TAS³ Architecture and Project

Sampo Kellomäki (sampo@zxidp.org)

Kantara Initiative Conference
10. March, 2010, Hillsboro, OR
Trusted Architecture for Securely Shareable Services

Outline

1. Business Case
2. Architecture at Glance
3. Context and Prior Art
4. Novelty of the Architecture
5. Wire interoperability, many software implementations possible
6. Trustworthy and Secure
7. Technology Demo
8. TAS$^3$ and Kantara
Trusted Architecture for Securely Shareable Services

Who are we?

• EU FP7 financed research program
  - 2 plus years into the project, ending end of 2011
• KU Leuven / Bart Preneel, Brendan Van Alsenoy
• SAP Sophia Antipolis (coordination)
• Oracle / Joseph Alhadeff (legal)
• Synergetics / Luk Vervenne (Commercial)
• Symlabs / Sampo Kellomäki (Architecture)
• Kent / David Chadwick (Authorization)
• CNR Pisa / Antonietta Bertolini (Online Compliance Testing)
• TU Eindhoven / Jerry Den Hartog (Trust scoring, feedback)
• Karlsruhe / Jutta Mülle (Business Processes)
• Koblenz (Dashboard, data layer)
• VUB (Ontologies)
• Nottingham (Employability pilot)
• Custodix (Healthcare pilot, Commercial)
• Risaris (Pilot, Commercial)
• Zaragoza (Usability, Preception)

http://www.tas3.eu/
http://zxid.org/tas3/
Goal

- General architecture with prospect of becoming endorsed and adopted in Europe
- Model for setting up trust networks
- Security layer for building sector specific applications ecosystems
- Grow from sector specific to multipurpose Trust Networks
- Initial aim at employability and health care
- First commercial PoC: Province of Limburg (Maastricht)
Privacy Protection

1. Awareness
   - Self audit (dashboard)
   - Identity mirrors

2. Confidentiality
   - Consent to release
   - Reputation and trust based screening
   - Trust and Privacy Negotiation

3. Control
   - Intended purpose
   - Sticky policies

4. Practise
   - Right to correct or delete, Right to response
   - Trust feedback
Combined approach is needed!

- SSO and Web Services identity plumbing
- Authorization
- Audit
- Credentials issuance, management, and validation
- Trust establishment
- Governance
Trust, Security, Privacy

- Single Sign-On and Identity Web Services
- Enable loosely coupled collaboration
- Ecosystem of providers
- Federation
- Separation of data from applications
  - Controlled reuse of data
- Earn user’s trust, gain adoption
- User centricity
- Privacy protection
- Comply with legal requirements
- User management by home organization
- Convenience
TAS3 Trust Network Domains

Audit

Audit & Monitor

Modelling & configuration Management

Runtime & Enforcement

Model

Organization A Domains

...
Prior Art and Reference Architectures

- Standards compliant
- Leverage existing art where available, adapt it for our novelty
- TAS³ Architecture draws from and is compatible with
  - OASIS SAML 2.0
  - Nessi’s NexofRA
  - Access-eGov Platform Architecture
  - Liberty Alliance’s ID Web Services Framework (ID-WSF 2.0)
  - OASIS XACML 2.0
- TAS³ Architecture is not as abstract as a reference architecture
  - Goal is to drive real, wire interoperable, implementations
Novelty of the Architecture (1/2)

- TAS$^3$ Architecture is novel as a blueprint that brings together
  - Identity Management (IdM)
  - Attribute based access control
  - Business process modelling
  - Ontology
  - Dynamic trust
  - Distributed auditing
  - Legal & Policy
  - Support for multiple policies in different languages

- User transparency features
  - Dashboard
  - User accessible audit trail
  - Automated compliance validation
  - Consent and control of policies
Novelty of the Architecture (2/2)

- Separation of data and processing
- Privacy protection using sticky policies
- Marriage of Trust and Privacy Negotiation with discovery and trust scoring
- Secure dynamic business processes
- Built-in first class support for delegation
- Architecture needs to be instantiated in context of a *business model* and legal / contractual framework
  - Leave many decisions to be decided in that context
  - Many business models are possible
Trustworthy and Secure (1/2)

- Operational, legal, and business model to ensure trustworthiness
  - Responsible entity, Trust Guarantor, ensures "buck stops here"
  - Legal framework developed hand-in-hand with architecture
  - Certification of software and deployments
  - Automated Compliance Validation keeps SPs in line
  - Manual audits complement automated approaches
  - Modeling network and its members provide consistent security configuration

- Legal concerns are built-in from the ground up
- Threat analysis to understand what we are defending against
Trustworthy and Secure (2/2)

• Technical
  - Fully encrypted, fully digitally signed
  - Fully pseudonymous design ensures maximum privacy
  - Fully cross organizational federation model
  - Explicit tokens based audit trail at all layers
  - Explicit authorization at all layers
  - Advanced trust and reputation management
  - Model and ontology driven to ensure accurate implementation

• End-to-End
  - Policies carried along the data
  - Comprehensive solution with all aspects addressed: no gaps
TAS$^3$ Technology Demo

- Multi-tier, recursive / deep, call capability
- Fully dynamic using discovery
- Fully pseudonymous at all layers: no privacy compromise
Front End Service

Web Application

Service Requester

Web Service

Service Application

Service Responder

Service Requester

Data Service

Data storage

Web Service

Service Responder

GRAPHIC (RecursiveProtocolPicv7-recurs-demo)
Implementing TAS³

- Set up legal and governance framework
  - Public Private Partnership

- Standalone server products - or SaaS
  - Identity Provider (IdP), Trusted Third Party (TTP)
  - Disovery Server
  - Delegation Server
  - Policy Decision Point (PDP)
  - Dashboard
  - Online Compliance Testing

- Integration tools for enabling applications
  - Apache integration
  - Java Servlet integration
  - SDKs for various languages
  - Integrated to SOA Gateway and Capitain Casa
• Services
  - Trust Network Management
  - Installation and configuration help
  - Audit services
  - SaaS: IdP, TTP, Discovery, ...
Architecture Drilldown

TAS3 Trust Network Domains

Audit

Organization A Domains

Model

Audit & Monitor

Modelling & configuration Management

Runtime & Enforcement

Organization B Domains

...
Web Service Authorization
Multi-tier Web Service Call
Details of Authorization
Integration

1. Request protected content
2. Single Sign-On (SAML)
3. Discover attribute sources
4. Get attributes
5. Authorize
6. Deliver content w/attributes

ZXID SSO and Attribute Broker
- 1. Request protected content
- 2. Single Sign-On (SAML)
- 3. Discover attribute sources
- 4. Get attributes
- 5. Authorize
- 6. Deliver content w/attributes

Dependency libraries
- zlib
- libcurl
- openssl

ZXID core
- Schema driven XML ENC/DEC

Backend Abstraction
- CoT Mgmt
- Session Mgmt
- Local User Mgmt

Backend Abstraction
- Filesystem (/var/zxid)
- MySQL
- LDAP

User

Browser

Apache httpd   (mod_auth_saml attaches to Apache API hooks)
check_user_id hook
mod_auth_saml
Singe Sign-On (SSO)
Liberty IGF
Attribute Broker
(ID-WSF, local)
Authorization
XACML PEP

ZXID simple API

IdP DS

ID-DAP WSP

XACML PDP

Apache subprocess environment

Static Content
- CGI
- mod_perl
- mod_php

ZXID SSO and Attribute Broker
= To be
implemented

= To be
implemented

©20091029 Sampo

10.3.2010
Sampo Kellomäki: TAS3 Arch 01, Hillsboro
Figure 1: Application Integration using ADPEP and (A) WP8 SOA Gateway, (B) WP8 as frontend to WP8 SOA GW, (C) WP8 database.
Figure 2: Application Integration: ADPEP implemented in application itself.
Figure 3: Application Integration: PEP implemented directly in application.
Front channel and back channel interaction

1. Front Channel, Web GUI Interaction
2. Authentication
3. Back Channel
4. Web Services Layer
5. Audit & Monitor
6. TAS3 TN Model
7. TAS3 TN Compliance, Audit, and Monitor
8. TAS3 TN Model
9. TAS3 TN Compliance, Audit, and Monitor
10. TAS3 TN Compliance, Audit, and Monitor

Org A (Context A)
Org B (Context B)

10.3.2010 Sampo Kellomäki: TAS3 Arch 01, Hillsboro
Model driven configuration

TAS3 TN Model
TAS3 TN Compliance, Audit, and Monitor
Audit & Monitor
Modelling
Org A
(model)
Model
Runtime
FE A1
FE A1
Az
WS A2
WS A2
Audit & Monitor
Org B
(model)
Model
Runtime
DashB
Re B
IdP B
IDMap
Back Channel
Web Services
Layer
Front Channel, Web GUI Interaction
Authentication
1
2, 4
3
10
6
5
7, 9
8
6
Model driven audit

TAS3 CoT Model

Modelling and Configuration Management Domain

- Modelling Tool
- Models and configurations

Discover usage & configuration

Automatically push consistent security configuration

Use model to drive visualization of workflow and system

Modelling and Configuration Management Domain Runtime and Enforcement Domain

Operation Monitoring

Audit and Monitoring Domain

- Auditing & Compliance Tools
- Operation Monitoring

Audit and Monitoring Domain

- Frontend Services
- Dashboard
- Middletier Web Services
- Backend WS

Dashboard

IdP

Disco

Connectors

* = Routing & aggregation

= PEP

10.3.2010 Sampo Kellomäki: TAS3 Arch 01, Hillsboro
TAS3 CoT Model

Audit & Monitor

Modelling

Org A
(Context A)

Core

Summit

Org B
(Context B)

IdP

Disco

FE

WS

WS layer

SSO sub CoT A

SSO sub CoT B
Modelling and Configuration Management Domain

Runtime and Enforcement Domain

Audit and Monitoring Domain

Trust Network level model

Modelling Tool

Models and configurations

Discover actual usage

Frontend Services

Dashboard

IdP

Disco

Trust Network level model

Modelling and Configuration Management Domain

Discover actual usage

Modelling Tool

Models and configurations

Runtime and Enforcement Domain

Trust Network level model

Modelling and Configuration Management Domain

Discover actual usage

Modelling Tool

Models and configurations

Frontend Services

Dashboard

IdP

Disco

Trust Network level model

Modelling and Configuration Management Domain

Discover actual usage

Modelling Tool

Models and configurations
Core Security Architecture Flows

1. User initiates request.
2. Authentication process starts.
3. Single Sign-On (SSO) is used for seamless access.

Front End service A
- IDP_1
- Web GUI
- PDP
- 123A
- Web Application
- Authentication
- PID E(123)A
- Service Requestor
**Acronym Expansion**

**TG** Trust Guarantor, the organization that operates TN ("Summit")

**TN** Trust Network

**IdP** Identity Provider (SAML role, aka authentication authority)

**SP** Service Provider: a member organization of TN that operates Frontend and/or Web Services

**Disco** Service discovery, sometimes specifically identity enabled service discovery such as Liberty ID-WSF Discovery Service.

**DB** Dashboard, a web GUI for viewing audit records, work flow status, and/or viewing and editing privacy settings and permissions.

**FE** Frontend, here means web site, i.e. SP

**WS** Web Service, SOAP based machine to machine communication. Sometimes specifically Identity enabled web service, e.g. Liberty ID-WSF based WS.
**WSC** Web Service Client, aka Service Requester

**WSP** Web Services Provider, Service Responder

**PEP** Policy Enforcement Point

**PDP** Policy Decision Point
**TAS³ and Kantara Initiative (Discussion)**

**eGovt** Potential framework and profile for Governments to Adopt

**UMA** Sticky Policies, user centricity, RESTful bindings

**ULX** Usability is important for trust preception

**IdP Selection** As TAS³ foresees multiple IdPs, this needs to be solved.

**ID-WSF Evolution** TAS³ is really a profile of ID-WSF plus some extensions. Push these for standardization.

- Usage Directive elaboration
- Pushing more complex credentials than just a token
- Credentials of the WSC / Requesting Party
- Trust and Privacy Negotiation

**ID-WSF RESTful** RESTful is in TAS³ v2 scope

- SAML-OAuth hybrid

10.3.2010 Sampo Kellomäki: TAS3 Arch 01, Hillsboro
- UMA

IAF WG Trust Network governance leverages IAF work

IOP and Certification TAS³ Trust Network intake, Online Compliance Testing

API standardisation Any interest? Reference implementation?

ID-HR-XML We are using this
TAS$^3$ and Other Standards

- **OASIS**
  - SSTC: SAML AuthnReq Query extension
  - SSTC: SAML support for XACML
  - Access Control (XACML): Obligations support

- **ISO**
  - Brendan
Apache Integration using mod_auth_saml

• No programming. Just add to your Apache configuration:

```apache
LoadModule auth_saml_module modules/mod_auth_saml.so

<Location /protected>
  Require valid-user
  AuthType "saml"
  ZXIDConf "URL=https://sp.demo.org:8443/protected/
  ZXIDConf "REDIR_TO_CONTENT=1"
</Location>
```

• All applications that support HTTP Basic Authentication will "just work" due to emulation of REMOTE_USER header.
SSO Servlet Approach for Tomcat

```java
01 import tas3.*; // Pull in the tas3.az() API
02 public class appdemo extends HttpServlet {
03 public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
04     String fullURL = req.getRequestURI();
05     if (req.getQueryString() != null) {
06         fullURL += "?" + req.getQueryString();
07     }
08     HttpSession ses = req.getSession(false);
09     if (ses == null) {
10         res.sendRedirect("sso?o=E&fr=" + fullURL);
11         return;
12     }
13 } }
```

10.3.2010 Sampo Kellomäki: TAS3 Arch 01, Hillsboro
res.setContentType("text/html");
res.getOutputStream().print("<title>Demo App Protected Content</title><body><h1>Demo App Protected Content at " + fullURL + "</h1>

String[] val_names = ses.getValueNames();
for (int i = 0; i < val_names.length; ++i) {
    res.getOutputStream().print(val_names[i] + " : " + ses.getValue(val_names[i]) + "\n");
}

// Render logout buttons (optional)
res.getOutputStream().print("[<a href="sso?gl=1&s=sesid">Local Logout</a> | <a href="sso?gr=1&s=sesid">Single Logout</a>]
"
SAML Hello World in PHP, the *tas3_sso()* approach

- 38 lines of PHP code of which only 22 do something (rest are comments or HTML)
- Complete
  - All profiles are handled
  - Single Logout handled
  - Well Known Location (WKL) metadata exchange handled
- Hides SAML protocol details
- This Hello World can be cut-and-pasted into any PHP application
Initialization once

```php
01 <?
02 dl("php_zxid.so"); # Pull in module (.so file)
03 # CONFIG: You must have created /var/zxid directory hierarchy.
04 # CONFIG: You must edit the URL to match your domain.
05 $conf = "PATH=/var/zxid/
          &URL=https://sp1.demo.org:8443/hlo.php";
06 $cf = tas3_new_conf_to_cf($conf);
07 ?>
```

- **PATH** configuration means multiple instances of ZXID can coexist (e.g. virtual hosting of web sites)
- **URL** configuration determines provider ID, can also be configured via `'/var/zxid/zxid.conf'`
Per protected page or until session is bootstrapped

```
08 <?
09 $qs = $_SERVER['REQUEST_METHOD'] == 'GET' ?
10      $_SERVER['QUERY_STRING']
11    : file_get_contents('php://input');
12 $res = tas3_sso_cf($cf, -1, $qs, &ses, 0x1814);
13
14 switch (substr($res, 0, 1)) {
15   case 'L': header($res); exit;
16   case '<': header('Content-type: text/xml'); echo $res;
```

- Read input and call `tas3_sso()` to handle SAML protocol details
- Act on outcome of `tas3_sso()` as indicated by the first letter
  - L: protocol requires redirect, perform it
  - <: Send out XML data (such as Metadata or SOAP response)
The IdP Selection Page

17 case 'n': exit;  # Already handled, do nothing further
18 case 'e':
19 ?>
20 <title>Please Login Using IdP</title>
21 <h1>Please Login Using IdP</h1>
22 <?=tas3_idp_select_cf($cf, null, 0x1800)?>
23 <?
24 exit;

• e: indicates that IdP Selection page needs to be rendered
• `tas3_idp_select()` generates the ZXID standard form
• Alternatively you could supply your own HTML for the form as long as you respect the form field naming convention
Login Successful Case

25 case 'd': break;  # Logged in case -- continue after
26 default: die("Unknown tas3_sso() res($res)"));
27 }
28
29 # Parse the LDIF in $res into a hash of attributes $attr
30
31 foreach (split("\n", $res) as $line) {
32   $a = split(":", $line);
33   $attr[$a[0]] = $a[1];
34 }
35 ?>

- d: login successful, return data is LDIF entry with attributes of SSO
Protected Content with Single Logout and Defederate Buttons

36 <title>Protected content, logged in</title>
37 <h1>Protected content, logged in as <?=$attr['cn']??>
38 <?=tas3_fed_mgmt_cf($cf, null, -1, $attr['sesid'], 0);

- *tas3_fed_mgmt()* generates the Single Log-Out buttons
- This is the place to bootstrap your application’s own session
The LDIF entry is used as convenient format for passing attribute-value pairs from \textit{tas3\_sso()} to application

• Some "attributes" are synthesized, others come actually from assertion
IdP Selection

ZXID SP Federated SSO (user NOT logged in, no session)

Login Using New IdP

A new IdP is one whose metadata we do not have yet. We need to know the IdP URL (aka Entity ID) in order to fetch the metadata using the well known location method. You will need to ask the administrator of the IdP to tell you what the EntityID is.

IdP URL: http://r90s.germany.sun.com:8080/amserver-fep-sym-saml-idp
Entity ID of this SP (click on the link to fetch the SP metadata): https://sp1.zxidsp.org:8443/zxidhlo?o=B

Login Using Known IdP

Login to https://lolo:8681/idp.xml (A2)  Login to https://lolo:8681/idp.xml (P2)
Login to https://idp.symdemo.com:8880/idp.xml (A2)  Login to https://idp.symdemo.com:8880/idp.xml (P2)

Technical options

- Create federation, NID Format: Persistent

zxid.org, 0.18 1178728139 lbzxid (zxid.org)
Login at IdP

Welcome to Id Provider "IdP3 A" Home Login

You may login using various methods (pick your poison)

(be sure browser accepts cookies from the same domain)

1. Cookie login Username: sue Password: ***

If any web site (SP) asks...

The IdP URL (Provider ID/Entity ID) of this IdP is https://a-idp.liberty-io.org:8881/idp.xml

You can cut and paste the above URL to any web site that allows Single Sign-On using IdP URL or "Any IdP" or "Other IdP". This mechanism allows the web site (SP) to dynamically join the Circle of Trust of this IdP. This is called Auto-CoT.
SSO Successful: Protected Page
TAS$^3$ API (Java, PHP, Perl, C / C++)

**tas3_sso()** SSO (with optional application independent authorization)

**tas3_az()** Application Dependent Authorization

**tas3_call()** Web Services Client: call a web service and validate response

**tas3_wsp_validate()** Validate that web service request can be processed

**tas3_wsp_decorate()** Create a web service response
30 if (tas3.az("PATH=/var/zxid/", "Action=Show", 
    ses.getValue("sesid").toString()) == null) {
    res.getOutputStream().print("<p><b>Denied.</b>
    res.setStatus(302, "Denied");
33 } else {
    res.getOutputStream().print("<p>Authorized.
35 }
36
Making web service call

45   ret = tas3.call(cf, tas3.fetch_ses(cf, sid),
46         "urn:hrxml:service", null, null, null,
47         "<idhrxml:Query>"
48         + "<idhrxml:QueryItem>"
49         + "<idhrxml:Select></idhrxml:Select>"
50         + "</idhrxml:QueryItem>" +
51         "</idhrxml:Query>");
Responding to a web service call

```java
public void doPost(HttpServletRequest req, HttpServletResponse res)
  throws ServletException, IOException {
  tas3.tas3_ses ses = tas3.alloc_ses(cf);

  String buf;
  int len = req.getContentLength();
  byte[] b = new byte[len];
  int here, got;
  for (here = 0; here < len; here += got)
    got = req.getInputStream().read(b, here, len - here);
  buf = new String(b, 0, len);
```
String nid = tas3.wsp_validate(cf, ses, null, buf);
if (nid == null) {
    System.err.print("Validate fail buf("+buf+")\n" +
    return;
}
String ldif = tas3.ses_to_ldif(cf, ses);

String ret;
ret = tas3.wsp_decorate(cf, ses, null,
    "<recursed>
    + "<lu:Status code="OK"></lu:Status>
    + "<data>nid="+nid+"\n"+ldif+"\n</data>" +
    "</recursed>");
res.getOutputStream().print(ret);
Thank You

Sampo Kellomäki (sampo@synergetics.be)
+351-918.731.007