specification", version 1.1, Liberty Alliance Project, 2004. (liberty-metadata-v1.1.pdf)
- [SAML2core] "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-core-2.0-os
- [SAML2prof] "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-profiles-2.0-os

- [SAML2bind] "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-bindings-2.0-os
- [SAML2context] "Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-authn-context-2.0-os
- [SAML2meta] Cantor, Moreh, Phipott, Maler, eds., "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-metadata-2.0-os
- [SAML2security] "Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-sec-consider-2.0-os
- [SAML2conf] "Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-conformance-2.0-os
- [SAML2glossary] "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-glossary-2.0-os
- [XML-C14N] XML Canonicalization (non-exclusive), <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>; J. Boyer: "Canonical XML Version 1.0", W3C Recommendation, 15.3.2001, <http://www.w3.org/TR/xml-c14n>, RFC3076
- [XML-EXC-C14N] Exclusive XML Canonicalization, <http://www.w3.org/TR/xml-exc-c14n/>
- [Shibboleth] <http://shibboleth.internet2.edu/shibboleth-documents.html>
- [XMLENC] "XML Encryption Syntax and Processing", W3C Recommendation, 10.12.2002, <http://www.w3.org/TR/xmlenc-core>
- [XMLDSIG] "XML-Signature Syntax and Processing", W3C Recommendation, 12.2.2002, <http://www.w3.org/TR/xmlsig-core>, RFC3275
- [Disco2] Liberty ID-WSF Discovery service 2.0
- [Disco12] Liberty ID-WSF Discovery service 1.1 (liberty-idwsf-disco-svc-v1.2.pdf)
- [SecMech2] Liberty ID-WSF 2.0 Security Mechanisms
- [SOAPAuthn2] Liberty ID-WSF 2.0 Authentication Service
- [SOAPBinding2] Liberty ID-WSF 2.0 framework document that pulls together all aspects
- [DST21] Liberty Data Services Template 2.1
- [DST20] Liberty DST v2.0
- [DST11] Liberty DST v1.1
- [IDDAP] Liberty Identity based Directory Access Protocol
- [IDPP] Liberty Personal Profile specification.
- [Interact11] Liberty ID-WSF Interaction Service protocol 1.1
- [FF12] Liberty ID Federation Framework 1.2, Protocols and Schemas
- [SUBS2] Liberty Subscriptions and Notifications specification
- [Schema1-2] Henry S. Thompson et al. (eds): XML Schema Part 1: Structures, 2nd Ed., W3C Recommendation, 28. Oct. 2004, <http://www.w3.org/2002/XMLSchema>
- [XML] <http://www.w3.org/TR/REC-xml>
- [RFC1950] P. Deutsch, J-L. Gailly: "ZLIB Compressed Data Format Specification version 3.3", Aladdin Enterprises, Info-ZIP, May 1996
- [RFC1951] P. Deutsch: "DEFLATE Compressed Data Format Specification version 1.3", Aladdin Enterprises, May 1996
- [RFC1952] P. Deutsch: "GZIP file format specification version 4.3", Aladdin Enterprises, May 1996

REFERENCES

REFERENCES

- [RFC2246] TLSv1
- [RFC2251] LDAP
- [RFC3548] S. Josefsson, ed.: "The Base16, Base32, and Base64 Data Encodings", July 2003. (Section 4 describes Safebase64)
- [MS-MWBF] Microsoft Web Browser Federated Sign-On Protocol Specification, 20080207, <http://msdn2.microsoft.com/en-us/library/cc236471.aspx>