Document type: Deliverable
Title: D2.1- TAS\(^3\) Architecture
Work Package: WP2
Dissemination level: Public
Editor(s): Sampo Kellomäki, EIfEL
Preparation date: 1 June 2009
Version: 17 (1.98)

Legal Notice
All information included in this document is subject to change without notice. The Members of the TAS3 Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the TAS3 Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.
## The TAS3 Consortium

<table>
<thead>
<tr>
<th>Participant name</th>
<th>Country</th>
<th>Participant short name</th>
<th>Participant role</th>
</tr>
</thead>
<tbody>
<tr>
<td>K.U. Leuven</td>
<td>BE</td>
<td>KUL</td>
<td>Coordinator</td>
</tr>
<tr>
<td>Synergetics</td>
<td>BE</td>
<td>SYN</td>
<td>Partner</td>
</tr>
<tr>
<td>University of Kent</td>
<td>UK</td>
<td>KENT</td>
<td>Partner</td>
</tr>
<tr>
<td>University of Karlsruhe</td>
<td>DE</td>
<td>KARL</td>
<td>Partner</td>
</tr>
<tr>
<td>Technische Universiteit Eindhoven</td>
<td>NL</td>
<td>TUE</td>
<td>Partner</td>
</tr>
<tr>
<td>CNR/ISTI</td>
<td>IT</td>
<td>CNR</td>
<td>Partner</td>
</tr>
<tr>
<td>University of Koblenz-Landau</td>
<td>DE</td>
<td>UNIKOLD</td>
<td>Partner</td>
</tr>
<tr>
<td>Vrije Universiteit Brussel</td>
<td>BE</td>
<td>VUB</td>
<td>Partner</td>
</tr>
<tr>
<td>University of Zaragoza</td>
<td>ES</td>
<td>UNIZAR</td>
<td>Partner</td>
</tr>
<tr>
<td>University of Nottingham</td>
<td>UK</td>
<td>NOT</td>
<td>Partner</td>
</tr>
<tr>
<td>SAP Research</td>
<td>DE</td>
<td>SAP</td>
<td>Project Manager</td>
</tr>
<tr>
<td>Eifel</td>
<td>FR</td>
<td>EIF</td>
<td>Partner</td>
</tr>
<tr>
<td>Intalio</td>
<td>FR</td>
<td>INT</td>
<td>Partner</td>
</tr>
<tr>
<td>Risaris</td>
<td>IR</td>
<td>RIS</td>
<td>Partner</td>
</tr>
<tr>
<td>Kenteq</td>
<td>BE</td>
<td>KETQ</td>
<td>Partner</td>
</tr>
<tr>
<td>Oracle</td>
<td>UK</td>
<td>ORACLE</td>
<td>Partner</td>
</tr>
<tr>
<td>Custodix</td>
<td>BE</td>
<td>CUS</td>
<td>Partner</td>
</tr>
</tbody>
</table>

## Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joseph Alhadeff</td>
<td>Oracle</td>
</tr>
<tr>
<td>Brendan Vanalsenoy</td>
<td>KUL</td>
</tr>
<tr>
<td>Guglielmo De Angelis</td>
<td>CNR/ISTI</td>
</tr>
<tr>
<td>Michele Bezzi</td>
<td>SAP</td>
</tr>
<tr>
<td>David Chadwick</td>
<td>KENT</td>
</tr>
<tr>
<td>Brecht Claerhout</td>
<td>CUS</td>
</tr>
<tr>
<td>Danny De Cock</td>
<td>KUL</td>
</tr>
<tr>
<td>Seda Gürses</td>
<td>KUL</td>
</tr>
<tr>
<td>Ingo Dahn</td>
<td>UNIKOLD</td>
</tr>
<tr>
<td>Jeroen Hoppenbrouwers</td>
<td>SYN</td>
</tr>
<tr>
<td>Sampo Kellomäki (main contributor)</td>
<td>EIF</td>
</tr>
<tr>
<td>Thomas Kirkham</td>
<td>NOT</td>
</tr>
<tr>
<td>Keqin Li</td>
<td>SAP</td>
</tr>
<tr>
<td>Gilles Montagnon</td>
<td>SAP</td>
</tr>
<tr>
<td>Jutta Mülle</td>
<td>KARL</td>
</tr>
<tr>
<td>Jens Müller</td>
<td>KARL</td>
</tr>
<tr>
<td>Jean-Christophe Pazzaglia</td>
<td>SAP</td>
</tr>
<tr>
<td>Quentin Reul</td>
<td>VUB</td>
</tr>
<tr>
<td>Marc Santos</td>
<td>UNIKOLD</td>
</tr>
<tr>
<td>Luk Vervenne</td>
<td>SYN</td>
</tr>
<tr>
<td>Gang Zhao</td>
<td>VUB</td>
</tr>
</tbody>
</table>
# Table of Contents

**List of Figures** .................................................................................. 8

**Executive Summary** ......................................................................... 11

## 1 Introduction ................................................................................... 13
  1.1 TAS³ Architecture at Glance ....................................................... 13
  1.2 Methodology ............................................................................. 14
  1.3 Normative Claim ........................................................................ 15
  1.4 Review of Previous Work .......................................................... 16
  1.5 Reader’s Guide ........................................................................... 17

## 2 TAS³ High Level Architecture ...................................................... 18
  2.1 Overview ..................................................................................... 18
  2.2 Basic Architectural Entities ....................................................... 19
  2.2.1 Major Components ............................................................... 19
  2.2.2 Enforcement Points on Web Service Call Path .................... 22
  2.2.3 Authorization Subcontinent .................................................. 23
  2.3 Major Flows: Front Channel and Back Channel ....................... 25
  2.4 Overview of Data Models .......................................................... 27
  2.4.1 Federation Relations for Core Security Architecture ............ 27
  2.4.2 Personal Data and Applications .......................................... 29
  2.4.3 Using Sticky Policies to Protect Data ..................................... 29
  2.4.4 Using Encryption to Protect Data .......................................... 29
  2.5 Authorization Process ............................................................... 29
  2.6 Enforcement Process ................................................................. 30
  2.7 Configuration Process .............................................................. 33
  2.8 Audit .......................................................................................... 34

## 3 Core Security Architecture ......................................................... 38
  3.1 Flows .......................................................................................... 38
  3.2 Tokens, Access Credentials ....................................................... 39
  3.2.1 Attribute Pull Model .............................................................. 39
  3.2.2 Linking Service: Attribute Push Model .................................. 49
  3.2.3 Simple Attribute Push Model ............................................... 53
3.3 DELEGATION ................................................................. 54
  3.3.1 Invitation Based Token Approach ................................ 55
  3.3.2 Mandate Tokens Approach ......................................... 60
  3.3.3 Delegation by Direct Authorization Rule .......................... 60
  3.3.4 Delegation by Role Based Authorization Rule .................. 61
  3.3.5 Token Based Delegation to Well Known Delegatee ............ 61
  3.3.6 Token Based Delegation of a Received Role .................... 62
  3.3.7 Multi-layer (Chained) Delegation ................................. 62
3.4 SUBJECT OF THE PII NOT PRESENT - TRANSACTION ............... 62
3.5 BREAK-THE-GLASS AUTHORIZATION .................................. 63
3.6 TRUST AND PRIVACY NEGOTIATION ................................. 64
3.7 INTEROPERATION ACROSS TRUST NETWORKS ....................... 65
  3.7.1 Semantic Interoperability Engine ............................... 65
3.8 PROPERTIES OF WEB SERVICE BINDING .............................. 66

4 APPLICATION SPECIFIC ARCHITECTURE ............................. 68
  4.1 PROTOCOL SUPPORT FOR CONVEYANCE OF STICKY POLICIES .... 68
  4.2 LEGACY INTEGRATION STRATEGY .................................... 69
  4.3 ADPEP ................................................................. 71

5 USING BUSINESS PROCESS MODELLING TO CONFIGURE THE COMPONENTS ...... 73

6 OVERSIGHT AND MONITORING ........................................... 75
  6.1 ON-LINE COMPLIANCE TESTING ..................................... 75
  6.1.1 Involved Actors .................................................... 77
  6.1.2 On-line Testing Process and Architecture ....................... 77
  6.2 OPERATIONS MONITORING AND INTRUSION DETECTION ............ 81
  6.3 LOG AUDIT ............................................................ 81
  6.3.1 Log Collection and Storage ...................................... 83
  6.3.2 Privacy Issues: What to Collect and What to Report ........... 83
  6.4 FORMAL COMPLIANCE AUDITS ...................................... 84
  6.5 ADMINISTRATIVE OVERSIGHT ....................................... 84

7 CONCLUSION: TAS³ IS SECURE AND TRUSTWORTHY .......................... 85

ANNEX A PROTOCOLS AND CONCRETE ARCHITECTURE .......................... 87
  A.1 SUPPORTED AUTHENTICATION AND LOGIN SYSTEMS .............. 87
  A.1.1 System Entity Authentication ...................................... 87
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1.2 SAML</td>
<td>87</td>
</tr>
<tr>
<td>A.1.3 Shibboleth</td>
<td>88</td>
</tr>
<tr>
<td>A.1.4 eID</td>
<td>88</td>
</tr>
<tr>
<td>A.1.5 Smart Cards</td>
<td>88</td>
</tr>
<tr>
<td>A.1.6 One-Time-Password Tokens</td>
<td>89</td>
</tr>
<tr>
<td>A.1.7 OpenID</td>
<td>89</td>
</tr>
<tr>
<td>A.1.8 CardSpace / InforCard and WS-Federation</td>
<td>89</td>
</tr>
<tr>
<td>A.1.9 CA / Netegrity Siteminder Proprietary SSO</td>
<td>89</td>
</tr>
<tr>
<td>A.1.10 Citrix, Sun, and other proprietary SSO</td>
<td>89</td>
</tr>
<tr>
<td>A.1.11 Web Local Login</td>
<td>89</td>
</tr>
<tr>
<td>A.1.12 Desktop Login</td>
<td>89</td>
</tr>
<tr>
<td>A.1.13 Fat Client Login</td>
<td>90</td>
</tr>
<tr>
<td>A.1.14 User Not Present or Batch Operations</td>
<td>90</td>
</tr>
<tr>
<td>A.2 SUPPORTED IDENTITY WEB SERVICES SYSTEMS</td>
<td>90</td>
</tr>
<tr>
<td>A.2.1 Framework</td>
<td>91</td>
</tr>
<tr>
<td>A.2.2 Liberty ID-WSF</td>
<td>91</td>
</tr>
<tr>
<td>A.2.3 Bare WS-Security Header or Simplified ID-WSF</td>
<td>91</td>
</tr>
<tr>
<td>A.2.4 WS-Trust</td>
<td>92</td>
</tr>
<tr>
<td>A.2.5 RESTful Approach</td>
<td>92</td>
</tr>
<tr>
<td>A.2.6 Message Bus Approach</td>
<td>92</td>
</tr>
<tr>
<td>A.3 AUTHORIZATION SYSTEMS</td>
<td>92</td>
</tr>
<tr>
<td>A.3.1 Authorization Queries</td>
<td>93</td>
</tr>
<tr>
<td>A.3.2 Policy Languages</td>
<td>93</td>
</tr>
<tr>
<td>A.4 TRUST AND SECURITY VOCABULARIES</td>
<td>93</td>
</tr>
<tr>
<td>A.4.1 Vocabularies for Authorization</td>
<td>93</td>
</tr>
<tr>
<td>A.4.2 Vocabularies for Basic Attributes (PII).</td>
<td>93</td>
</tr>
<tr>
<td>A.4.3 Discovery Vocabularies</td>
<td>94</td>
</tr>
<tr>
<td>A.4.4 Security and Trust Vocabularies</td>
<td>94</td>
</tr>
<tr>
<td>A.4.5 Audit Vocabularies</td>
<td>94</td>
</tr>
<tr>
<td>A.5 REALIZATION OF THE DISCOVERY FUNCTION</td>
<td>94</td>
</tr>
<tr>
<td>A.6 REALIZATION OF THE TRUST AND PRIVACY NEGOTIATOR FUNCTION</td>
<td>94</td>
</tr>
<tr>
<td>A.7 REALIZATION OF THE AUDIT AND DASHBOARD FUNCTION.</td>
<td>95</td>
</tr>
<tr>
<td>A.7.1 Audit Event Bus</td>
<td>95</td>
</tr>
<tr>
<td>A.7.2 Audit Event Ontology</td>
<td>95</td>
</tr>
<tr>
<td>A.7.3 Dashboard Function</td>
<td>95</td>
</tr>
<tr>
<td>A.8 REALIZATION OF DELEGATION FUNCTION</td>
<td>95</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

**ANNEX B  RESILIENT DEPLOYMENT ARCHITECTURE (NON-NORMATIVE) .................. 97**

B.1  ZERO DOWNTIME UPDATES ................................................................. 97

**ANNEX C  COMPLIANCE REQUIREMENTS .................................................. 99**

C.1  OTHER WORK .................................................................................. 99

C.2  GENERAL COMPLIANCE REQUIREMENTS .......................................... 99

C.2.1  Legal and Contractual Compliance Requirements .......................... 99

C.2.2  General Technical Compliance Requirements ............................. 100

C.3  COMPLIANCE REQUIREMENTS FOR GOVERNING AGREEMENTS .... 102

C.4  COMPLIANCE REQUIREMENTS FOR TRUST GUARANTORS ............. 103

C.5  COMPLIANCE REQUIREMENTS FOR SERVICE PROVIDERS ........... 103

C.6  COMPLIANCE REQUIREMENTS FOR SERVICE REQUESTERS .......... 104

C.7  COMPLIANCE REQUIREMENTS FOR DATABASES STORING PII ......... 104

C.8  GENERAL COMPLIANCE REQUIREMENTS FOR TRUSTED THIRD PARTIES 104

C.9  COMPLIANCE REQUIREMENTS FOR IDENTITY PROVIDER .............. 105

C.10 COMPLIANCE REQUIREMENTS FOR DISCOVERY PROVIDERS .......... 105

C.11 COMPLIANCE REQUIREMENTS FOR TRUST AND REPUTATION PROVIDER 105

C.12 COMPLIANCE REQUIREMENTS FOR AUDIT PROVIDER .................... 105

C.13 TAS³-LITE COMPLIANCE PROFILE .................................................. 105

**ANNEX D  GENERALIZED USE CASES .................................................... 108**

D.1  USER USES SERVICE (FIRST TIME IN THE SESSION) ................. 109

D.2  ALREADY-LOGGED-IN OPTIMIZATION (SSO) .................................... 110

D.3  USER USES DASHBOARD .............................................................. 110

D.4  IdP DETECTED-OPTIMIZATION (SSO) ............................................. 114

D.5  USER USES SERVICE, IDENTITY SELECTOR CASE ...................... 114

D.6  USER USES SERVICE, LOCAL LOGIN CASE ................................... 116

D.7  USER USES SERVICE, PROXY IdP CASE ....................................... 116

D.8  CONSENTING TO PII RELEASE OR MANIPULATION .................... 116

D.8.1 Interaction on Front Channel ..................................................... 117

D.8.2 Interaction on side channel ........................................................ 117

D.8.3 Interaction via Dashboard .......................................................... 117

D.9  USING LINKING SERVICE .............................................................. 120

D.10 CHOOSING AMONG MULTIPLE SERVICE PROVIDERS ............... 121

D.10.1 Simple Choice of Provider ....................................................... 121

D.10.2 Trust and Privacy Negotiation Assisted by User Interaction ........ 122

D.11 USER-NOT-PRESENT TRANSACTION ............................................. 124
D.12 USER PRESENT DELEgATION .................................................. 124  
D.13 USER-NOT-PRESENT DELEGATION ........................................ 125  
D.14 OTHER USE CASE WORK ..................................................... 126  
D.15 FUTURE USE CASE WORK .................................................... 126  

ANNEX E  TAS³ BUSINESS MODEL (NON-NORMATIVE) .......................... 128  

ANNEX F  THREATS (NON-NORMATIVE) .......................................... 129  
F.1 OTHER WORK ................................................................. 129  
F.2 BUSINESS THREATS ........................................................... 129  
F.3 TRUST MODEL THREATS ....................................................... 129  
F.4 ARCHITECTURAL THREATS ................................................... 129  
F.5 TRUST MISCONFIGURATION THREATS ....................................... 129  
F.6 PROTOCOL MISCONFIGURATION THREATS .................................. 130  
F.7 AUTHORIZATION MISCONFIGURATION THREATS .............................. 130  
F.8 ONTOLOGY THREATS ........................................................... 130  
F.9 EXPOSURE THREATS ............................................................ 130  
F.10 PRIVACY THREATS ............................................................. 131  
F.11 AUTHENTICATION THREATS .................................................. 132  
F.12 IMPERSONATION THREATS .................................................... 132  
F.13 REPUDIATION THREATS ........................................................ 133  
F.14 UNAUDITABILITY THREATS ................................................... 133  
F.15 SOFTWARE BUG THREATS ..................................................... 133  
F.16 SERVICE AVAILABILITY THREATS .......................................... 135  

ANNEX G  ENUMERATION OF AUDIT EVENTS ..................................... 136  

ANNEX H  EXAMPLE PROTOCOL MESSAGES (NON-NORMATIVE) .............. 140  
H.1 SAML 2.0 ARTIFACT RESPONSE WITH SAML 2.0 SSO ASSERTION AND TWO BOOTS- 
STRAPS ............................................................ 140  
H.2 ID-WSF 2.0 CALL WITH X509v3 SEC MECH ................................. 144  
H.3 ID-WSF 2.0 CALL WITH BEARER (BINARY) SEC MECH ..................... 145  
H.4 ID-WSF 2.0 CALL WITH BEARER (SAML) SEC MECH ....................... 146  

ANNEX I  GLOSSARY .................................................................... 149  

BIBLIOGRAPHY ............................................................................ 183
0 List of Figures

Figure 2.1: Using TAS3 top level model to start modelling of organizations that participate in Trust Network. ................................................................. 18

Figure 2.2: Major components of Organization Domain. ................................................................. 20

Figure 2.3: Front End calls Web Service, passing through 4 enforcement points. ....................... 23

Figure 2.4: Recursive Web Service calls. ....................................................................................... 24

Figure 2.5: Authorization. ............................................................................................................. 25

Figure 2.6: Front Channel and Back Channel Flows (the numbering indicates typical sequence of events). ........................................................................... 26

Figure 2.7: Audit Event Bus (the numbering indicates typical sequence of events, the e-numbers indicate audit events) ......................................................... 28

Figure 2.8: Authorization Process ................................................................................................. 30

Figure 2.9: Using model to configure Authorization Process. ....................................................... 31

Figure 2.10: Arrangement of enforcement points in web service call flow. .................................. 31

Figure 2.11: Pushing configurations from model. ......................................................................... 33

Figure 2.12: Configuration from Model (the numbering indicates typical sequence of events) .... 34

Figure 2.13: Auditing an authorization decision. ......................................................................... 35

Figure 2.14: Monitoring operation of the network using the configured model. .......................... 36

Figure 3.1: General detailed flow of a service request ................................................................. 38

Figure 3.2: Flow at front channel ................................................................................................. 41

Figure 3.3: Front channel flow when using Identity Selector........................................................... 42

Figure 3.4: Flow of front channel call that makes a call on back channel. .................................... 43

Figure 3.5: Single Sign-On (2,3), Discovery (4), and call to WSP (5). The blue ball represents discovery bootstrap. ................................................................. 45
LIST OF FIGURES

Figure 3.6: Discovery Registration Using Front Channel Interface. ................................. 46
Figure 3.7: Flow of recursive calls on back channel. ....................................................... 47
Figure 3.8: Linking Service: Registration phase .............................................................. 49
Figure 3.9: Linking Service: Login with attribute push phase ............................................. 50
Figure 3.10: The enhanced login screen for attribute aggregation ................................. 51
Figure 3.11: Alice Prepares ePortfolio ............................................................................ 56
Figure 3.12: Bob using Alice’s Web Services by way of invitation ................................... 56
Figure 3.13: Invitation phase ......................................................................................... 57
Figure 3.14: Accept invitation phase .............................................................................. 59
Figure 3.15: Interoperation across contexts ..................................................................... 65

Figure 4.1: Application Integration: PEP implemented directly in application ............... 69
Figure 4.2: Application Integration: ADPEP implemented in application itself .............. 70
Figure 4.3: Application Integration using ADPEP and (A) SOA Gateway, (B) WP9 DB as frontend to SOA GW, (C) WP9 database ................................................................. 70

Figure 6.1: Overview of On-line Compliance Testing ....................................................... 76
Figure 6.2: UML Component Diagram of the On-line Compliance Testing (OCT) ........... 80

Figure A.1: Liberty Alliance Architecture ......................................................................... 92

Figure B.1: Layering of resilience features for Front Channel, Back Channel, and data center Back End services ..................................................................................... 97
Figure B.2: Resiliency implemented using hardware load balancers ............................... 98
Figure B.3: Resiliency implemented using software load-balancing-fail-over functionality and clustering .......................................................... 98

Figure D.1: User accesses Front Ends using Single Sign-On ........................................... 108
Figure D.2: Story board: Using service for 1st time in a session ..................................... 109
LIST OF FIGURES

Figure D.3: Story board: Using further services after logging in at IdP - Single Sign-On (SSO). ........... 111
Figure D.4: Story board: Using Dashboard to audit a business process. ................................. 112
Figure D.5: Story board: Fully automatic login - Single Sign-On (SSO) - when IdP can be detected. .... 114
Figure D.6: Story board: Identity Selector provides IdP User Interface. .................................... 115
Figure D.7: Story board: Using services with local login (not recommended). ............................ 115
Figure D.8: Story board: Login using IdP not trusted by Job Agency. ....................................... 116
Figure D.9: Story board: Presenting a PII consent question in Front Channel interaction. ............. 118
Figure D.10: Story board: Presenting a PII consent question using Side Channel interaction. ......... 119
Figure D.11: Story board: Choice of Service Provider. ............................................................. 122
Figure D.12: Story board: Trust and Privacy Negotiation with User Interaction. .......................... 123
Figure D.13: Story board: Alice invites Bob to view her ePortfolio. ........................................... 125

Keyword List

Architecture, Security, Trust, Privacy
Architecture Executive Summary

This document contains version 1 of the TAS³ system architecture (by system architecture we mean the conceptual design that defines the structure and behaviour of a TAS³ trust network). As the Description of Work states, the TAS³ project's main objective is to provide a next generation trust & security architecture that is ready to (1) meet the requirements of complex and highly versatile business processes, (2) enable the dynamic user-centric management of policies and (3) ensure end-to-end secure transmission of personal information and user-controlled attributes between heterogeneous, context dependent and continuously changing systems. This architecture has been designed to fulfill the above objectives through a combination of:

- providing users with the ability to meaningfully give their consent to the use of their personal information
- ensuring a complete set of audit information is recorded by a TAS³ trust network and that users have the ability to directly or indirectly see the audit information that pertains to their personal information. Note that there will not be a single central audit log. If a person needs to drill down into the distributed audit trail, he will need to be authorised and obtain sufficient permissions to access the various local audit logs in order to correlate the events and see the "big picture".
- a legal framework and set of model contracts that will contractually bind all service providers into operating in a trustworthy manner e.g. so as to honour the choices of users concerning the handling of their personal information
- a set of trusted third parties that facilitate the sharing of trust related information such as public keys, authorization attributes, and reputation information
- strong cryptographic algorithms and privacy preserving protocols
- end to end security through application layer encryption and digital signing
- sticky policies that cryptographically bind data and policies together, along with a policy enforcement infrastructure that controls access to all resources
- quality assurance and testing technology and actors to test if on-line services actually behave in compliance with their specifications.

This architecture document describes the conceptual entities that are needed and the services they should provide in order to operate a TAS³ trust network. These trust and privacy enhancing services include: authorization services, secure business process management services, delegation services, privacy preserving discovery services, identity management services, secure repository services and trust and reputation services. All of these services are usually needed regardless of the applications that might run in a TAS³ trust network. However, small centralized trust networks may be able to dispense with one or more of these trust and privacy enhancing services, e.g. discovery or delegation services, depending upon their requirements.

This architecture contains many novel features such as: a trust infrastructure based on novel metrics, actor behaviour and structural components which can be correlated together, an authorisation infrastructure which supports multiple policy languages and conflict resolution, an obligation infrastructure which enforces privacy throughout the trust network, and a distributed audit system which can
be cross correlated with the necessary permissions. These are described in more detail in the specific work package deliverables.

The TAS³ architecture is designed to be standards, protocol, data and application agnostic so that any protocol capable of implementing the flows and satisfying the service requirements can potentially be used by any application. Annex A maps these services onto the latest state of the art application independent protocols as far as is currently possible. This is to ensure interworking between the prototypes that will be developed in this project. Further standardization effort will be needed in order to fully complete this mapping and this will be documented in a future version of this architecture (or in other TAS³ deliverables).

Annex B shows an example deployment architecture that maximizes a service’s availability and is resilient to both system and network failures including denial of service attacks.

Annex C states the compliance requirements for participants in a TAS³ trust network. Legal, policy and technical compliance requirements are covered.

Annex D provides a set of use cases which allows the reader to see how an end user might use the services of a TAS³ trust network.

Annex E contains the first version of a business model that could be used to successfully operate a TAS³ trust network.

Annex F summarizes the threats that the TAS³ architecture is designed to protect against.

Annex G lists the events that should be captured in the secure audit trails of a TAS³ trust network.

Annex H gives some example protocol messages based on the mapping provided in Annex A.

Annex I provides a glossary of terms.

Scope. The TAS³ project has a narrower scope than the architecture that is documented here. This is natural as the novel research contributions of TAS³ are being made only in some areas of the architecture. However the full architecture needs to be documented as this will be needed both to successfully test the research results and to provide a production service. We present a comprehensive architecture that addresses actual use cases end-to-end, rather than simply an architecture of the services that are within the scope of our research.
1 Introduction

1.1 TAS\textsuperscript{3} Architecture at Glance

The TAS\textsuperscript{3} architecture provides the high level design of an infrastructure intended to provide the next generation of trust & security eco-systems that can (1) meet the requirements of complex and highly versatile business processes, (2) enable the dynamic user-centric management of policies and (3) ensure end-to-end secure transmission of personal information and user-controlled attributes between heterogeneous, context dependent and continuously changing systems.

The technical architecture is built on three foundations: technical, policy and legal.

The technical architecture, introduced and described at a high level in this document, presents the different services that are needed in order to operate a trust network (or eco-system). Other work package deliverables provide more detailed designs of some of these services.

The technical architecture proposes a number of Policy Decision Points (PDPs) that are services capable of evaluating policies of various kinds and returning policy decisions to their callers - the Policy Enforcement Points (PEPs). The correct enforcement of user's policies engenders trust in a network. Many policies in a TAS\textsuperscript{3} trust network will be sticky policies, meaning that the policy and the data to which it pertains, are cryptographically bound together, thereby ensuring that the policy is always there to be correctly enforced. Various types of policy and PDP are envisaged, trust PDPs, privacy PDPs, authorisation PDPs, delegation PDPs etc. Details of these PDPs and the policies they support will be provided in more detail in other workpackage deliverables e.g. from WP4, WP5, and WP7.

The legal framework and set of model contracts will be further developed in WP6. They are being designed to contractually bind all the service providers into operating in a trustworthy manner, for example, so as to honour all the choices of users concerning the handling of their personal information. As many trust enabling factors as possible will be built into the technical infrastructure described in this deliverable, thereby automating the controls and freeing organisations from the worry and overhead of ensuring that they do the right thing. When it is not possible to engender trust through technical controls alone, then legal controls through our model contracts will be used as the controls of last resort.

This architecture document describes a service oriented trust network. All the conceptual entities that are needed to form a trust and privacy preserving secure network operate as service providers and service consumers, and they collaborate together to provide the security services to end users. These trust, privacy and security services are application independent and are designed to ensure that whatever application the user is using, the application and its data are as secure, trustworthy and privacy preserving as is possible, given the risk assessment and cost constraints of the trust network. (We accept that absolute security is both technically impossible and financially unaffordable.)

The trust and privacy enhancing services offered by TAS\textsuperscript{3} include:

- authorization services, whose purpose is to answer the question "is this subject authorised to access this resource in this way"
- authentication services, whose purpose is to identify a subject and validate that the communicating party is indeed the identified subject
- privacy preserving services whose purpose is to provide pseudonymous identities for users and
1.2. METHODOLOGY

minimise the cross linking of identities

• trust negotiation services whose purpose is to determine if the remote communicating party is trustworthy enough to start a dialogue

• secure business process management services whose purpose is to ensure that business processes operate securely, and can be dynamically modified securely

• delegation services, whose purpose is to delegate credentials from a delegator to a delegate

• discovery services, whose purpose is to inform clients where particular services can be found

• trusted registries, whose purpose is to keep a directory of all services in the trust network who are known to provide services conforming to the TAS³ specifications

• attribute authorities whose purpose is to assert that particular users have particular attributes

• identity management services, which are a combination of an authentication service and an attribute authority

• secure repository services, whose purpose is to store users’ personal data securely and give users complete control over who should access their data and how they should handle it once they are given access to it

• trust and reputation services, whose purpose is to answer the question "how trustworthy is this actor (service provider or end user)?"

• secure audit services whose purpose is to keep a tamper resistant record of transactions within the trust network so that legally admissible evidence can be obtained in the case of a dispute.

• on-line compliance testing services whose purpose is to ensure that all the services in a trust network comply with their published specifications and policies.

All of these services are usually needed regardless of the applications that might run in a TAS³ trust network. However, small centralized trust networks may be able to dispense with one or more of these trust and privacy enhancing services, e.g. discovery or delegation services, depending upon their requirements.

The TAS³ architecture is designed to be standards and protocol agnostic so that any protocol capable of implementing the message flows and service requirements of the conceptual service providers can potentially be used by any application. However, in order to ensure interworking between the prototypes being developed in this project, we have had to choose a subset of current state of the art protocols. Annex A maps (some of) our services onto the latest state of the art application independent protocols as far as is currently possible. Further standardization effort will be needed in order to fully complete this mapping and this will be documented in a future version of this architecture (or in other TAS³ deliverables).

1.2 Methodology

In presenting the architecture, we follow FMC (Fundamental Modeling Concepts) [FMC03] methodology for presenting the high level static structure. For flow diagrams we use a mixture of UML [UML2]
sequence diagrams and ad-hoc “white boards”. The richness of the latter allow us to better convey relevant control flow and dataflow aspects simultaneously.

For more detailed descriptions we use UML [UML2] modelling, with occasional ad-hoc diagrams to clarify aspects that are not easily communicated using formalisms.

While we usually define, inline, the terminology we use, the authoritative definitions are in [TAS3GLOS] reproduced in Annex I. All architecture documents use this same Glossary and it will not be duplicated in the individual documents.

The stakeholders in context of TAS³ Architecture are

- Users accessing their own data
- Professionals working on data of others
- Service Providers, TTP Operators, and Trust Guarantor (jointly Deployers)
- Security Officers
- Implementers
- TAS³ Members
- Policy Makers
- EC Framework Program 7.

The TAS³ mandate is to build secure, trustworthy, and user-centric technology ([TAS3DOW] section B.0 “Summary”), thus we have adopted methodology where every composition and flow includes a User facet. Most of the flows are viewed from the User perspective and the business and regulatory aspects are filled in from this perspective. Given that gaining trust of the Users is fundamental to widespread adoption, we have opted to emphasize security, transparency, privacy, and user control when trading off efficiency and simplicity.

This document has two goals: (1) Act as an authoritative and prescriptive definition of the TAS³ architecture, and (2) communicate the architecture to the stakeholders, especially Deployers and Implementers. The latter goal is much in line with “Architect as Communicator” in Fig-1 of [FMC03].

1.3 Normative Claim

This document describes the TAS³ Architecture version 1 in a normative and prescriptive way. Any implementation or deployment claiming “TAS³ compliance” MUST abide by this document as well as Annex A “Protocols”, and Annex C “Compliance”. A deployment usually has to satisfy additional requirements of the Trust Guarantor’s Governance or Consortium Agreement and certification procedures, some of which concern the software implementation and others the organizational properties. Use of TAS³ Brand is governed by a separate TAS³ Brand Agreement.

This document uses the keywords (MUST, SHOULD, etc.) of [RFC2119]. All text is normative unless expressly identified as non-normative. Prose and specification have precedence over examples, which in absence of normative text, should be considered RECOMMENDATIONS. Examples as used in the documents are illustrative of the application of the relevant principles contained in the documents and are not statements of principles.
1.4 Review of Previous Work

TAS$^3$ extends the State of the Art, as established by Identity Web Services Framework [IDWSF08], [HafnerBreu09], the Nessi Reference Architecture [NexofRA09], and Access-eGov Platform Architecture [AeGArch07]. [IDWSF08] includes a high level view, derived from documented requirements, and a low level implementable profile of various specifications backed up by interoperability and certification programs that verify interoperability in real life. [NexofRA09] only provides high level view and does not address identity issues (they even use term “federation” inaccurately, liable to cause confusion with Identity Federations) or interoperable protocol profiles - the definition of NEXOF Compliant Platform (NCP) is too vague and there are no interoperability or certification programs - [NexofRA09] fails to recognize clear prior art in [IDWSF08]. TAS$^3$ extends the State of the Art by combining the web service, or SOA, framework with comprehensive authorization and trust management system, modelling domain, compliance validation (i.e. interoperability), and legal framework - in a whole that is concretely implementable. TAS$^3$ addresses Long lifetime, Different Owners, and heterogenous IT environment concerns listed in [NexofRA09], Section 3.3. NexofRA discovery does not address discovery indexed by identity, though it does address discoverability by developers, which may be important for adoption. [AeGArch07] architecture does not specify any concrete and interoperable implementation profile and its security details are vague. Never-the-less, they mention (but do not normatively reference) SAML SSO (no version), and WS-Security (no specific version or profile). They do recognize need for registry and discovery function, but do not discuss the interesting parts. Overall it appeared that their main ambition is not in architecture. They overviewed existing art and picked SOA and applied it to their problem domain using existing concepts without details research in the architecture area.

They use WSMO (http://www.wsmo.org/) based WSMX (Web Services Execution Environment). The Web Service Modeling Ontology (WSMO) aims at describing Web Services in a machine understandable format, and thus enabling the automatic discovery, selection and composition of Web Service. As a result, WSMO provides a semantic to allow multiple organisations to cooperate for the completion of a service. For example, the Accredetation of Prior Learning APL process [TAS3D91PilotUC] requires multiple organisation to be contacted to build the portfolio of a candidate. WSMO is divided in four core components; namely ontologies, web services, goals, and mediators. The ontology element provides a syntax to describe ontological entities (e.g. concept, relation, axiom), which can then be used to represent the semantic of a domain of discourse. In other words, the ontology provides a common conceptualisation of the domain used the other WSMO components. The web service element semantically defines every aspect relevant to web services, such as functionalities and interfaces. For example, the functionality of a web service is expressed in terms of its capabilities and of the pre- and post-conditions associated to them. The goal element specifies the users’ objectives to be fulfilled by the execution of one or more web services. Finally, the mediator element establishes interoperability between mismatched resources. For example, it resolves mismatches in heterogeneous ontologies by finding mappings between their respective ontological entities.
1.5 Reader’s Guide

This document conforms to the TAS$^3$ project-wide glossary [TAS3GLOS] reproduced in Annex I.

If you are a nontechnical reader you may want to start from Annex D to get overall understanding of the user experience, then skim the main document and perhaps consulting Figs 2.1 and 2.2 may be useful. You should also consult [TAS3BIZ], reproduced in Annex E "Business Model", which gives a good motivation for the work shown here. You may also find [TAS3WP] and web site www.tas3.eu helpful in understanding the overall TAS$^3$ concept.

If you are a researcher, this document is the right place to start to see where your research may fit within the architecture.

If you are a software developer you will want to read this document, but you will also want to read carefully Annex A "Protocols", which details protocol versions and gives suggestions about available open source packages that implement these protocols.

If you are a deployer, you should skim this document, perhaps look at Annex A "Protocols", and then work through Annex C "Compliance" as you prepare for your TAS3 certification.

If you are a reviewer, you should read Section 2 and then any other sections or annexes that interest you.
2 TAS³ High Level Architecture

2.1 Overview

**Basic security measures.** Secure encryption, message digest, and digital signature algorithms are used throughout where applicable. All Users and System Entities are authenticated to appropriate degree. For the latter this means PKI authentication, but for the former anything from passwords to hardware tokens is possible. The details of these algorithms are not repeated here, but are covered in Annex A "Protocols" and Annex C "Compliance".

The TAS³ Architecture is a reusable overarching design that can be instantiated any number of times. It specifies a Trust Network (TN) and the manner in which the players, including Users and Service Providers, interact in the Trust Network. The TN may be composed of several organizations, mainly Service Providers (SPs), each of which may constitute a subnetwork and may participate in several other Trust Networks. The architecture addresses interaction of the subnetworks with each other and the top level Trust Networks. We also foresee multiple Trust Networks coexisting and interacting to various degrees. An organization can simultaneously belong to multiple TNs as long as it can simultaneously satisfy the requirements of each network.

![TAS3 Trust Network Domains](image)

*Figure 2.1: Using TAS³ top level model to start modelling of organizations that participate in Trust Network.*

Each Trust Network works in the legal context defined by its Governance Agreement. This architec-
2.2. BASIC ARCHITECTURAL ENTITIES

ture specifies some functions that are strictly necessary for protocol flows to work, and other functions
that are necessary to satisfy nonfunctional properties like "secure" and "trustworthy". To impose on
the players that the latter functions are implemented as well, we rely on legal obligation that stems
from the Governance Agreement, as well as certification and audit programs, operated by the Trust
Guarantor, to check that the legal obligations are met initially and on continued basis.

**TAS³ Trust Network Domain.** Consider Fig-2.1 where a Trust Network (TN), has chosen to adopt
the overall TAS³ approach (which this and other documents specify). This means that at the “Summit”
there is a Trust Guarantor (TG) who imposes on the TN the rules and model of operation. TG usually
employs a Security Officer to maintain and enforce the model. The individual organizations may also
have Security Officers responsible for their internal modeling and auditing.

**Model.** The Trust Network Domain configuration will be expressed using business process models,
ontologies, and other models. The models are refined by each organization in their Modelling and Con-
figuration Management. There will be several ontologies: architectural roles (e.g. Service Requester,
Services Provider, Identity Provider), security ontology, privacy and data protection ontology and trust
ontology. Payload services may define application specific ontologies, but they are not in scope of the
TAS³ architecture. Ontologies in TAS³ are further discussed in [TAS3D22UPONTO]. Some manda-
tory policies emanating from EU will be modelled by the TAS³ project and incorporated to every TAS³
Compliant Trust Network Model (Req. D1.2-6.15-MinPolicy).

**Audit and Oversight.** The Trust Guarantor in its oversight role will operate compliance validation
and audit functions. Each organization is expected to operate similar functions locally as Audit & Monitor.
The audit trail stays principally within the organization, with Trust Guarantor only seeing pointers.
There are some networkwide reporting and auditing requirements that guarantee that other parties in
the network, and especially users, have enough transparency to operation of each party. This helps
to transparently understand that what has happened is legitimate, prevent fraud, and increase overall
trust in the network - a key business goal of TAS³.

**Runtime and Enforcement** concerns delivering the useful payload services, with appropriate mech-
nisms to authenticate and identify Users and Systems, as well as authorize the operations. Most of
technical realization of TAS³ happens in this area.

**Cross Domain and Cross Context.** TAS³ Architecture expressly enables operation of services
across domains. This can mean several organizations in one Trust Network, or it could even mean
interworking of several Trust Networks.

2.2 Basic Architectural Entities

In this section we drill down in the static component view of TAS³ architecture.

2.2.1 Major Components

Our architecture, see Fig-2.2 starts with User interacting with the Runtime & Enforcement area. Since
TAS³ architecture is user centric, all action starts directly or indirectly with the User. Even offline,
user-not-present, processes are seen to have been authorized by the User at some earlier time.

In the Runtime area, the User will interact with Payload services to obtain the tangible business
benefits that motivated him to use the services in the first place. However, for the Payload to work in
secure and trustworthy manner, services from Infrastructure and Discovery areas are needed. For the
system as a whole to remain secure and trusted, functions in the Audit and Monitor area are needed.
They will receive their input through Audit Event Bus of the Runtime environment.

**Front End Service.** User’s principal point of interaction with the system is a GUI, most commonly a Web GUI. This is a special kind of Service Provider that instead of speaking Web Services, e.g. SOAP, offers a user friendly interface. The Front End Services often call Web Services to perform all or parts of the functionality they provide. It is possible that the GUI is generated to match a Business Process Model.

**Web Service.** Machine accessible endpoint from which data or action services can be obtained. Machine-to-machine nature of Web Services is in contrast with the user-to-machine nature of the Front End Services.

The exact sequence of Web Services called will depend on a business process, whether expressly modelled or implicit to the design of the web services. A business process can encompass several Front Ends and the Web Services they call.

**Business Process Engine** is an orchestrating entity that controls how Front Ends and Service Providers, often Web Services, work together to achieve the objectives of the business process. It is depicted here as being a separate service, but "in process" realizations are equally likely. In such case the Business Process Engine would be inside the Front End Service, perhaps as linked in library. The role of the Business Process Engine is to serve payload business processes. There is a similar Trust Network Process Manager entity that, while technically similar, will exclusively execute business processes critical to the TN itself.

**Dashboard** is an important auditing and trust building feature of the TAS³ Architecture. It is a user
2.2. BASIC ARCHITECTURAL ENTITIES

interface, a Web GUI, that allows the User to understand and audit how the system as a whole uses his Personally Identifiable Information (PII). The Dashboard may also integrate a user interaction facility, PII Consent Service, for asking users consent or other input that is required for a business process to advance. All these features provide transparency. (Reqs. D1.2-2.11-Transp, D1.2-3.3-Dash, D1.2-6.3-WhatHowWhyWho, and D1.2-12.15-Valid)

Identity Provider is the point where Users actually authenticate to the system. After authentication, the IdP issues a Single Sign-On (SSO) token so that the Front End Service can complete the login process. IdP has also an important role in providing Id Mapper bootstrap token for the User.

Authorization. This box actually represents an entire subcontinent of functionality. Authorization is pervasive in TAS3 architecture. This topic is treated in more detail in Section 2.2.3.

Delegation provides mechanisms for one User to allow another User to use FE or WS services on his behalf. Delegation also includes mechanisms for introducing users to one another, such as invites. In some cases User can be replaced in delegation by a juridical person. In delegation both the delegator and delegatee may be authenticated indirectly. A situation similar to delegation arises when User instructs a service to act on his behalf. In this case the delegatee is a system entity, usually a Service Provider, and is authenticated directly. The act-on-behalf delegation is handled by the ID Mapper component. (Req. D1.2-7.1-Deleg)

Trust Reputation encompasses a number of components that deal with gathering reputation data, usually via Audit Event Bus, and computing trust scorings that are then used in Authorization and Trust and Privacy Negotiator components. The trust and reputation system is also used to detect certain classes of fraud (Req. D1.2-7.21-Safe). The architecture and design of this subsystem is further elaborated in WP5 deliverables.

Trust Network Process Manager. There are many maintenance processes that a trust network must realize in order to work dynamically and react to threats rapidly. These include intake process for users (Req. D1.2-6.1-IntakePers), intake and certification process for organizations (Req. D1.2-6.2-IntakeOrg), and user's access to his own data and audit trail (Req. D1.2-6.6-UserAccess). The application specific business processes belong to Business Process Engine, above.

Id Mapper is used to translate User's IM token (Id Mapper bootstrap token) to a token usable for Web Service that is about to be called. Such translation is necessary as the user is known by different pseudonym at different services. This is used to express act-on-behalf relationships where Service Provider (delegatee) wields a token provided by Id Mapper (or in some cases by IdP). (Req. D1.2-2.3-BMs)

Registry Server contains knowledge about which end point serves which type of service for any given User. Typically Registry is queried as a preparatory step of web service call proper, but it could be queried in advance. (Req. D1.2-2.3-BMs)

Linking Service provides a facility for a user to indicate how he wishes his attributes to be aggregated.

Obligations Service (not depicted) provides a way to process many commonly occurring obligations such as data retention limit. Obligation handlers register with the obligations service. The service uses this information to advertise its capabilities in satisfying obligations. This leads to trust and privacy negotiation.

Trust and Privacy Negotiator. This is the server side of the negotiation. Every Service Requester, such as Front End Service, must implement Trust and Privacy Negotiator Client (not shown in the figure). Trust and Privacy Negotiator functions in many ways similar to the registry, but instead of returning all end points, only some are returned based on trust scoring.
2.2. BASIC ARCHITECTURAL ENTITIES

Modelling and Configuration Management is connected to the TN level modelling. It also contains local ontologies, such as trust and privacy ontologies, and local Models and Configurations. All of these may be edited using Modelling Tools. From Models and Ontologies, configuration items can be generated and pushed to the Runtime using Management Event Bus, as governed by the Trust Network Process Manager.

An essential element of this architecture are community-managed ontologies (Model in Fig-2.2), which allows for unambiguous, but flexible, meaning agreement at all times. We can envisage several roles for these ontologies. It first provides a machine-understandable documentation of the architecture as well as a formal vehicle to exchange explicit semantic agreements (i.e. commitments) between partners and, eventually, systems. Thus, these commitments will enable the enforcement of (organisational and/or legal) policies within the TAS3 architecture. For example in Role-Based Access Control (RBAC), the role of a subject need to be provided with some semantics (e.g. a list of attributes) to be able to enforce authorization based on the privileges assigned to that role.

Secondly, the ontologies will assure that relevant parts of the system commit to the same interpretation of possibly ambiguous elements to allow for meaning alignment, certification and early conflict discovery. These ontologies will enable improved understanding; common methods of expressing terms enabling people and organisations to better trust each other in these application environments. TAS3 will integrate these architecture elements into a fully embedded trust framework to automate business processes managing personal information, which will result in considerable societal benefits.

The Semantic Interoperability Engine (Fig-3.15) will facilitate the interoperability across different contexts (e.g. across different organizations). Ontologies are further discussed in [TAS3D22UPONTO].

2.2.2 Enforcement Points on Web Service Call Path

Considering Fig-2.3, a Front End (FE) is composed of a Web GUI, a Web Application (the payload of the front end), and a Service Requester module which is used to call Web Services. The counterpart of the Service Requester is the Service Responder module of the Web Service.

Service Requester is a software module that encapsulates the mechanics of performing a Web Service call. An implementation of the Service Requester module will be provided as a deliverable of the TAS3 Project. However, it is possible to implement this independently as long as all requirements prescribed here are maintained.

Service Responder is a software module that encapsulates the mechanics of accepting a Web Service call and responding to it. An implementation of the Service Responder module will be provided as a deliverable of the TAS3 Project. However, it is possible to implement this independently as long as all requirements prescribed here are maintained.

Traffic Lights

PEPOut-Rq. Service Requester Outbound Policy Enforcement Point (PEP). This PEP is used to check whether data can be submitted to the Web Service, or whether the call can be made at all. The PEP will contact organization’s Master PDP to obtain a policy decision.

PEPIn-Rs. Service Responder Inbound PEP. This PEP is used to check whether data or call can be accepted by the Web Service.

PEPOut-Rs. Service Responder Outbound PEP. This PEP is used to filter the data on responder side and to perform any responder obligations attached to the data. If no data can be returned, an error response will still be returned.

PEPIn-Rq. Service Requester Inbound PEP. This PEP is used to perform any obligations attached to the response.
2.2. BASIC ARCHITECTURAL ENTITIES

Figure 2.3: Front End calls Web Service, passing through 4 enforcement points.

Recursive Call
As shown in Fig-2.4, it is possible to chain web services calls, such that the application layer of upstream server may invoke as client a down stream service. There is no difference whether the Service Requester module resides in right hand side of a Front End or a Web Service, turned into Web Services Client (WSC). This pattern can be repeated in any tree topology to any depth of call - however in practical implementation the call depth MAY be limited to 7 to avoid infinite recursion.

2.2.3 Authorization Subcontinent
Authorization is everywhere in TAS³ Architecture. It often gets rolled up in small, but very meaningful symbol in the architecture. This is why we call authorization a "subcontinent" unto itself. It is described more fully in [TAS3D71IdMANAz]. This section addresses Reqs. D1.2.2.19-AzCredi, D1.2.2.20-Az, D1.2.4.5-ComplyPolicy, D1.2.4.6-BrkGlass, D1.2.6.4-Min, D1.2.7.6-Az.

Fig-2.5 depicts some of the components involved in the authorization. By far the most common case is that some payload service, such as a Front End or Web Service, needs to get an authorization decision and initiates the subflow.

Policy Enforcement Point (PEP). This is a software module usually built into the payload service. There are four fundamental types of PEP, as shown in Fig-2.3: in and out variants on Service Requester and Service Responder sides.

Master Policy Decision Point (Master PDP). The PEP calls Master PDP to obtain the authorization.
decision. Typically each organization will run a Master PDP (though other arrangements are possible). All logic of the authorization decision is masked behind the Master PDP. Thus the exact implementation details of Policy Decision Point Stack are irrelevant for the PEPs. The Master PDP handles coordination and routing of requests to the PDPs in the stack and aggregates the authorization decisions received from the PDP. In a way it can be viewed as a PDP proxy with some smarts in it.

The Master PDP is responsible for arranging Break-the-Glass Authorization, see Section 3.5 and \[\text{TAS3D71IdMANAz}\].

Trust Network PDP processes the policies that are coordinated at the Trust Network level. It can be implemented as a central Trust Network-wide service, or it can be distributed so that there is an instance of a Trust Network PDP at each SP, but the policies are centrally coordinated and pushed to the instances, perhaps using the Trust Network Process Manager.

Organization PDP processes the policies that an organization maintains. These policies may be over and above the the Trust Network-wide policies. The distinction from Trust Network PDP is maintained because the authority for deciding the policies is different.

User PDP function may implement User specific policies, i.e. policies set by the User. This could also involve evaluation of Sticky Policies. In practise, the User PDP may be implemented inside the Master PDP process.

Trust PDP is an interface to the Trust and Reputation Management subsystem which allows the Master PDP to query whether a contemplated action is acceptable from Trust and Reputation perspective. Such query has the advantage that the Trust and Reputation system does not need to
disclose to the Master PDP the exact parameters that lead to this decision. The deliverables of WP5 will elaborate on structure and design of Trust PDP and Trust and Reputation System at large.

**Credential Validation Service (CVS)** is a subsystem that helps PEP to establish the validity of the credentials and attributes it is about to pass to the Master PDP. Typically these are received from front channel interaction or from an earlier web service call. The validation involves checking that they are properly signed and that PKI trust to the signing authority exists. Some namespace and syntax checks may be performed as well. The CVS may call on other components of the architecture to perform its functions.

**Policy Information Point (PIP)** is used to fetch additional attributes that may be needed for policy evaluation. PIP may call, in a recursive manner, on other components of the architecture to perform its functions. Special care needs to be taken in preventing infinite recursion and to ensure that the policies in the recursive levels allow the information to be returned for purpose of policy evaluation. PIP may be called either from PEP or from Master PDP. Exact choice is a question of optimization. The set of attributes needed for policy evaluation is difficult to determine. This is a research problem we hope to solve.

### 2.3 Major Flows: Front Channel and Back Channel

**Implementable Flows.** The flows we present are designed to be implementable with existing state-of-the-art protocols and software stacks. In particular standards based ap-
2.3. MAJOR FLOWS: FRONT CHANNEL AND BACK CHANNEL

Approaches are used for authentication, delegation, token passing, identity mapping, service discovery, authorization, and web services calls. Despite this, the present high level architecture is designed to be standards agnostic so that any protocol capable of implementing the flows and satisfying the requirements can potentially be used. See Annex A "Protocols" for details.

From Fig-2.6 we can identify certain important principles (the authorization process is depicted in summary form as box "Az" to reduce clutter, see Section 2.5 for full description):

1. There can be any number of organizations in the Trust Network and each of these organizations may run a number of web sites (labelled as FE - Front End in the figure), Web Services (WS), and infrastructure services (sometimes called Trusted Third Parties).

2. Some architectural roles, like Identity Provider (IdP) can usefully be operated by several organizations in a Trust Network. The important point is that all the components are part of the model of the Trust Network and subject to its oversight.

3. Users will use their "home" IdP (e.g. IdP provided by their employer or educational institution) for Single Sign-On (SSO), but this does not prevent them from using web sites (labelled as FE - Front
End in the figure, this is often called "front channel usage" or "user present scenario") of the other organizations (Req. 3.1 from D1.4), subject to access control decisions, of course.

4. The usage of a web site often triggers Web Services calls on the back channel. Finding out exactly which servers to contact and what credentials to use is handled by User's Discovery and ID Mapper services ("IDMap" in the Fig-2.6) (Req. 8.1 from D1.4). Usually the Discovery Service is rather tightly coupled to the IdP.

5. It is feasible and common that Web Services can be called across organizational boundaries. Discovery and trust negotiation within the model set by the Trust Network will enable this to be possible.

6. When auditable events happen, in addition to local logging, a summary of the data is sent to the Audit Event Bus. Subscribed to the audit summaries are: (i) User's Dashboard service so that the User can always see what happened and is in control; and (ii) the organizational and Trust Network audit layers. See blue arrows in Fig-2.7.

7. Although all organizations can potentially have all components, the fact that cross organization web site usage and service calls are explicitly provided for, makes it possible for an organization to outsource some, or all, of these services. Or the other way around, some organization may specialize in only providing the infrastructure services. This approach is often desirable to manage conflicts of interest.

This is a very flexible architecture and allows the responsibility for provision of services and infrastructure to be sliced and diced in many ways, according to business needs rather than technical limitations.

2.4 Overview of Data Models

2.4.1 Federation Relations for Core Security Architecture

N.B. On first reading it may be advisable to skip this section as understanding of flows shown in Fig-3.4 will be useful.

One of the fundamental principles of the Core Security Architecture is use of federations, which may support persistent or transient identifiers. When correctly used, these types of identifiers allow privacy to be preserved by not leaking any correlation handles. This section addresses Reqs. D1.2-2.14-Priv, D1.2-7.8-NoColl, and D1.2-7.16-Nym.

In order to implement persistent and pseudonymous federations, the IdP and IM have to keep state. In general, the federation table for an IdP that supports persistent pseudonymous identifiers will hold mappings as follows:

User at IdP1 \(\rightarrow\) [ encrypted pseudonym of user at SPA,  
encrypted pseudonym of user at SPB,  
...  
encrypted pseudonym of user at SPN ]

The federation table for IM needs similar mappings.
Figure 2.7: Audit Event Bus (the numbering indicates typical sequence of events, the e-numbers indicate audit events)

User’s pseudonym at IM --> [ encrypted pseudonym of user at SPA,
    encrypted pseudonym of user at SPB,
    ...
    encrypted pseudonym of user at SPN ]

The IdP and IM may include attribute data in the tokens they emit. This attribute data can be kept in any suitable data structure, usually indexed by user and sometimes by SP, or both.

The IM needs additional data structure to determine what services are available to a User. In its simplest form this would consist of

User’s pseudonym Service Type SP EntityID
-------------------------------
789IM  Role Author. C.example.com
789IM  HR Authority B.example.com
579IM  Role Author. C.example.com
579IM  HR Authority B.example.com
but other more general realizations can include data needed for Trust and Privacy Negotiation phase of Discovery. These will be explored in the Trust and Privacy Negotiation documentation.

An IdP may have a limited form of this table to cover the necessity of emitting IM bootstrap token during SSO.

All parties - IdP, IM, and SP (FE or WS) - need to maintain some metadata about each other. Such metadata may include SOAP endpoints, protocol profiles and bindings to use, etc. These will generally be specified in protocol specific documents as adopted in Annex A "Protocols", but for general idea the reader may want to see [SAML2meta].

There is also the requirement for a user to be able to aggregate his attributes together in order to gain access to web services. This requires an attribute linking service, which is fully described in [TAS3D71IdMANAz].

### 2.4.2 Personal Data and Applications

A SP can use whatever data model it desires (TAS³ Architecture is not prescriptive in this regard) in storing the data about the Users as long as it meets security and privacy guarantees detailed in Annex C "Compliance". The persistent pseudonym of the User suggests an obvious database key, but other arrangements are possible.

TAS³ Architecture foresees aggregation of data from multiple sources with its support for policy aggregation. One common realization of this approach is to consider a document as a collection of external data streams, please see [TAS3D81RepoSW]. This approach will be supported by some of the TAS³ software deliverables (e.g. output of WP8).

### 2.4.3 Using Sticky Policies to Protect Data

Sticky policies can be attached to most data items and are especially foreseen to protect personal data and control its dissemination. The purpose for which the data was collected is expressed as sticky policy. This section addresses Reqs. D1.2-2.21-DataProtLaw, D1.2-6.5-Purpose, and D1.2-4.1-EnfUCPol. Data origin and collection method can also be indicated using sticky policies (Req. D1.2-6.8-UserAccess).

Sticky policies are evaluated as part of the authorization process. They should ideally be bound to the data they protect by encryption and signing solution that would prevent disclosure of the data unless the policy evaluates to permit. However, this is a difficult research problem and will be addressed in other TAS³ deliverables.

### 2.4.4 Using Encryption to Protect Data

All protocol flows use encryption. Usually this will be in form of connection level encryption, but in certain cases application layer public key encryption will be used to protect tokens or attribute data while it is in transit through an intermediary (e.g. IM token when passing through FE).

### 2.5 Authorization Process

This section partially addresses Req. D1.2-6.12-Sec.

Fig-2.8 depicts refined structure of the Authorization Process.
2.6. ENFORCEMENT PROCESS

This section partially addresses Req. D1.2-6.12-Sec.

When a Web Services call is made, there are several control points in the flow, as shown in Fig-2.10.

1. Central notion is that the Web Service PEP ("=" above the WS1 in the figure) calls a Master PDP, which then gathers the authorization from whatever sources it can.

2. Some of the data used for the decision may have come from the Web Service itself (it may have been inline in the Request, or the Web Service may know it otherwise), but if additional data is needed, the Master PDP will contact Policy Information Points (PIPs) as appropriate. (Processing of PIP request itself is an instance of Enforcement and Authorization Process, thus giving all of this rather recursive flavour.)

3. Trust and/or Reputation may be a factor in the authorization decision. This is handled by modelling the Trust and Reputation Provider (labelled as just “Trust” in the figure) as just another PDP that the Master PDP calls. The feedback and inputs to the Reputation computation are not shown here.

4. Given that sticky policies may potentially be written in different policy languages, the Master PDP will detect the language and call appropriate PDP to have the policy interpreted.
2.6. ENFORCEMENT PROCESS

Figure 2.9: Using model to configure Authorization Process

Figure 2.10: Arrangement of enforcement points in web service call flow.
2.6. ENFORCEMENT PROCESS

1. Request is first controlled by a Requester, i.e. on Client side, for being an acceptable request. For example, if the request is about to submit data to the Service Provider, there may be several policies about what can be submitted.

2. The controls can have multiple facets, i.e. the application programmer may have programmed in some implicit policies, the organization that operates the application may have some policies of its own, the Trust Network is certain to have policies, and finally the User himself may have set up some policies (which may involve attaching sticky policies to the data). Conceptually these are addressed by a PEP contacting Master PDP which may contact stake holder specific PDPs. If different stake holder policies result in a conflict, the Master PDP implements a Conflict Resolution Policy to arrive at a decision. An alternative approach is to use Identity Governance Framework [IGF] CARML declaration to set up the PEP, or some part of it.

3. After request has been authorized to send, the Service Provider will examine if the request is acceptable using a similar stack of PEPs. Examination on Service Provider side is the "traditional" enforcement point that most people think about. It filters out inappropriate data requests as well as illicit writes.

4. When preparing to ship response, the Service Provider uses a PEP and Master PDP to further filter the response. Although the request side PEP should have made sure that only legitimate requests can ever get processed at the Service level, the returned results may still need some scrutiny, or this facility can be used to attach obligations and sticky policies to the returned data.

5. When Client receives the response, it examines it with a PEP and Master PDP. Such examination may be necessary to understand if there were sticky policies attached, or to perform obligations. Given the rules under which the Service Provider released the data, it may be that Client finds that it cannot use the data for the intended purpose and therefore has to reject the request.

6. Not depicted, but logically part of the Client Request sending side are also
   a. discovery
   b. trust negotiation and establishment
   c. signing of request

7. Not depicted, but logically part of the Service Provider Request processing are also
   a. trust negotiation and establishment
   b. validation of message structure
   c. signature validation

8. Not depicted, but logically part of the Service Provider Response sending side are also
   a. signing of response

9. Not depicted, but logically part of the Client Response reception side are also
   a. validation of message structure
   b. validation of correlation
   c. signature validation
   d. processing obligations
2.7 Configuration Process

TAS³ is pervasively model driven. Fig-2.11 shows how a business process model can drive auditing processes, or even influence the Dashboard user interface so that Users can visualize the processes. Fig-2.9, shows how models are used to configure policies for the PDPs. It also shows an alternate approach where PEP itself can be directly configured, e.g. using Identity Governance Framework [IGF] CARML and/or AAPML.

From the model the Trust Guarantor is able to derive

- Basic trust configuration, i.e. who belongs to the network
  - a white list of members
  - metadata to configure trust in the members

- Configurations to be pushed to operational elements of the network so that they will consistently enforce the process and trust model and the security model of the Trust Network.

- Operations Monitoring setup, e.g. if alerts are coming from some node, what Networks Operations Centre (NOC) process should they enter, where should they be disseminated, who should see them, and who is responsible for response
On-line compliance testing configuration. This will drive a robot, spider if you like, that will comb through the Trust Network on a regular basis to verify that each service is in compliance with the policies that it publicly manifests.

The Organizations A and B participate in the Trust Network. They also model their business processes, extending and refining the global model. They, too, will benefit from ability to automatically configure and monitor the components of their infrastructure.

2.8 Audit

No central audit log. There will not be any central audit log. Only audit data released routinely out of an organization or Service Provider are references to audit events and anonymized summary data. If an audit needs to drill into the audit trail, the authorized auditor will be given access, upon escalation, to fetch or view the local audit trails and ability to correlate the events to form a "big picture". Without such authorization correlation will not be possible. This principle applies to the User's Dashboard as well.
The audit domain is essential to maintain the validity of the trust fabric in the infrastructure. The domain will receive data on authorisation decision as illustrated in Fig-2.13. This enables the domain to become a central point for monitoring of authorisation processes in individual TAS\textsuperscript{3} instances.

The services in the auditing and monitoring domain will receive other forms of data linked to trust from the TAS\textsuperscript{3} infrastructure. This data will also include information on service invocations and workflow execution. The data from the results of these events will be stored in two main sets of services in the auditing and monitoring domain, there are auditing and compliance tools and operation monitoring tools.

It is important to note that these two sets of information will be handled quite separately. The operation monitoring tools will be operated by the applications code and will be application specific, whereas the auditing and monitoring will be operated by the TAS\textsuperscript{3} security layer and will be application independent.

The data collected from the monitoring in the audits can be then used by elements in the infrastructure such as the Dashboard. This will enable users to look at both how their data has been used in the infrastructure and also if any services have failed in this execution. In cases of failure or rogue behaviour the negative feedback from this can be fed to the Trust and Reputation service.

Some Audit Principles:

- Nobody should be able to tamper with the audit trail without detection. This includes insertion or removal of audit records, altering of audit records and deletion of entire audit files.
2.8. AUDIT

- Nobody should be able to put together the entire audit trail without proper authorization
- When answering a user audit request, the initial answer may have coarse granularity, such as organizations that have accessed. Only upon more thorough, authorized, investigation more detail, such as employees that have accessed would be revealed.

Relevant prior art will be incorporated in a future version of this document including regulatory compliance and best practises from

- Directive 95/46/EC and other guarantees offered by EU wide data protection legislation
- SOX [SOX02]
- SAS70
- COBIT
- ISO/IEC 27001
- Liberty Alliance
  - Identity Assurance Framework [IAF]
2.8. AUDIT

- Identity Governance Framework [IGF] Privacy Constraints

  • COSO
  • PCI DSS [PCI08]
3 Core Security Architecture

This section specifies much of the logistics that allow the identity of the user to be passed around between the architectural entities. This is a nontrivial problem, especially if pseudonymous delegated identity is to be supported, combined with recursive calls.

3.1 Flows

This section addresses Reqs. D1.2-2.14-Priv, D1.2-2.15-Resp, D1.2-2.18-AnCredi, D1.2-2.19-AzCredi, D1.2-2.20-Az, D1.2-6.12-Sec, D1.2-6.17-TechBind, D1.2-7.3-An, D1.2-7.8-NoColl, D1.2-7.16-Nym, D1.2-7.21-Safe, D1.2-4.2-BPPrivacy, D1.2-4.4-CourtProof.

![Diagram of a service request flow](image)

Figure 3.1: General detailed flow of a service request

Fig-3.1 shows the core flow.

1. A client application wishing to call some service in another organization, initiates the call.

2. The Client PEP will enforce outbound authorization decision. To be able to do this, it first engages in Trust and Privacy Negotiation, which is a discovery process, see Section 3.6, and then forwards the request to the web services stack.
3. Web services Stack (the "Stack") will compose a request message including the identity tokens that are needed and signs the message. It then sends the message to the Stack on the service side.

4. The service Stack will authenticate the sending Stack and verify the digital signature. The acceptance of the message will depend on a degree of trust on the signing party, which was established during the Trust and Privacy Negotiation.

5. The service inbound PEP will consult the Master PDP to determine if the service request should be allowed to go forward.

6. The inbound PEP will pass the request to the payload service, which will reply.

7. The outbound PEP of the Service will validate that the data can be released and attach obligations.

8. The Stack at the service correlates the response to the request, signs it and sends the response.

9. The client Stack receives the response, checks its correlation with the request, and verifies the signatures in the response.

10. The client inbound PEP checks that the response is authorized and complies with the obligations that were received.

11. The payload message is passed to the Client Application.

3.2 Tokens, Access Credentials

A central problem in multi-tier (or recursive) web services architecture is propagation of identity, or identity handle, to all tiers, while preserving privacy separation (resilience to collusion) between the parties.

The identity handle can allow, if chosen, linking of user's consecutive visits together so that the service can collect data about the user for future reference and provision of the service. In this case the user is persistently identified, but to preserve privacy, the user will be identified differently towards different parties. This prevents collusion by the parties.

Sometimes it is undesirable for the service to link relate visits of the user together. In this case user is identified transiently, i.e. by one-time pseudorandom identifier (Req. D1.2-7.18-Seq). Within one overall session, user can be identified persistently towards one service while at the same time transiently towards another service.

In general access credentials come in the form of tokens that are digitally signed by a system entity, usually a Trusted Third Party, such as an IdP or ID Mapper service. Reader can use SAML assertion [SAML2core] as a mental model, though this is not the only possible technology choice.

This section addresses Reqs. D1.2-7.4-MultiCred and D1.2-7.18-Seq.

3.2.1 Attribute Pull Model

Target model. Fully capable. All use cases work and best privacy properties. This model has been extensively studied in Liberty Alliance standardization work (n.b. this does not limit its applicability to Liberty ID-WSF - same concept can be implemented using other web service specifications, albeit with lesser maturity). This model addresses minimal
3.2. TOKENS, ACCESS CREDENTIALS

disclosure particularly well, thus contributing to satisfying Reqs. \( D1.2-2.14-Priv \) \( D1.2-2.21-DataProtLaw \), \( D1.2-6.4-Min \), \( D1.2-7.5-Min \), and \( D1.2-7.12-CredStepUp \).

The Pull Model consists of front channel SSO layer and back channel web services layer. The "pull" refers to the strategy where attributes are requested from their authoritative sources only on as needed basis. This has several benefits:

1. Minimal disclosure - only needed attributes are generated and shipped

2. Direct relationship with Attribute Authority. No intermediaries which could gain undue knowledge. This may also reduce crypto overhead as protection against the in-transit man-in-the-middle is not needed.

3. Intermediaries do not need to guess what attributes might be needed down the web services call chain or in a particular variant of a business process.

4. Fully dynamic and recursive operation that supports several Business Process Topologies. At least all forms of sequence (horizontal or vertical) and trees are supported. Support for a DAG would seem feasible. Other topologies need further study.

Use Case User U authenticates with a service provider A through her IdP1. A needs to invoke further service providers with reference to U.

Problem Definition If the trusted architecture uses a unique, even if random and transitory, userID throughout then such a userID would allow multiple parties to collude and correlate all data belonging to U.

Objective The system must avoid producing correlation handles in the process.

Solution Idea Each service provider knows the user by a different random userID, a persistent pseudonym. And these pseudonyms are held by a mapping service. When one service provider wants to pass on the request to another service provider, it can ask mapping service for a lookup of the pseudonymous userID in the target service provider.

Given that the user’s pseudonym at the other provider is encrypted in transit, this solution avoids any service providers sharing correlation handles. (N.B. In this system the two service providers invoking each other's services may still be able to directly collude, see Threat \( T107-LogTokLeak \), but the service providers at the ends of a chain of services where chain length > 2 cannot collude. The solution is to not log the tokens, see \( CR53-DontLogTok \).)

3.2.1.1 Front Channel

Trivial situation is when the payload application consists entirely of a web gui or web site, without any web services call. Never-the-less, this is a very important flow because it is the most common way for Users to interact with the system. It is also a necessary precondition for the web services flows to be initiated and bootstrapped with the necessary tokens, including the IM access token.

Example: In our concrete example U authenticates and makes a service request to A which invokes another service provider B which also contains information about user U.

Assumptions:
IdP has a table that lists the user name and the password of the user.

IdP passes the permanent userID with a given Service Provider to that Service Provider every time the user logs in to the IdP from the Service Provider. The identifier is conveyed in a token, e.g.,&lt;username: sampo; attribute: member of tas3; other: permanent userID of user with different service providers&gt;

The Steps of the Protocol with one layer of Service Provider Invocations

1. User U wants to access service provider A and starts interaction with A.
2. U is redirected to IdP1 (n.b. IdP discovery is not addressed in this flow, though industry standards like [SAML2core] and [CardSpace] do address it)
3. U logs in at IdP1. The authentication method is out-of-scope for this flow.

IdP1 returns two encrypted tokens to A:

- **TokenAuthn** the token contains U’s permanent pseudonymous userID 123A4. It is encrypted such that only A can read it and authenticate the user.
- **TokenIM** the token is encrypted for the IM and contains the permanent pseudonymous userID of U with IM which is 789IM. The token is bound to A (contains an indication that only A can use it towards IM).
3.2. TOKENS, ACCESS CREDENTIALS

A authenticates U with TokenAuthn (possibly Single Sign-On - SSO). If it is a stand alone service, A returns the results of the services to U and A is done.

### 3.2.1.2 Front Channel Using Identity Selector

Identity Selector technology aims at solving the IdP selection problem. The central proposal in the area is InfoCard, which is realized by Microsoft CardSpace and some open source Identity Selectors. InfoCard can be deployed in direct fashion, but the problem has been availability of SAML 2.0 tokens. This is usually solved by deploying InfoCard in a proxy setup, as shown in Fig-3.3.

#### 3.2.1.3 Back Channel, Simple

This flow expands on front channel by adding one web services call on back channel. This section addresses Req. D1.2-3.10-JITPerm.

**Example:** In our concrete example U authenticates and makes a service request to A which invokes another service provider B which also contains information about user U.

**Assumptions:**
3.2. TOKENS, ACCESS CREDENTIALS

Figure 3.4: Flow of front channel call that makes a call on back channel.

- There is service provider IdMapper (IM). Each user usually has one IM that knows the permanent user IDs at the different service providers.
- IdPs know the IMs of the users (there are several ways to know. See section on user registration in this document to be written.)
- IdP/IM produce IM tokens. The IM tokens include the following information (which means this information is known to the IdPs and IMs):
  - IM address
  - the permanent pseudonymous userID of the user at the IM
  - which service provider can use the token
  - how many times and how long the token can be used (some of that could be pushed to a PDP attached to the IM, except the constraint about who can use it)

The Steps of the Protocol with one layer of Service Provider Invocations

1. (Same as above.) User U wants to access service provider A and starts interaction with A.
2. (Same as above.) U is redirected to IdP1 (n.b. IdP discovery is not addressed in this flow, though industry standards like [SAML2core] and [CardSpace] do address it)
3. (Same as above.) U logs in at IdP1. The authentication method is out-of-scope for this flow.

IdP1 returns two encrypted tokens to A:

- **TokenAuthn** the token contains U's permanent pseudonymous userID 123A4. It is encrypted such that only A can read it and authenticate the user.
- **TokenIM** the token is encrypted for the IM and contains the permanent pseudonymous userID of U with IM which is 789IM. The token is bound to A (contains an indication that only A can use it towards IM).

A authenticates U with TokenAuthn (possibly Single Sign-On - SSO).

4. A needs to use other service provider B to complete the services and needs the permanent pseudonymous userID for U in B. For this A passes TokenIM from IdP1 to IM.

   The service provider B is selected based on Trust and Privacy Negotiator's efforts to find a suitably trusted SP from the database maintained by the IM (or some other part of the Discovery functionality). (Req. D1.2-3.10-JITPerm)

   IM decrypts TokenIM from IdP1 and sees the user U registered as 789IM in its database. This with the token serves to authenticate the user to IM. Provided that the expiration time of the token is relatively short, the user can be assumed to be present (User Present scenario).

   B looks up the userID of 789IM for service provider B which is 456B (the lookup values can be in encrypted form).

5. IM encrypts two new tokens for the invoked service providers B and gives them to A.

   - **TokenUIDinB** the token is encrypted for B. The token contains the pseudonymous identity of user U at B. In this case it is: 456B.
   - **TokenIM** the token is encrypted for the IM and contains the userID of U with IM which is 789IM. The token is bound to B.

6. A sends a request to B for a service for U and sends the two tokens from the IM.

   B decrypts the token and recognizes the user as having UserID 456B in its database.

   B sees that 456B is the user U. It calls the authorization function to see if U is authorized. Assuming the answer is granted, the service is provided.

   If B needs to invoke further services with a service provider C it communicates with the IM of U using its TokenIM and repeats the steps 4 through 6. See the recursive case, below.

7. B returns a result to A which completes the service and returns result to user U.

   If a User has multiple IMs, multiple IM tokens would be generated if there was no way to ask User’s choice or other deciding rule to pick just one. This may result in practise nearly all IMs being aware of each other, but this need not always be the case and even partially populated IM matrix would remain useful to the user. Further, the IM matrix may be different for different users.
3.2. TOKENS, ACCESS CREDENTIALS

3.2.1.4 IM Bootstrap Token Minting and Passing through Front Channel

A key complication in the operation of the back channel is how to get the ball rolling, i.e. where do the first tokens come, before we can discover more tokens. The simple idea of just using the front channel token has undesirable privacy ramifications as it would provide a correlation handle between the SP and the discovery.

Such correlation handle can be avoided by bootstrapping procedure where the IdP provides a separate, encrypted, token for access to the discovery. Although SP will be an intermediary in passing the token to the discovery, it can not learn a correlation handle due to the encryption. Consider Fig-3.5 where the Single Sign-On (SSO) assertion (a7n), shown as red oval, is minted by the IdP, with another assertion, the discovery bootstrap token shown as blue ball, in it. The SP will establish session for the User (Principal) using the SSO assertion. When it needs to call a web service, it will extract the bootstrap token and pass it to the discovery.

One might ask how does the discovery know all the services the user has and what identity to include in the token. Many methods are possible, but ultimately the discovery maintains a federation database of pseudonyms at each web service for the user. This is very similar to what IdP maintains and it is not uncommon for IdP and discovery to be operated by the same organization.

One way to create the database is to bulk provision it.

Other way is to have user’s actively register the services they consider theirs. Consider Fig-3.6 where user first (1) visits a service, performing a Single Sign-On, thus establishing his pseudonymous identity at the service. Then (2) user triggers the service to register itself as one of the user’s services. At this point the discovery database records what it should send as users identity in a subsequent web service call. When the call is made, first the discovery step (4) is made to obtain the token and then (5) the actual web service call with the correct identity.

3.2.1.5 Improvement Idea: Late IM Token Request

N.B. The IM does not get used at the last step of the chain. It has to produce \( n + 1 \) tokens for \( n \) invocations. This introduces a slight inefficiency.
An improvement of the efficiency of the process is as follows:

Each service provider is only given the authn token and is not given the IM token. If the service provider can provide the service then no IM token is needed. If the service provider needs to contact another service provider, then it contacts the IM to ask for the ID of the user at the next service provider. It refers to the user using the permanent ID by which the user is known to the IM and B (e.g. 456B for B or 123A for A). In this case 789IM is never known to any of the service providers and is internal to the IM. The IM can use the permanent ID to look up the user, find its local ID (789) then locate the permanent ID at the next service provider and send this encrypted for the next service provider, back to the requesting service provider for it to forward to the new service provider.

Problems with this approach, if there are multiple IMs, the service provider will not know which one to contact. If there is only one IM, this is ok. But, the protocol is not standard. Where as the protocol we defined above is a standard liberty alliance protocol.

3.2.1.6 Back Channel, Recursive

The steps of the protocol with one layer of Service Provider Invocations

1. (Same as above.) User U wants to access service provider A and starts interaction with A.

2. (Same as above.) U is redirected to IdP1 (n.b. IdP discovery is not addressed in this flow, though industry standards like [SAML2core] and [CardSpace] do address it)

3. (Same as above.) U logs in at IdP1. The authentication method is out-of-scope for this flow. IdP1 returns two encrypted tokens to A:
3.2. TOKENS, ACCESS CREDENTIALS

Figure 3.7: Flow of recursive calls on back channel.
3.2. TOKENS, ACCESS CREDENTIALS

**TokenAuthn** the token contains U’s permanent pseudonymous userID 123A4. It is encrypted such that only A can read it and authenticate the user.

**TokenIM** the token is encrypted for the IM and contains the permanent pseudonymous userID of U with IM which is 789IM. The token is bound to A (contains an indication that only A can use it towards IM).

A authenticates U with TokenAuthn (possibly Single Sign-On - SSO).

4. A needs to use other service provider B to complete the services and needs the permanent pseudonymous userID for U in B. For this A passes TokenIM from IdP1 to IM.

The service provider B is selected based on Trust and Privacy Negotiator's efforts to find a suitably trusted SP from the database maintained by the IM (or some other part of the Discovery functionality).

IM decrypts TokenIM from IdP1 and sees the user U registered as 789IM in its database. This with the token serves to authenticate the user to IM. Provided that the expiration time of the token is relatively short, the user can be assumed to be present (User Present scenario).

B looks up the userID of 789IM for service provider B which is 456B (the lookup values can be in encrypted form).

5. IM encrypts two new tokens for the invoked service providers B and gives them to A.

**TokenUIDinB** the token is encrypted for B. The token contains the pseudonymous identity of user U at B. In this case it is: 456B.

**TokenIM** the token is encrypted for the IM and contains the userID of U with IM which is 789IM

The token is bound to B.

6. A sends a request to B for a service for U and sends the two tokens from the IM.

B decrypts the token and recognizes the user as having UserID 456B in its database.

B sees that 456B is the user U . It calls the authorization function to see if U is authorized. Assuming the answer is granted, the service is provided.

7. In course of providing the service, B wishes to call C. This is termed “recursive call” and such pattern can occur to any depth. B starts by discovering service of type “Role Authority”, sending the IM token to the Identity Mapper.

8. The Identity Mapper decrypts the IM token and recovers the pseudonymous persistent ID 789IM, which is then used to locate from the database of IM the pseudonym of the User at service C, which is the only service of type “Role Authority” registered for the User. Identity Mapper returns two tokens: (i) the pseudonym of user at C encrypted such that only C can open it (“E(fgh)C” in figure), and (ii) IM token that C may use to make further web services calls.

9. B calls C, passing the tokens along.

10. C decrypts the token "E(fgh)C" and recovers the persistent pseudonym "fgh". It uses this key to look up the role from the database and returns it to B.

11. B uses the role to authorize the request (6) and returns a result to A which completes the service and returns result to user U.
3.2. TOKENS, ACCESS CREDENTIALS

Steps 7 through 10 can be repeated any number of times in a recursive fashion.

3.2.2 Linking Service: Attribute Push Model

This section addresses Req. D1.2-7.15-PushCred.

The Linking Service model is described more fully in [TAS3D71IdMAnAz]. We just give a brief summary of the model here.

The Linking Service model is based on the following assumptions/requirements:

- users typically have multiple attributes (authorisation credentials) assigned by multiple authorities and they are known by different identifiers at each authority
- some service providers will require many of these credentials in order to grant access
- the user does not want the inconvenience of having to authenticate (login) to each of the attribute authority in order to obtain credentials to give to the service provider
- the service provider does want strong cryptographic evidence that each of the authorisation credentials does belong to the user who has initiated the session
1. User makes a service request. 2. User is redirected to her chosen IdP 3. User authenticates to IdP 1. 4. IdP 1 returns an authentication statement + attribute assertions + referral to linking service 5. SP follows referral 6. Linking service looks up IdP 1:PID 1 of user and finds links to other IdPs. 7. Linking service requests attributes from linked IdPs using respective PIDs 8. IdPs return signed and encrypted (to SP) attribute assertions. 9. Linking service relays all attribute assertions to SP.

Figure 3.9: Linking Service: Login with attribute push phase.

- the user should be able to set up his policy for which attributes are aggregated by which SPs
- the user should be able to provide consent each time his attributes are aggregated by an SP.

The Linking Service is a new component that is under the control of the user, and allows the user to set his link release policy for which of his IdPs may be linked together so that their attributes can be aggregated and sent to the same SP. The user may in addition set an attribute release policy at each of his IdPs that is authoritative for more than one attribute, to say which SP can receive which subset of his attributes from this IdP. Taken together, the link release policy and the set of attribute release policies give the user complete control over which of his attributes can be aggregated together and released to which service providers. The GUI for the link release policy has already been described in Section D.9. The GUI for the attribute release policy will be very similar to this, but instead of associating IdPs with SPs, the user will associate attributes with SPs. Note that in many cases attribute release policies will not be needed since most IdPs are typically only authoritative for one attribute.

Once the user has linked his IdPs together and set his link release policy at the Linking Service, the user contacts an SP for a service. The front channel steps are as follows

1. User contacts SP asking for a Service
2. SP and/or user interact, allowing IdP choice as usual
3. User is redirected to chosen IdP and Authentication with the IdP takes place as usual

4. In addition the User can tick a box giving consent for attribute aggregation to take place in this session (see Fig-3.10)

5. User is redirected back to SP and is granted access to the service.

Welcome. Login please.

[Fields for Username, Password, and a checkbox for attribute aggregation]

Figure 3.10: The enhanced login screen for attribute aggregation.

The back channel communications that take place between steps 4 and 5 above are as follows:

1. The IdP sends an authentication SSO statement to the SP, containing a random identifier for the user. This prevents the SP from correlating the user’s requests in different sessions. (Note that if the user wishes to be correlated he can arrange for the linking service to send a unique identifier attribute from one of his attribute authorities during the attribute aggregation process, for example, a National Health Number from the Health Authority if the SP is a medical application, or a unique ID from the authenticating IdP). The IdP also sends an attribute statement containing the user’s attributes at this IdP, and an attribute statement containing referral(s) to the user’s linking service(s). Each referral contains the unique ID of the user as known by the Linking Service and it is encrypted to the public key of the Linking Service.

2. The SP acts on the referral(s) and contacts the LS’s discovery service asking it to return the linked IdPs’ discovery services, along with a Boolean saying “I will do it or You do it for me”.

3. The LS decrypts the unique ID of the user, looks this up in its database and finds all the user’s linked accounts. If the SP has asked to perform aggregation itself, the linking service returns a Response containing referrals to the discovery services of the user’s linked IdPs.

4. The SP now sends a query message to each of the IdP’s discovery services, requesting the contact details of the user’s attribute authority. Alternatively, if the linking service is performing the aggregation on behalf of the SP, it sends the same message to each IdP.

5. The IdP’s discovery service locates the user’s local account by decrypting the user’s unique ID in the referral, and maps the random identifier from the authentication assertion into the user’s local account id. The IdP returns a Response containing the contact details of the attribute authority where the random identifier is now valid.

6. The SP or linking service sends an Attribute Query message to the attribute authority, using the random identifier, whereupon the attribute authority returns a digitally signed attribute assertion encrypted so that only the SP can read it.

7. If the linking service is doing the aggregation, it collects together all the encrypted responses from all the IdPs and then forwards the complete package to the SP.
3.2. TOKENS, ACCESS CREDENTIALS

8. The SP now has the following digitally signed assertions:

a. An authentication assertion from a trusted IdP saying that the user has been authenticated, and is to be known by this random id for this session

b. A set of attribute assertions from trusted attribute authorities saying that the user known by this random id possesses this set of attributes

c. Based on the above the SP can authorize the user to access the requested resources, sure in the knowledge that trusted authorities have both authenticated the user and assigned attributes to her.

3.2.2.1 N-Tier Linking Service Model

This section addresses Req. D1.2-7.1-Deleg.

If the SP above needs to subcontract one or more tasks to a backend service, and that backend service is to act from an authorization perspective as if the user herself had contacted it directly, then the backend service will need to be given the user's authorization attributes. The exact model for how this is to be done is for further study in the following years of the project. Whilst there have been many previous models for dynamic delegation of authority none of them to the best of our knowledge have supported attribute based delegation simultaneously with privacy protection of the delegator's and delegate's identities. The following models at least are suitable for further study:

A. Trusted SP. The first SP simply forwards the initial authentication statement and referrals from the authenticating IdP to the backend SP, and the backend SP follows exactly the same process as the first SP (described above). (Note that the attribute statements cannot be forwarded as these were targeted specifically for the first SP and encrypted to it.) The disadvantages of this method are:

a. the backend service must trust that the first SP is authorised to forward the user's authentication statement to it. Otherwise a rogue SP could illegally forward the authentication statement to a backend SP in order to defraud the user;

b. the user either has to know beforehand which backend services are going to be contacted, so that she can proactively sets up specific Link Release Policies for these backend SPs, or if she does not know or is unaware that backend services are to be involved, she has to set up a default policy for All Other Services (as described in Section D.9). Otherwise the backend SP is likely to deny access to the subcontracted task.

B. Direct Delegation. The user contacts a delegation service and delegates her attributes to the first SP, bestowing these credentials with delegation rights. The first SP uses these credentials to authorize the user, and then delegates them further to the backend SP, optionally bestowing further delegation rights on the backend SP in case it needs to subcontract further. The advantage of this model is that the backend SP does not need to trust the first SP as much, since the latter has specifically been delegated rights to it by the user. Further research is still needed here, since it is currently not known precisely how to delegate an aggregated set of attributes issued by multiple attribute authorities when the user is known by different identifiers at each authority. We will need to find a way to combine the linking and delegation services to work together in a trustworthy manner.

C. FileSpace. We use a completely different model for multiple credential authorisation, termed FileSpace [Chadwick09]. This gives the user a set of files which he can copy freely between
devices and service providers. Service providers can also freely copy the files between each other
to delegate authority on behalf of the user. The files are actually encrypted authorization creden-
tials signed by their respective attribute authorities. Consequently they are useless to the recipient
(who can be a thief or a genuine SP) until it gets the decryption key. Herein lies the twist. The
user is the only person with the private key that can decrypt them. Thus before any front end or
backend SP can utilise the credentials they must ask the user for the decryption key. Once they
have the decryption key they know that (a) the user is the genuine owner of the credentials as she
was able to decrypt them, (b) the attributes genuinely belong to the user because they are signed
by a trusted attribute authority, and (c) the user has given consent for them to be used (because
she provided the decryption key). If we place the user's private key in the Dashboard, then each
SP that wants to authorize the user must first contact the Dashboard asking for a decryption key
and the user can keep track of progress of his task as various events arrive at the Dashboard. She
can then give consent (or not) as appropriate.

D. Shibboleth Portal. The Shibboleth community has developed a delegation model, initially targeted
at web portals and portlets, but generalized so that it can be used for any web services based
delegation. It is based on the Liberty Alliance ID-WSF Enhanced Client or Proxy SSO Profile
[SOAPAuthn2] instead of the SAML 2.0 Web Browser SSO Profile [SAML2prof] which Shibboleth
currently uses for SSO. It works by the first SP going back to the user’s IdP to authenticate itself
and ask for a new authentication statement allowing it to become a delegate of the user and a
deleagor to a backend SP. The user's initial authentication exchange with the IdP is enhanced to
allow the user to specifically authorize delegation by the SP it is contacting. This causes the IdP
to insert a new token into the authentication statement which authorizes the recipient (the SP) to
call it back to become a delegate. After the SP has authenticated and authorized the user, the SP
contacts a backend SP and quite naturally is required to authenticate to the latter, as is any user.
So the SP then contacts the IdP, authenticates to it (say by using mutual TLS) and passes the
token from the user’s authentication statement. This notifies the IdP that the SP is entitled to act
as a delegate of the user, so it issues an authentication statement in the name of the original user,
to the SP. This statement is targeted at the backend SP, and again contains a token that allows
the backend SP to call it back to become a delegate of the user in future. The first SP passes the
authentication statement to the backend SP and thereby masquerades as the user.

E. Subcontracting. Of course the first SP can always subcontract the backend SP to perform the task
on its own behalf (as is typically done in manufacturing supply chains today), in which case the
user’s attributes are not needed by any backend service.

3.2.3 Simple Attribute Push Model

Recommended approach for initial deployments that have not yet developed full infras-
tructure.

In this model some commonly needed, or "enabler", attributes such as Trust Network membership
or role are supplied directly as part of Single Sign-On (SSO) or web service tokens. Other perhaps
justifiable attributes, that do not provoke overdue privacy or legal implications, could be

- legally nonbinding nickname for greeting user
- user’s preferred language
3.3. DELEGATION

This model implies that IdP or IDMapper assume some of the responsibilities of an Attribute Authority. This is well supported in existing protocols and available software implementations. It is also probably the largest operation model in use today in existing federations. For example, this is the model used by all Shibboleth implementations such as the UK academic community federation which has over 800 IdPs and SPs since it was launched in August 2008. The number is still continuing to grow linearly with another 20 or so providers being added per month.

Drawbacks of this approach are

1. Only a very narrow set of attributes will be universally needed by nearly all Front Ends or Web Services.

2. Danger of nonadherence to minimal disclosure principles - its easy to have creep where "just one more" attribute is added to support "just one more" application. This is also wasteful in that cost for generating attribute statements that are seldom needed is still paid on every transaction.

A solution to this is to have an Attribute Release Policy (ARP) at the IdP which provides rules for which attributes should be released to which SPs. In this way the attributes can be effectively filtered before release. The ARP is set by the user and/or the IdP itself, and open source software does exist for this. The design is very similar to the IdP Release Policy of the Linking Service described in Section 3.2.2, above. Still, this approach lacks granularity as the attribute needs of a SP are assumed to be always the same, while in reality SP may run various different business processes with different needs.

3. Postponement of moving to full pull model.

3.3 Delegation

This section addresses Reqs. D1.2-3.7-Deleg and D1.2-7.1-Deleg.

Technical clarification: Here "Delegation" means assignment of decision powers, and access rights they imply, from one User to another User. This is distinct from merely instructing some web application or service to act on ones behalf - some other practitioners call also this "delegation", but we find a service merely executing on instructions of a user so common that it is the base case, and does not need special mention.

Delegation is analogous to giving personal banker the right to manage your portfolio, while action on behalf happens when bank executes wire transfer upon you direct orders.

It should be noted that some system entities may be modelled as juridical persons and can, thus, participate in Delegation like the Users can.

General properties of delegation are

- Express and auditable act of creation (e.g. issue power of attorney), with indication of registration (where needed).
  - Specification of delegatee ("Performer" of a web service action)
  - Specification of delegator ("Target" of a web service action)
  - Specification of scope
3.3. DELEGATION

- Actions and/or
- Resources
- Role based delegation
- Specification of expiry and other constraints

- Sub-delegation, when this ability is expressly mentioned in constraints
- Ability to revoke
  - At any step of sub-delegation chain
  - Any superior ancestor can revoke any descendant
- Audit
- Divulgence of issuer's signing private key MUST NOT be used as a mechanism of delegation.
- Expression of the delegation and target in web services calls at any level of recursion
- Verification by Relying Party: mandate assurance & authenticity (typically verify sig on token + query of MA for revocation info + evaluation of possible constraints)
- Transparency: ability for user to verify which mandates have in fact been exercised or formally accepted.

This could be implemented by the Delegation Service feeding information about each invitation usage to the Audit Event Bus, where the Dashboard can pick up the information and display it to the user when user comes to consult it. Also, when Service Responder, or its CVS or PDP, consumes a delegation token, it will inform the Audit Event Bus so that the Dashboard can have the big picture of the delegation usage.

Delegation is also discussed in section 6 “The Delegation Service” of [TAS3D42Repo] and in section 6 of Deliverable D7.1 [TAS3D71IdMAnAz].

3.3.1 Invitation Based Token Approach

Assume Alice has used ePortfolio Front End to construct some artifacts about her career (see Fig-3.11), including attestation from University A that she has a degree, and some exhibits, in Album A of the work she has already done.

Now, Alice can invite her job coach Bob to access these artifacts as Web Services, see Fig-3.12. First Bob resolves the invitation to the necessary access tokens and then accesses the services. Of course Bob is action through the Job Search Front End.

On access (2a) two tokens will be present

**Target Identity** Delegation Token, encrypted for Album A and usable by Job Search, containing Alice’s pseudonym at Album A. This token indicates that the access is by invitation and not by Alice being present in the transaction. The Delegation token can also include description of which part of the user’s resource at the Service Provider can be accessed and how.

**Subject Identity** Token, encrypted for Album A and usable by Job Search, containing Bob’s pseudonym at Album A. This token indicates that Bob is performing the operation and that Bob is present in the transaction (at front channel).
3.3. DELEGATION

Figure 3.11: Alice Prepares ePortfolio.

Figure 3.12: Bob using Alice’s Web Services by way of invitation.
3.3.1.1 Details of the Invitation Flow

Consider Figs 3.13 and 3.14 and the depicted steps as follows:

1. Delegator (Bob) selects the resource to share at the Service Provider. Service Provider knows who Delegator is as Bob used Single Sign-On to login to the SP.

2. Once a resource has been selected, Delegator is transferred to the user interface of the Delegation Service. The Delegation Service generates an Invitation Ticket, which is formatted as URL pointing back to the Web GUI of the Delegation Service, but with a nonce component that is hard to guess. The invitation is remembered in the database of the Delegation Service.

Figure 3.13: Invitation phase
3. The Delegator then gives the invitation to Delegatee. The Delegator does not need to know the
specific identity of the Delegatee in the network. However, if the Delegator cares, he could use out
of band means of ascertaining that the Delegatee that receives the invitation is the person to whom
he wishes to delegate, e.g. instead of emailing the invitation, deliver it personally.

4. The Delegatee now resolves the invitation by clicking the URL and lands at the Web GUI of the
Delegation Service, which asks the Delegatee to identify itself by means of an IdP (usually Single
Sign-On). This IdP can be different from Delegator's IdP as long as the Delegation Service and
the SP are willing to trust it. Various mechanisms, that are described in Sections 3.7 and 3.6, to
establish the trust exist.

If Delegatee does not have any IdP, then Delegatee should register with some IdP, otherwise he
can not continue the flow.

N.B. A flow is possible where the invitation itself grants access to the resource without
any need to authenticate the delegatee. While this flow may be interesting in some com-
mmercial settings, it lacks sufficient audit and accountability safe guards to be included in
the TAS³ architecture.

5. Delegation Service invokes Delegatee’s Identity Mapping Service to convert the identity of the del-
egatee are the Delegation Service to the identity of the Delegatee at the SP and stores it in the
invitation database. The identity will be encrypted so that only SP can open it.

6. The Delegation Service generates an artifact with nonce properties and associates it with the in-
vitation. It then redirects the Delegatee to the user interface of the SP, passing the artifact in the
query string.

7. SP dereferences the artifact by contacting on back channel the Delegation Service, which looks up
the invitation using the artifact and generates a Delegation Token binding together the authorized
resource (known from time when invitation was created) and the Delegatee (known from step 5);
signs the token, and ships it to the SP.

8. The SP PEP now passes the invoking identity, i.e. the Delegatee, known from Single Sign-On
and the (contents of) Delegation Token to the PDP, which will authorize the operation. Typically
the authorization rule will require that the accessed resource and invocation identity match the
Delegation Token.

In this flow, the Delegation Service and the SP could be merged, effectively optimizing steps 2 and
7 to function calls. While such merge is technically sound, it may hinder development of competitive
market for the delegated services. The flows in [TAS3D42Repo] Appendix 2 “Proof of Ownership
Using Permanent IDs” essentially presents the invitation flow where the invitation is then embedded in
a composite document. That Appendix does not show how the Proof of Ownership URL (PoOURL) is
dereferenced. Essentially the steps 4-8 could be performed, or other means to authorize the access
could be used. It is important to understand that reliance on mere invitation does not leave audit trail
trace about who accessed. The delegatee really needs to be identified (e.g. by SSO to Delegation
Service) for the audit trail to be useful.

Essentially the steps 4-8, above, could be performed, or other means to authorize the access could
be used. It is important to understand that reliance on mere invitation tokens (secret URLs) does not
provide identity information to the audit trail about who accessed the document. But if authentication
and authorisation of the PII recipient took place because a fixed proof public URL was used, then full audit information is provided.
3.3. DELEGATION

3.3.1.2 Reuse of Invitation

One salient property of this flow is that the first time the invitation is dereferenced, it gets bound to a concrete user. In subsequent attempts to dereference the invitation a check can be made that it is the same user (but if requirement really is that the token should be reusable by different users, then this check can be omitted).

3.3.1.3 Application of Invitation Approach to Back Channel Web Service Calls

The invitations approach can be extended to cover back channel web service calls as well. This topic will be elaborated in the next version of this deliverable (D2.1 M30).

3.3.2 Mandate Tokens Approach

This approach is described in [Peeters09].

Next version of this deliverable (D2.1 M30) will outline application of Mandate Tokens to TAS³ architecture, including

- Emission
- Push Model
- Pull Model

3.3.3 Delegation by Direct Authorization Rule

In this model the resource owner, knowing delegatee’s full identification, creates an access control rule at the resource, i.e. sticky policy, that simply authorizes the delegatee to access the resource.

The ability to assign access to other user can be regulated by policies that are checked by the sticky policy interface, e.g. by consulting a PDP.

When delegatee accesses the resource, he identifies himself using the usual token passing flow, see Section 3.2, and the PDP is able to match his identity to the sticky policy authorizing the access.

Difficulties are

1. The Delegator has to have a solid identifier for the delegatee. This may be difficult for a human to know and accurately enter. It is possible to offer to the delegator a browsing or search interface of all potential delegatees, but such list may be unacceptable from data protection perspective and can be difficult to implement in a distributed way.

A better alternative may be to adopt the invitation steps 1-4 from section 3.3.1 and modify the step 4 so that it edits the access control rule.

2. The resource has to have a sticky policy editing interface and access to this interface needs to be well controlled. Online editing of policies increases exposure and potential for compromise.

3. If policy editing interface is centralized (e.g. in Dashboard), the identification of the resource to which the policy applies may be difficult. This can be solved to an extent by Discovery. The the policy editing interface of the resource would have to be web service based, with proof of presence of the delegator.
4. Privacy is poor as the requirement to know delegatee’s identity widely excludes pseudonymous approaches.

5. Invitations are not supported

### 3.3.4 Delegation by Role Based Authorization Rule

In this model the resource owner creates an access control rule at the resource, i.e. sticky policy, that authorizes anyone having a given role to access the resource.

The ability to assign access to a role can be regulated by policies that are checked by the sticky policy interface, e.g. by consulting a PDP.

The assignment of a role to a delegatee is performed by some role authority outside control (but still accountable through TAS\(^3\) oversight) of the resource owner. The usually role assignment is performed using a special Sharing GUI that the role authority accesses (his authoritativeness is checked by separate access control policy verifying that he is a role authority). The role authority needs to identify the delegatee and then assign the role.

The role is stored in delegatee’s role repository, which can be his Attribute Provider.

When delegatee accesses the resource, he identifies himself using the usual token passing flow, see Section 3.2, and the PDP is able to retrieve his role and match that to the sticky policy authorizing the access. The role retrieval may be through push or pull model.

In its basic form the RBAC allows any role holder access to any resource that accepts the role and the identity of the role holder is irrelevant, thus permitting use of pseudonyms or transient identifiers that improve the privacy.

However the fact that the resource is not constrained is a problem. This could be solved by having as many roles as there are resources, but this would lead to combinatorial explosion in the number of roles and quickly become unworkable. Also, the role identifier itself could become a correlation handle across the resources of a user.

A way to constrain the role is to parametrize it. Parametrized roles have a main part that specifies the role proper and then a parameter that specifies to which resource the role applies. Since parametrized role identifier has unique identifier properties, it is highly liable to become a correlation handle.

This approach adds flexibility, but still suffers from need to have exposed policy editing interface and to some degree about the problem of identifying the delegatee and resource. In parametrized variant any privacy protection is quickly lost.

### 3.3.5 Token Based Delegation to Well Known Delegatee

If Delegatee’s accurate identification can be known, the delegator can instruct a Delegation Service to create a signed (by Delegation Service) token for accessing his resource. The fact that he can make this delegation is checked against issuing policies, such as "data owner can delegate access to it".

The token can then be stored for future use in several places:

1. In Attribute Authority of the Resource Owner
2. In Attribute Provider of the Delegatee
3. In token store of the Delegation Service
When Delegatee accesses the resource, he can push the token, obtained from (2) or (3), to the web service that provides the resource. The token will identify the resource to which it applies, this is the Target Identity.

Sometimes the mere presence of the delegation token is sufficient for accessing the resource, this would be considered a bearer token.

However, in a more common case, the Delegatee also pushes a token identifying himself, e.g. a SSO token. This expresses the Subject Identity, i.e. the identity of the user performing the access. This allows the delegation token to be constrained to a user who can present token evidence of actually being the Delegatee. This is essentially a Holder-of-Key proof and produces strong audit trail that identifies who delegated and to whom.

Another variant is for the resource providing the web service to pull the delegation token from one of the sources. If Subject Identity is not known, but the accessed resource is known, then (1) can be used. In effect the discover of the Attribute Authority is keyed on the identity of the resource owner. Token can also be retrieved from (3). Finally, if the Subject Identity is known, pulling the token from (2) becomes possible.

### 3.3.6 Token Based Delegation of a Received Role

If a user has a role, given to it by some role authority, and policy permits delegation of this role, the user can go to the Sharing GUI and ask a delegation token to be issued with known delegatee and known target resource. Accurately identifying the delegatee and the target are serious problems as described above.

The ability to delegate only "received role" is expressed by Delegation Service having a policy which states that delegator can only delegate roles he has himself, as determined by the roles the authority has assigned to the user.

The issued token is then stored for use by method (2) or (3) as discussed earlier. The token can be used for access either by push or pull methods.

### 3.3.7 Multi-layer (Chained) Delegation

The token based delegation methods described above lend themselves to multi-layer delegation. In this case the delegation is with right to sub-delegate and the delegatee is able to request that further delegation tokens are created, or that an invitation is created.

For access authorization generally only the last step of a delegation chain is important, but for audit purposes the full chain is needed.

The flows and tokens associated with multi-layer delegation are a topic of research in the TAS³ project and will be further elaborated in a future version of this deliverable (D2.1 M30).

### 3.4 Subject of the PII Not Present -Transaction

There are two main categories for supporting PII-Subject-Not-Present transactions

1. PII Subject consented at some earlier time.

   If the consent was expressed by the PII Subject by creating a policy that authorizes the delegation, then we need to establish from the audit trail that only the user could have created the policy.
3.5. BREAK-THE-GLASS AUTHORIZATION

If token based delegation is used, PII-Subject-Not-Present could be implemented by “cacheing” an access credential for long time. Due to obvious problems with long term valid access credential, we only allow caching the discovery bootstrap credential. Using this credential the WSC MUST request a fresh access credential for the PII-Subject-Not-Present transaction immediately before attempting the transaction. This ensures that the PII Subject has control and visibility since the Discovery will be a third party to the transaction, allowing additional audit correlation. Discovery can also be used as centralized revocation point for the consent to the off-line transaction up to the point when the transaction is actually made.

The discovery token was originally issued for a purpose and when a transaction token is requested, including the PII-Subject-Not-Present situation, a purpose MUST be declared so that the authorization process at the discovery can make appropriate decisions.

2. Transaction is permitted by law or contract without consent of the PII Subject. In this case the big problem is that as the PII Subject might never have been present we need to consider how the Legitimate Initiator (LI) identifies the PII Subject in the first place.

a. The PII Subject has a federated account at the Legitimate Initiator. In this case the LI can ask the IdP to issue a discovery token. Such issuance needs to be strictly controlled. The LI is first authenticated (e.g. using PKI at TLS level) and then LI presents the legitimate purpose why the token is sought. The IdP will consult a PDP which will make the policy decision whether under these circumstances the discovery token should be issued. The audit trail shall rigorously reflect these events. After the discovery token has been obtained, the rest of the steps are as in (1).

b. The PII Subject has an account at the Legitimate Initiator, but it is not federated. The issuance step of (a) will have additional request to create the federation. If the PII Subject eventually ends up using the LI through SSO, the already created federation will be used.

c. The PII Subject does not have an account anywhere. In this case a stand-in account needs to be created first and then the process of (b) and (a) will be followed. If the PII Subject eventually starts using the LI using SSO, a problem remains on how to associate the PII Subject to the already existing account. This may be possible by asking some identifying information, such as sector specific or national identifier upon first SSO use. If asking such information is infeasible, or the PII Subject can supply wrong information, we may well end up with user having two accounts at the LI. Depending on the circumstances, this may not be a problem, or an administrative procedure may be needed to combine the accounts.

3.5 Break-the-Glass Authorization

This section addresses Reqs. D1.2-3.9-BPRecover, D1.2-4.6-BrkGlass, and D1.2-7.22-BrkGlass. See [TAS3D71IdMAAz] for further discussion of the Break-the-Glass authorization.

In a Break-the-Glass scenario an operation would not normally be authorized given the actor and the resource (including owner of the resource), but given justified and legitimate imperative need, the operation is authorized, usually with additional auditing enabled.

A classical example would be an unconscious patient brought to an emergency ward. The physician has imperative need to access the patient records, yet the patient can not consent to the access.

The Break-the-Glass authorization is an area of active current (2009) research and TAS³ architecture will in its future versions incorporate the best innovations in this area. Our current approach is as follows.
3.6 TRUST AND PRIVACY NEGOTIATION

PEP driven Break-the-Glass (RECOMMENDED approach): The PEP attempts authorization which fails for normal access. At this point PEP determines if Break-the-Glass could be applicable (could be indicated by PDP or structurally known by the PEP). If so, the PEP may proceed to ask consent from the actor ("Do you want to invoke Break-the-Glass privileges and additional auditing?") using interaction facilities of the architecture, or in some cases may enable the Break-the-Glass automatically.

Enabling Break-the-Glass forces additional audit messages to be generated at the PEP and by the service overall. It is also possible that the Break-the-Glass authorization is explicitly asked from the PDP in which case there will be audit entry also on the PDP side, permitting better cross correlation of the audit trails.

Policy editing driven Break-the-Glass: In this approach, after initial denial of the authorization, some mechanism is invoked to modify the policy such that the same actor will succeed in obtaining authorization. This approach is NOT RECOMMENDED due to complex security implications of allowing policy editing.

Role driven Break-the-Glass: This is similar to the Policy Editing approach, but the edit happens in the role authority. This approach is NOT RECOMMENDED for much the same reasons as Policy Editing: the role authority needs to be strictly controlled and introducing automated role editing mechanism is not desirable.

Delegation driven Break-the-Glass: In this approach after the failure, the actor visits the delegation authority and claims Break-the-Glass situation. The delegation authority verifies according to its policies if the Break-the-Glass delegation is appropriate given the actors role, etc., the resource, and the context of the request. If acceptable, the Delegation Authority issues a delegation token with special field that indicates the Break-the-Glass nature of the delegation and records the fact to its audit trail. When the delegation token is wielded at the PEP, it recognizes the delegation authority as trustworthy and notices the Break-the-Glass indication, enabling more extensive logging. This approach can be made to work.

External Break-the-Glass IdP driven approach: This approach is a front channel special case of the PEP driven approach with some features of the Delegation driven approach. The particularity is that the PEP redirects the user, with indication of the resource to be accessed, to a special IdP which is responsible for verifying (probably querying a PDP) whether the user is eligible for Break-the-Glass access. If so, it issues a token indicating that the user is eligible and has been granted access, what resource is to be accessed, and the fact that Break-the-Glass has been invoked. The PEP seeing this token grants access, but enables additional logging. The special IdP will also have audit trail of the events, so it will be possible to cross check the trails.

3.6 Trust and Privacy Negotiation

This section addresses Req. D1.2-7.17-Inc Rem.

Trust and Privacy Negotiation logistics and flows are subject to TAS³ research. A future version of this deliverable (D2.1 M30) will report on these mechanisms in detail.

To give rough overview, the Trust and Privacy Negotiation can be carried at two levels: on Front Channel a user interface (Web GUI of a Front End) will step user through sequence of mutually revealing more information until agreement can be reached and transaction can be completed. This may involve the user agreeing to relax policies on his data, or the Service Provider agreeing to honour some additional user policy that it did not offer originally. The second level is Trust and Privacy Negotiation in the back channel. The Discovery Service plays a large role in the Trust and Privacy Negotiation at
3.7. INTEROPERATION ACROSS TRUST NETWORKS

3.7 Interoperation Across Trust Networks

General approach for interoperation across Trust Networks is described in [LibertyInterFed], which focuses on the token passing flows in inter-federation situation. In addition to token passing, interoperability at data level is needed, i.e. the ontologies in use in the different Trust Networks either need to be the same or they need to be mapped. In particular, authorization critical data needs to be mapped.

Next version of this deliverable (D2.1 M30) will provide specifics of interoperation, based on TAS³ research and implementation experience.

3.7.1 Semantic Interoperability Engine

This section satisfied Reqs. D1.2-2.23-SemIOP, D1.2-3.14-PlIPolicyDisco, and D1.2-3.15-SecPreserve.

A semantic interoperability provides different mechanisms to map ontological entities from heterogeneous entities. Castano et al. [Castano07] identify two main categories of ontology matching techniques; namely linguistic and contextual matching techniques. Linguistic techniques evaluate the similarity among ontological content (i.e. classes, roles and instances) based on their names or labels. The main characteristic of these techniques is that they evaluate the similarity between two strings of characters. For example, the edit distance counts the minimum number of changes, such as insertion, deletion and replacement of characters, required to transform one string into the other string [Levenshtein66]. Although linguistic techniques often provide highly precise mappings, these techniques tend to fail when there is little lexical overlap between the labels of ontological entities. Context-
3.8 Properties of Web Service Binding

Web Service Binding is a set of features that the communications layer is assumed to have. These features are often required by more sophisticated protection mechanisms like the token passing flows. They often address basic and well-known threats like replay, unauthorized, and man-in-the-middle attacks in a basic way while other mechanisms may address the same topics comprehensively, but in a more expensive way. Many of these features may seem self-evident, but we need to list them even if just to state the obvious.

1. Mutual authentication of the communicating entities MUST be possible. Usually this is done using transport layer digital certificates, but other approaches are possible.

2. Link confidentiality MUST be possible, usually using transport layer encryption.

3. Correlation
   - Request-Response Correlation
   - Business Process identification in correlation

4. Redirection support for flexibility

5. Recredentialing support (Req. D1.2-3.9-BPRecover)

6. Asynchronous support SHOULD be implemented (this will be addressed in a future version of this document)

7. Interaction Callback (or Exception Request)
   - Interaction Redirect (Req. D1.2-3.9-BPRecover)
3.8. PROPERTIES OF WEB SERVICE BINDING

- Interaction Service (Req. \textit{D1.2-3.9-BPRecover})

8. Digital signing of messages for nonrepudiation (Reqs. \textit{D1.2-2.11-Transp, D1.2-2.15-Resp, D1.2-4.4-CourtProof})
4 Application Specific Architecture

4.1 Protocol Support for Conveyance of Sticky Policies

Most of the protocol flows of TAS³ use industry standard Web Services bindings and Web Services payload protocols. It is an explicit design goal that existing services are enabled with minor disruption.

A pertinent problem with existing payload service protocols is how to express the sticky policies that generally have to be bound to the data with a digital signature. Following approaches have been identified:

1. Treat all data in one request-response pair as having the same Sticky Policies. In this cases relatively nonintrusive methods like SOAP headers and LDAP controls can be used to indicate the sticky policies. We call this Security Header (SH) approach. This approach is already available as UsageDirective SOAP header defined in [IDWSF08].

2. Use the extension points of the payload protocol to express the Sticky Policies. We call this approach Application Protocol Enhancement (APE), see [TAS3D71IdManAz] section 8.2. This approach gives granular Sticky Policies that are naturally associated with the data and does not alter top levels of protocol processing. If client and server are updated to understand this scheme then it works well. Eventually new payload protocols should be specified with TAS³ APE feature built in. A danger of this approach is that if the client is not updated, it may just silently ignore the Sticky Policies.

3. Expand the data model to carry sticky policies. This is really a special case of APE with similar merits and problems. One benefit is that it is sometimes easier to extend a datamodel than a protocol.

4. Encapsulating Security Layer (ESL), see [TAS3D71IdManAz] section 8.2. Wrap the payload protocol in a TAS³ defined encapsulating protocol that contains all the TAS³ specifics and in particular the sticky policies. Advantages of this approach include:

   • The encapsulated protocol does not need to be modified at all
   • Possibility to add sticky policies to protocols that do not offer extension points and that are not under control of the implementer.

Disadvantages of this approach are:

   • More invasive on outer layers of the protocol stack. This may make it difficult to integrate to existing protocol stack.
   • If the payload protocol is not SOAP, or otherwise has poor impedance match to the TAS³ ESL protocol, then integration may be impossible.
   • Association of the sticky policies to the data will require awkward correlation of data items to the policies. In particular, if the data does not have item specific IDs, it may be necessary to resort to use of techniques such as [XPATH99].
4.2 Legacy Integration Strategy

For TAS\(^3\) architecture to be useful, it needs to be adopted. To adopt TAS\(^3\) an existing application faces some implementation choices.

Conceptually simplest, but in terms of new code to write probably most costly approach is shown in Fig-4.1: just build a TAS\(^3\) compliant PEP into the legacy application. This approach has the advantage of allowing full control over the enforcement process, including the inputs to the Master PDP. The disadvantage is the learning curve to learn TAS\(^3\) architecture in sufficient detail to implement it correctly and to get certified.

Fig-4.2 depicts a slightly different strategy where application only implements a simple PEP, which then communicates with the Application Independent PEP (AIPEP) supplied by the TAS\(^3\) Project. In this approach, the AIPEP component handles most of the TAS\(^3\) specific parts and can be an already certified component, making compliance certification easier.

However, in this model the need to communicate between AIPEP and ADPEP arises. The communication pattern could be either Encapsulating Security Layer (ESL) or Application Protocol Enhancement (APE), as described in the Section ??
4.3. ADPEP

Figure 4.2: Application Integration: ADPEP implemented in application itself.

Fig. 4.3 illustrates some specific integration strategies with the intent of enabling legacy data sources that can not be modified. In (A) the SOA Gateway evolves to support TAS³ architecture, in (B) the SOA GW is frontended by the WP9 database which supports the TAS³ architecture aspects. If export of the legacy data is an option, then it may be simplest to import the data to WP9 database and dispense with the legacy data source entirely (C).

Figure 4.3: Application Integration using ADPEP and (A) SOA Gateway, (B) WP9 DB as frontend to SOA GW, (C) WP9 database.
4.3 ADPEP

The Application Dependent Policy Enforcement Point (ADPEP) is a gateway that provides access to the TAS³ infrastructure for applications like web applications with a web frontend, business process engines, databases or repositories and many other systems, which are either requesting or responding over a TAS³ secured and trusted channel. Per section 2.2.1 (see also Fig-2.2), the ADPEP belongs to the Front End Services and Web Service components inside the Payload boundary.

As described in Section 2.2.2, the ADPEP is divided into two different types of Application Dependent Policy Enforcement Points:

1. ServiceRequester ADPEP: This web service is part of the Front End Services. Internally, the ServiceRequester ADPEP constitutes together with the Stack, the Service Requester. The Stack handles SOAP protocol details. The Application Independent Policy Enforcement Point (AIPEP) contacts Master PDP, which contacts different PDPs like User PDP, Organization PDP or a Trust PDP to decide whether a request is trusted or not.

   The main task of the ServiceRequester ADPEP is to collect all required information for an appropriate request that has to be checked by the TAS³ authorization infrastructure. Further information about the payload, which builds up the request, can be found in [TAS³D81RepoSW] figure 8. Common information about the functionalities of the ServiceRequester ADPEP can be found in [TAS³D81RepoSW] and in [TAS³D83CliSW].

   The next steps before sending the request are done by the 'Stack'. As mentioned before, the 'Stack' (and its main component: the AIPEP) is application independent. Its main task is the preparation of the request. The message has to be signed and augmented according to web services binding. WP4, WP5 and especially WP7 work on this security related part of the service requester. Whereas WP8 is responsible for the application dependent part.

2. ServiceResponder ADPEP: This second application dependent service, which functions as responder, is part of the Service Responder component in the Web Service boundary (see Fig-2.3). In analogy to the ServiceRequester ADPEP, the ServiceResponder ADPEP also needs the 'Stack' (with AIPEP and its underlying PDPs) to function correctly. That means, signing and preparation of the message according to the web service binding, the policy checks and the communication with the 'Trust policy decision point', as done by the 'Stack components'.

   The main task of the ServiceResponder ADPEP is to receive requests, route them in an appropriate way to the 'Service Application' and then send back the response to the requester. More details about the functionalities of the ServiceResponder ADPEP can be found in [TAS³D81RepoSW] in chapter 3.2.

Auxiliary components in the both ADPEP Services

To fulfil the mentioned functions of the both ADPEP Services (Requester and Responder), some auxiliary services are required. These services belong to tasks (Task 8.3 - see DoW), which are documented in [TAS³D82BackOffice].

These services neither store person related data nor serve the user directly. They provide ontologies and metadata, perform search and aggregation operations and transform data into specific formats. The back office services are a component of the TAS³ Trusted Application Infrastructure but not of the core TAS³ Trust and Security Infrastructure.

The main Auxiliary or Back Office Services and Components are:
4.3. ADPEP

- The Generic Data Format ([TAS3D82BackOffice], section 2.1.2) used to store data in TAS³ repositories\(^1\)
- Services to transform ([TAS3D82BackOffice], section 2.1) data from a custom source format to the Generic Data Format and from the Generic Data Format to a format, which is requested (and supported).
- Aggregation Service ([TAS3D82BackOffice], section 2.2) and Policy Aggregation ([TAS3D82BackOffice], chapter 8)
- Request Logger Service ([TAS3D82BackOffice], section 3.2) to store information on requests issued and responses received by TAS³ web services for auditing and maintenance purposes

\(^1\)Marc: not applicable for eHealth because of legal issues.
5 Using Business Process Modelling to Configure the Components

This section addresses Reqs. D1.2-3.2-ModelDrivenCfg, D1.2-3.12-SPManifest, D1.2-6.3-WhatHowWhyWho, and D1.2-6.4-Min.

The TAS³ architecture covers a lot of functionality and some of this functionality needs to be configured carefully to match each other to ensure smooth operation from the perspective of the users, such smooth operation is perceived by users as dependability and trustworthiness, so it is a prerequisite for good public image of a Trust Network.

Correct configuration will also be essential for ensuring that services function securely. Given that most security technology is quite brittle and even minute misconfigurations lead to failure, there will be operational and commercial pressure to turn off those nonfunctional, but essential features that appear to be "causing trouble". This is an extremely dangerous slippery slope that any Trust Network MUST avoid. Liberty Alliance and SAML Interoperability and Certification programmes have clearly demonstrated this to be a real peril. Therefore it is necessary that it is possible to correctly configure the trust network such that it will work right on the first try.

Complexity of a typical Trust Network, along with all of its member systems, is of such a high degree that it is infeasible to configure it sufficiently correctly by a manual approach. Humans make mistakes. An automated, model driven configuration is the only way to create accurate and correct configuration.

The corner stone is Business Process Modelling. From this model, which exists both at the top Trust Network level and at the organizational level, it should be possible to derive the following outputs:

1. Circle of Trust parameters to facilitate federation and SSO configuration
   a. White list of roots of trust for both authentication and authorization
   b. Trusted Certificates for TLS [RFC3548] and Signing
   c. Metadata for entities

2. Declarative Statements about attribute needs of the Clients as well as policies under which providers are willing to release attributes. This output will come in the form of [CARML] and [AAPML] files, see further [IGF], that can be used to automatically configure IGF enabled layers of Client Request PEP and Provider Request PEP.

3. Some of the top level policies that apply to the Trust Network and its members. This should facilitate configuration of the PDPs.

4. Policies, Business Process Models, and Interface Descriptions (e.g. WSDL) that are needed as input for the Automated Compliance Validation.

5. Business Process Models that are needed as input for Business Process Visualization at the Dashboard.

6. Policy and business process model descriptions needed by the Configuration and IT infrastructure management, e.g. CMDB, ITIL, etc.
Contractual information behind a business process will influence the business process model itself and has an impact on the security rules, roles, and policies related to the business process.

Security rules that guide selection of web services and use of secure entities (data), i.e. influence the discovery service.

Security exceptions during business processes will raise an exception. Unhandled exceptions will block or break the process (go to an operator or help desk). The challenge is to handle the exceptions by explicit routines in the business process modelled routines or in some cases by using alternative paths (i.e. subprocesses) to allow the process to complete or even to look ahead and avoid exceptions before they occur by such means.

Business processes can have more complex topology than just trees.

There is no centralized log, just references to local logs are passed out.
6 Oversight and Monitoring

For a TAS³ compliant Trust Network to gain a trustworthy reputation and to ensure that belonging to the Trust Network really enables lower cost of operation through lesser fraud, improved trust, and ultimately less need for formal audits, it must take proactive and mandatory activities to monitor its activities and stop any fraudulent practices before they become a problem, ideally even before they become publicly known.

This section addresses Reqs. D1.2-2.11-Transp, D1.2-2.12-Compr, D1.2-2.15-Resp, D1.2-2.16-Mitigate, D1.2-2.17-AuditUntamp, D1.2-2.21-DataProtLaw, D1.2-2.22-GovtAccess, D1.2-12.13-Vfy, and D1.2-12.15-Valid.

In TAS³, the monitoring should happen at levels of

1. Continued automated, robotic, testing that compares results to both modelled expectations and past results. This is one of the focus areas of TAS³. See: On-line Compliance Testing (OCT).

2. Operations monitoring to determine upness and performance of services, as well as detection of anomalies. Trouble ticket system for reporting and rectification of operational errors, as well as intrusion detection scans and monitoring are included here as well. Use of industry standard solutions is recommended as TAS³ does not plan additional research in this area.

3. Log audit. Some part of log audit is handled in operations monitoring, above, but logs will contain a wealth of additional information, such as usage patterns to inform new investment and areas of innovation, which can be extracted using data mining techniques. Use of industry standard solutions is encouraged in general as the only connection with TAS³ research is in the area of gathering inputs for reputation scoring.

4. Formal compliance audits should occasionally be carried out manually to ensure that the automated monitoring and audit mechanisms, above, are functioning correctly. These audits may be mandated by legislation or by governance agreement and are typically fairly costly affairs with reputable outside consultants specializing in organizational and IT audits. TAS³ contribution for this area stems from recommendations and guidelines of the project legal team.

5. Administrative Oversight. The Trust Guarantor will take necessary administrative steps to ensure that the Trust Network is adequately monitored, mostly automatically, but with necessary and timely manual intervention. The Trust Guarantor may, according to the Governance Agreement, be monitored by an Advisory Board, Management Board, and ultimately General Assembly.

Section of 4.3.7 "Management" of [NexofRA09] discusses the need for management interfaces in services components. TAS³ is compatible with these requirements.

6.1 On-line Compliance Testing

This section addresses Reqs. D1.2-6.14-Compat, D1.2-6.15-MinPolicy, and D1.2-12.16-OnlineTst.

Implementation of SOA based applications result from the integration of several services. Services composing an application can change at run-time without informing all the other services integrated in the application. Furthermore, features like dynamic binding, or context-dependency prevent knowing before run-time the actual interaction among service instances.
Speaking in general terms, services are typically controlled and owned by different organizations. Thus, dealing with architectures that are not under the full control of one organization, means that the service lifecycle cannot be structured in well-defined development stages. In particular, for a (composite) service it is not clear when testing activities starts or should end.

To ensure trust and dependability the TAS³ architecture must also include adequate technology and actors to test that services within a TAS³ Trust Network behave in compliance with their expected specifications. Such testing activities must be performed on-line by special TAS³ guards, verifying that the services with a choreography actually behave as expected.

To achieve trustworthy SOA, there is the need to develop and use a methodology and tools supporting the “perpetual” (i.e. event-driven, periodically) and automatic testing of software services. The benefits with the “perpetual” and automatic testing we envision:

- repeatability of testing (improving the efficiency and the efficacy of the test)
- increase the quality and the trust perceived by the users of the service

The extent to which compliance is tested may vary, also depending on the registration information that should accompany services (e.g. models describing interfaces, policies, usage, etc.), which is part of the governance contract.

A minimal assumption is that services should access and perform within a TAS³ infrastructure according to an explicitly declared set of policies and that the infrastructure should not allow violation to the declared policy, or at least should recognize such violation. Testing is applied in order to reduce the risk that services within a TAS³ infrastructure will get in contacts with unreliable services. Therefore services within a TAS³ compliant infrastructure will be regularly submitted to testing session aiming to assess that a service does not break its policy.
6.1. ON-LINE COMPLIANCE TESTING

As important remark, we advise that this on-line testing approach does not prevent the execution of canonical off-line testing activities (e.g. where the service is tested by its developer trying to anticipate possible usage scenarios), rather it is an additional mean to increase the trustworthiness of the TAS³ architecture.

6.1.1 Involved Actors

On-line Compliance Testing impacts on the following of actors of the TAS³ scenario.

- Ecosystem
  - Service Provider
  - Reputation Providers
  - Software Certification agencies
  - Deployment certification and audit agencies
  - Compliance Authority

- Components
  - Web Service (WSP)
  - IdP (Identity Provider)
  - Service Registry
  - Id Mapper
  - Linking Service
  - Organizational PDPs
  - Trust Network PDP

6.1.2 On-line Testing Process and Architecture

The TAS³ architecture includes an on-line testing infrastructure, in which, the TAS3 services are tested according to a set of published models describing specific behavioural characteristics of the service itself. In the following, the term OCT refers both to the on-line compliance testing process and to the infrastructure that implements it.

With respect to the current scope of the TAS³ architecture, the compliance testing functionality requires that each service exposes within the TAS³ choreography its public interface (i.e. its exported operations) and the public policies it will comply with (or a references to them). In particular, each service is tested on-line when it requests to be registered to a TAS³ directory service. Further on-line test sessions can be activated by the compliance testing (i.e. Trust Network level) directory service either in event-driven or periodic fashion.

Directory Services within a TAS³ architecture interact with testing components to detect services failures. The compliance testing increases trust both on the TAS³ architecture and the linked services. A software service willing to be registered to a TAS³ directory service, may also have to comply with other policies that are not publicly manifested. In this case the service will not be tested with respect to such policies since such behaviour is hidden from the external interfaces of the service.
During the on-line compliance testing process, references to the service should not be retrievable by means of the directory service. At the end of the compliance verification process, if and only if all the tests have been successfully executed, service references will be listed by the directory service.

During the on-line testing, the OCT activates tester robots. Each tester invokes the service under test simulating service requests with identity credentials taken from a pre-packed identity test suite. The tester robot collects the service reply (i.e. either a response message, or a deny access), and compares it with the expected results: a difference between the service reply and the expected result reveals a mismatch between how the service policy is manifested and how the policy is implemented within the service.

Note that the TAS$^3$ architecture has a precise requirement that error messages returned after a request for a resource (e.g. "access denied" message) must be identifiable as such. Applications might masquerade error messages for user-friendliness (e.g. they could produce a "pretty formatted" page); nonetheless, the TAS$^3$ architecture needs to be able to unambiguously recognize error messages without the need to delve into the semantics of the payload of the message. This is accomplished by each application declaring to the on-line compliance testing infrastructure at least one successful and one failing test case with exact description of what the messages look like and what are the relevant parts.

In real life scenarios, a service under test may need to access external services when invoked by the tester robot. Indeed, in some cases a testing interaction between the service under test and externally invoked service may have permanent effects (e.g. on a stateful resource). Let’s consider that the service under test queries the directory service to lookup a relevant end point. In this case, OCT should consider that the registry may return a reference to a Proxy version of the required service: this service will implement the same interface as the required service. Doing so, the real implementation of registered services is hidden to those services waiting for compliance validation - a useful feature while project is ongoing and full service is yet unavailable.

In such cases, the directory service and OCT have to be able to link to an existing service proxy or to generate new ones. Obviously this will increase the complexity of the framework and asks for the provisioning of service description models suitable for automatic generation of service stubs.

Fig-6.2 depicts a UML Diagram describing the components of the On-line Compliance Testing framework. In the following we list a detailed descriptions of each component:

- Compliance Validator Discovery Service: according to the architecture given in Fig-2.2, this component enhances the functionality provided by the Discovery Service. In particular, the Compliance Validator Discovery Service is a Discovery Service able to apply the compliance validation at runtime. The Compliance Validator Discovery Service aggregates three sub-components: the Compliance Validator, the Proxy Factory and the Pending Services DB.

- Compliance Validator: this component activates the testing session when a service makes a request for being included in a TAS$^3$ infrastructure. The testing session will result in a sequence of invocations to the services requesting to be registered. In case the testing session does not highlight any error the service will be registered otherwise the request will be rejected.

- Tester Robot: this is the component that actually runs the test on a given service. In particular, the Compliance Validator activates the Tester Robot passing to it a reference to the service that needs to be tested and to the corresponding test suites to use.

- Request Generator: this component defines which is the next invocation to make to the service under test in order to assess its correctness;
6.1. ON-LINE COMPLIANCE TESTING

- Test Checker: this is the component that checks that the replies of the service under test actually conform to what is specified;

- Test Attribute Generator: this component defines which attribute values are to be used in the testing invocations.

- Front Channel Tester: this component extends the Tester Robot adding to the tester component specific features to interact with the front channel of a service application in TAS\(^3\). Since this component is a Tester Robot, it has all the features that a Tester Robot has, including the relationship with the other components (e.g. Request Generator, Test Checker, Test Attribute Generator).

- Back Channel Tester: this component extends the Tester Robot adding to the tester component specific features to interact with the back channel of a service application in TAS\(^3\). Since this component is a Tester Robot, it has all the features that a Tester Robot has, including the relationship with the other components (e.g. Request Generator, Test Checker, Test Attribute Generator).

- DB of Roles as Signed Test Response: this DB contains the identity test suite that the tester can use to test other services simulating a different identity.

- DB Test Report: in this DB the compliance validator logs the result of a testing session and the possible failures highlighted.

- Proxy Factory: this component automatically generates proxy services to simulate already registered services.

- Pending Services DB: this DB will contain the identities of services that requested to enter a TAS\(^3\) infrastructure precinct but still did not pass the testing session. Services in such DB are not returned as result of a discovery request. Identities are removed from this DB when the corresponding testing session is terminated.

Note that, each entity (i.e. components or artifacts) in the diagram, has a role with respect to the domain organization. Specifically, according to the general architecture depicted in Fig-2.2:

- The artifacts Public Interface, and Public Policy, are entities within the Modelling & Configuration Management area,

- The OCT components (Compliance Validator Discovery Service, Compliance Validator, Tester Robot, Request Generator, Test Checker, Test Attribute Generator, Front Channel Tester, Back Channel Tester, DB of Roles, DB Test Report, Proxy Factory, and Pending Services DB) are entities within the Audit & Monitor area,

- The components IDP, Discovery Service, and Web Service, are entities within the Runtime & Enforcement area.
6.1. ON-LINE COMPLIANCE TESTING

For example according to the SP-Initiated SSO with Redirect and POST Bindings schema at Fig.12 of the SAML Specification.
6.2 Operations Monitoring and Intrusion Detection

[NexofRA09], section 4.3.7 "Management", paragraph 4, highlights the need for operational monitoring. While such monitoring is not a requirement for technical interoperability of TAS\(^3\) framework, it will be necessary to maintain reputation of TAS\(^3\) Service Provider and/or Trust Guarantor. This topic is not an area for TAS\(^3\) research work. Consequently:

1. Standard operations monitoring approaches such as SNMP [RFC1157] and Nagios [Nagios] SHOULD be implemented.

2. Each organization in the Trust Network MUST be protected by network level firewall or packet filter. Any deny\(^1\) events from the firewall SHOULD be fed to the Intrusion Detection Channel of the Audit Event Bus.

3. Each organization in the Trust Network SHOULD operate an Intrusion Detection System (IDS) to
   a. Detect well known attacks (e.g. ping of death)
   b. Port scanning
   c. Abusive patterns of usage

   Any suspicious\(^2\) events from the IDS SHOULD be fed to the Intrusion Detection Channel of the Audit Event Bus.

6.3 Log Audit

This section addresses Reqs. D1.2-2.17-AuditUntamp, D1.2-3.3-Dash, D1.2-6.10-Redress, and D1.2-7.21-Safe.

Log audit has several goals

1. In case of attempted repudiation, prove that events happened

2. For investigation, browse and visualize events so that a human investigator can get a relevant and sufficient overview.

3. On an ongoing basis automatically detect noncompliant events

4. On an ongoing basis ensure that the systems are functioning correctly

5. Provide statistics about users and their behaviour

6. Provide statistics about system use and behaviour

7. Provide baseline of information that allows trust, security, and access control mechanisms to be cross checked

Log audit raises several issues

A. Collection
The log audit could also be used for billing in some circumstances, but in general we recommend that billing systems be built separately so that they can be cross checked against the logging to detect errors.

A taxonomy of audit events is presented in Annex G "Enumeration of Audit Events". Some design requirements for Audit Log Files are discussed in [TAS3D2 Repo]. Further requirements include:

I. Append mode of access: Only append mode of access should be allowed, so that users or applications cannot rewind an audit log file and delete or modify information that has already been stored there.

II. Authorised writing: Only authorised parties should be able to append log records to the audit trail. Though unauthorised applications or attackers may gain access to the audit trail and try to append fake log records to the audit trail, or modify or remove the audit trail, this should be detected by the tamper detection mechanism.

III. Timestamps: Every record in the audit trail should be timestamped to provide a trusted record of when the audit data was received. We note that if the audit service is trusted to record the audit data without tampering with it, then it should also be trusted to append the correct time to the data. Therefore we do not require a secure time stamping service.

IV. Secure communication: if the audit service operates as a web service then there should be secure communications between the clients and the server in order to ensure tamper resistance, data integrity and authorised connection.

V. Secure storage on untrusted media: Since an audit trail may be viewed on untrusted machines, the security mechanisms should ensure persistent and resilient storage of the audit trail, and ensure detection of tampering of the audit trail - modification, deletion, insertion, truncation, or replacement. If tampering is detected, the audit service should be able to notify the security auditor.

VI. Support multiple simultaneous clients: The audit service should be easily and conveniently accessible and it should be able to serve multiple client applications simultaneously.

VII. Logging efficiency: The computational work and the storage size required by the audit service should be as efficient as possible.

VIII. Contents transparency: the audit service should be able to record any digital content coming from any repository service.
IX. Authorised reading: Since the audit trail may contain personal or sensitive information, then the audit service should ensure that only authorised applications or people have the privilege to read the audit trail. The audit trail may be encrypted to further protect confidentiality.

6.3.1 Log Collection and Storage

This section addresses Req. D1.2.2.22-GovtAccess.

In the TAS³ architecture the audit trail is collected and stored locally primarily at the system entities, such as SPs, IdPs, the IM, and the like or near them in the organizations that operate these entities. Everyone that collects a log is bound by a Governance Agreement so that responsible behaviour can be enforced when technical solutions fall short in some area of protection.

The log events originate in various components at various times, see Annex G “Enumeration of Audit Events” for an idea of the types of events that will be generated. For example, Web Services Stack component will check signatures on the tokens (assertions) that are presented and log both positive and negative outcomes.

The system entities that collect the audit trail or the centralized audit function of the organization report the events in summary form, essentially just pointers to the actual audit records, to the Audit Event Bus. Each component may keep its local log in its own format (in future we may provide standard format), but the summary logging to the Audit Event Bus will follow TAS³ standard format (this format will be presented in a future version of this architecture). To facilitate standard format summary logging, TAS³ may provide a reusable software library.

The Audit Event Bus is divided in channels to which different events are broadcast. This allows minimal exposure as subscriptions can be on the basis of only relevant events. The subscriptions can also be controlled such that only authorized parties with "need to know" can see certain types of events (see req IX above).

The Audit Event Bus is potentially implemented as part of a more generic Event Bus infrastructure, but due to special privacy and security requirements, Audit Events MUST NOT be mixed with other business messages, unless in encrypted form. If the generic event bus supports an encrypted private channel, a VPN if you like, then sharing of the infrastructure may be possible.

The Audit Bus infrastructure MUST be free of conflicts of interest. In particular, it should not be operated by one of the SPs. In case the Event Bus sharing is implemented, then the operator of the shared infrastructure MUST be free of conflict of interest as well.

6.3.2 Privacy Issues: What to Collect and What to Report

This section to be fleshed out in project month M30 release of D2.1. It will satisfy Req. D1.2.4.2-BPPrivacy.

The main issues are

1. Avoid logging anything that could become a correlation handle
2. Avoid logging PII unless absolutely necessary

Generally a lot of detail will be logged locally. This will include the tokens used in identification the user, usually in pseudonymous form as well as the PII handled by the Service Provider. This detail tends to be necessary to legally protect the Service Provider.
6.4 Formal Compliance Audits

This section will be developed in the D2.1 of M30.

- Legal compliance
- Penetration testing
- Disaster recovery exercises

6.5 Administrative Oversight

This section partially addresses Req. *D1.2-6.10-Redress.*

Administrative oversight and stake holder issues are covered in [TAS3BIZ], reproduced in Annex E "Business Model", .

This section will be developed in the D2.1 of M30.
7 Conclusion: TAS$^3$ is Secure and Trustworthy

Comprehensive approach of the TAS$^3$ architecture and framework achieves real and tangible overall security and trustworthiness gains when compared with state of the art for multiplayer networks of comparable size. TAS$^3$ features that contribute to this are:

1. Legal concerns are built-in from the ground up
2. A comprehensive and strong digitally signed audit trail
3. A conditionally pseudonymous audit trail to guarantee the privacy of Users who play by the rules, while allowing abuse to be exposed through collaboration of Service Providers.
4. A fully pseudonymous design at all layers to protect user privacy
5. Fully encrypted and digitally signed messages using strong algorithms
6. Based on state-of-the-art Single Sign-On protocol standard (SAML 2.0) which has had extensive security review
   - Extensive security review and scrutiny already done
   - Multiple commercial and open source implementations that are mature.
   - Certification program for implementations further ensures quality
7. Based on state-of-the-art Identity Web Service Protocol standards (ID-WSF 2.0) which have had extensive security review
   - Extensive security review and scrutiny already done
   - Multiple commercial and open source implementations
   - Certification program for implementations further ensures quality
8. Enhanced authorization infrastructure which significantly improves upon the current XACMLv2 standard
   - Extensive security review and scrutiny already done
   - Multiple commercial and open source implementations
9. Ability to use risk control and reputation
10. Use of ontologies to ensure consistent interpretation of data and authorization rules
11. On-line Compliance Testing for early detection of discrepancies and problems
12. Business Process Modelling driven configuration to ensure consistently correct configuration
13. TAS$^3$ has performed a systematic threat analysis (see Annex F) to ensure that the architecture addresses the widest possible range of security and privacy threats.
14. Software engineering techniques used by the project to consistently achieve high quality and absence of security bugs in the software components that are TAS$^3$ deliverables.
TAS³ Architecture is novel as a blueprint that brings together identity management, attribute based access control, business process modelling, and dynamic trust. The architecture, with Annex A, acts as an interoperability profile for various standards based protocols covering these areas. Other areas of innovation are user transparency features like Dashboard, user accessible audit trail, and automated compliance validation; privacy protection using sticky policies; marriage of trust and privacy Negotiation with discovery and trust scoring; secure dynamic business processes; and built-in first class support for delegation.
Annex A: Protocols and Concrete Architecture

This section describes the TAS³ Concrete Architecture and protocol choices in a normative and prescriptive way. Please note that the high level architecture described in the preceding sections can be implemented using other protocols than the ones chosen here. To promote interoperability we make fairly specific choices. If you choose to use other choices, you MUST NOT call your system TAS³ compliant and you SHOULD write a profile that describes your choices.

To complement the specification of protocols here, the reader may want to consult Fig-8.18 in [HafnerBreu09] for an overview of the functionality available in various specifications.

The choice of protocols has been guided by commitment to open standards as recommended in section 2 of [UNDP07]. This also serves to address Reqs. D1.2-2.4-MultiVendor, D1.2-2.5-Platform, and D1.2-2.6-Lang.

A.1 Supported Authentication and Login Systems

This section addresses Reqs. D1.2-2.18-AnCredi, D1.2-6.12-Sec, D1.2-7.3-An, and, D1.2-7.10-Target.

A.1.1 System Entity Authentication

TAS³ adopts X.509v3 public key certificates as primary means of authenticating system entities. This will apply over TLS and ClientTLS connections and may also apply in digital signatures.

For bilateral authentication Client TLS MUST be supported. HTTP Basic authentication MAY be supported.

A.1.2 SAML

Given the already broad adoption of SAML 2.0 by the eGovernment and academic communities across the world (e.g. DK, NZ, etc.), this choice is effectively already made for us. By choosing SAML 2.0 we enable many existing eGovernment and academic projects easily to become TAS³ compliant in future.

1. TAS³ adopts SAML 2.0 Assertions, see [SAML2core], as primary and recommended token format
2. TAS³ adopts SAML 2.0 as primary and recommended SSO system, see [SAML2core]. (Req. D1.2-3.10-JITPerm)
3. TAS³ RECOMMENDS that SAML 2.0 implementations are Liberty Alliance Certified.
4. SAML 1.0, 1.1 [SAML11core], 1.2, as well as Liberty ID-FF 1.2 [IDFF12] MAY be supported
5. Redirect - POST SSO profile MUST be supported by all front channel participants, see [SAML2prof] and [SAML2bind].
6. Redirect - Artifact - SOAP SSO profile MUST be supported in IdP and SHOULD be supported in Front End (SP), see [SAML2prof] and [SAML2bind].
7. Redirect Single Logout Profile MUST be supported, see [SAML2prof] and [SAML2bind].
A.1. SUPPORTED AUTHENTICATION AND LOGIN SYSTEMS

8. IdP Extended Profile, see [SAML2conf], namely IdP Proxying, MUST be supported

9. Other SAML profiles MAY be supported

10. SAML metadata MUST be supported, see [SAML2meta]

11. Well Known Location (WKL) method of metadata exchange MUST be supported, see [SAML2meta]

section 4.1 “Publication and Resolution via Well-Known Location”, p.29, for normative description
of this method.

12. In redirect binding [RFC1951] deflate compression MUST be used. [RFC1952] format MUST NOT
be used.

A.1.2.1 Authentication Request

1. MUST use NameIDPolicy/Format of Persistent when implementing Pull Model (Req. D1.2-7.8-
NoColl).

2. MUST use NameIDPolicy/Format of Transient when implementing Linking Service based Push
Model.

3. MUST set NameIDPolicy/SPNameQualifier

4. MUST set NameIDPolicy/AllowCreate flag at all times true

5. SHOULD not set IsPassive flag (in some cases there may be justified reasons to do otherwise)

6. MUST use AssertionConsumerServiceIndex

7. MUST NOT use ProtocolBinding or AssertionConsumerServiceURL

8. Step-up authentication, using Authentication Context Class References MUST be supported.

A.1.3 Shibboleth

Shibboleth MAY be supported. Shibboleth based on SAML 2.0 is RECOMMENDED. Supporting Shib-
boleth enables higher education institutions to adopt TAS³ with minimal reconfiguration and reinvest-
ment.

We have not fully validated all use cases with Shibboleth. Specific points of contention include lack
of full user identification, e.g. statement that User is a student or staff member of university, without
giving out a persistent pseudonym. While a valid approach that better protects the user’s privacy than
the use of a persistent ID, it may not be able to address all the use cases, especially in the commercial
world where service providers wish to link a user’s requests together.

A.1.4 eID

European eID cards are supported as an authentication method available at SAML 2.0 IdP.

A.1.5 Smart Cards

Smart cards are supported as an authentication method available at SAML 2.0 IdP.
A.1.6 One-Time-Password Tokens

One-Time-Password Tokens, such as RSA Tokens or Yubikey, are supported as an authentication methods available at SAML 2.0 IdP.

A.1.7 OpenID

OpenID [OpenID] MAY be supported. If supported, OpenID 2.0 MUST be used as earlier versions have known security flaws.

It should be noted that OpenID’s globally unique identifier model does not provide privacy protection.

We have not validated whether it is possible to implement TAS³ architecture using OpenID. One specific point of uncertainty is passing the IM bootstrap token at SSO time. No native OpenID mechanism is known to exist (standadized; ad-hoc approaches are known). One suggestion, applicable to the RESTful binding would be to use OAUTH.

A.1.8 CardSpace / InforCard and WS-Federation

Card Space MAY be supported. If supported, at least SAML 2.0 token format MUST be supported. The token MUST also support passing IM bootstrap token.

A.1.9 CA / Netegrity Siteminder Proprietary SSO

Siteminder MAY be supported. However, we have not validated whether it is possible to implement TAS³ architecture using Siteminder. Prospects do not look particularly good as the Siteminder protocol and product can not easily be configured to convey the IM bootstrap token.

• Not standards compliant, but by far the most relevant player on the market

A.1.10 Citrix, Sun, and other proprietary SSO

MAY be supported. However, we have not validated whether it is possible to implement TAS³ architecture using these.

A.1.11 Web Local Login

We have not validated whether it is possible to implement TAS³ architecture using local login approach.

• Ad-hoc approaches

A.1.12 Desktop Login

We have not validated whether it is possible to implement TAS³ architecture using desktop login approach.

• Terminal servers: Mind-The-Box, Citrix, Windows TS, etc.
• Active Directory PDC
A.2. SUPPORTED IDENTITY WEB SERVICES SYSTEMS

The Desktop login approach suffers from similar security problems as the Fat Client Login, which see below.

A.1.13 Fat Client Login

We have not validated whether it is possible to implement TAS\textsuperscript{3} architecture in this case. The main security problem in Fat Client Login is that the fat client itself becomes an intermediary to the authentication process, handling sensitive credentials. Some notion of Trusted Computing Path may help to address verifying that the fat client is not compromised.

If Fat Client Login is a requirement, we RECOMMEND Liberty Advanced Client approach, see \[\text{AdvClient}\] and \[\text{SOAPAuth2}\].

A.1.14 User Not Present or Batch Operations

Currently no standards based support. TAS\textsuperscript{3} specifies some approaches for doing this, see \[\text{TAS3D41ID}\], mainly based on using advanced authorization to obtain discovery token without authenticating the User.

A.2 Supported Identity Web Services Systems

The web services must satisfy some technical requirements

- Messages MUST be correlated, so each response is bound to request in an auditable way
  - Message ID correlation
  - Business Process Model and Instance IDs (or context or instance) to allow overarching correlation of several request-response pairs (e.g. to avoid actors who would have conflicts of interest overall that might not be identified when only working at level of individual request-response pairs)
    - PDP can receive this easy enough as an environment parameter and this is needed to support dynamic separation of duties
    - Gap: business process modelling does not express this?
    - Consider URL format hierarchical ID
    - Better typed, like LDAP DN format, or query string
  - Requester and Responder MUST be identified (Req 10.4)
  - Synchronous web service calls MUST be supported
  - Asynchronous calls SHOULD be supported where needed. Business Process Engines will handle asynchrony.
  - Subscribe - Notify mechanism SHOULD be supported where needed
    - subscription for events will be vital to pick up errors and notify of events like break the glass
    - subscribe and publish ws-eventing
A.2. SUPPORTED IDENTITY WEB SERVICES SYSTEMS

- Event bus as a subscribe and publish mechanism

- Maximum availability and use digital signature and encryption technologies, i.e. technical solutions to security and trust problems.

A.2.1 Framework

1. MUST support SOAP 1.2
2. MUST support XML-DSIG [XMLDSIG], a.k.a. RFC3275
3. MUST support Exclusive XML Canonicalization [XML-EXC-C14N]
4. MAY support simple sign [SAML2SimpleSign]
5. MUST support XML-Enc [XMLENC]

A.2.2 Liberty ID-WSF

1. MUST support 2.0
2. MAY support 1.2
3. MUST support following sec mechs, see [SecMech2]
   - "urn:liberty:security:2005-02:TLS:Bearer"
   - "urn:liberty:security:2006-08:TLS:SAMLV2" (Holder-of-Key)
4. MAY support following sec mechs for testing, but MUST NOT permit their use in production environments
   - "urn:liberty:security:2005-02:null:Bearer"
   - "urn:liberty:security:2006-08:null:SAMLV2" (Holder-of-Key)
5. MAY support other TLS [RFC2246] based sec mechs
6. MUST NOT permit non TLS sec mechs in production environments
7. Implementations SHOULD be Liberty Alliance certified, see [IDWSF2SCR].

A.2.3 Bare WS-Security Header or Simplified ID-WSF

1. SHOULD NOT use, as many important security features such as message correlation, replay detection, and identification of end points are not supported by this mechanism.
2. Document resultant limitations if not implementing full ID-WSF.
### A.3. AUTHORIZATION SYSTEMS

<table>
<thead>
<tr>
<th><strong>Liberty Federation Framework</strong></th>
<th><strong>Liberty Identity Service Interface Specifications (ID-SIS)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-FF</td>
<td>Enables interoperable identity services such as personal identity profile, contact book, presence, and so on.</td>
</tr>
<tr>
<td>SAML 2.0</td>
<td></td>
</tr>
</tbody>
</table>

Enables identity federation and management through features such as identity/account linkage, Simplified Sign-On, and simple session management.

<table>
<thead>
<tr>
<th><strong>Liberty Web Services Framework (ID-WSF)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Provides the framework for building interoperable identity services, permissions based attribute sharing, identity service description and discovery, and the associated security profiles.</td>
</tr>
</tbody>
</table>

Liberty specifications build on existing standards (SAML, SOAP, WS-Addressing, WS-Security, XML, etc.)

![Figure A.1: Liberty Alliance Architecture.](image)

#### A.2.4 WS-Trust

- MAY support [WSTrust]

We have not validated whether it is possible to implement TAS³ architecture using WS-Trust.

- Needs heavy profiling to be interoperable, users of WS-Trust should undertake to write such profile.

#### A.2.5 RESTful Approach

MAY support. We RECOMMEND support on basis of OAUTH [OAUTH], but implementers should take in account security advisories published on oauth.net web site.

We have not validated whether it is possible to implement TAS³ architecture using RESTful approach.

RESTful enablement is nice to have, but should not compromise elegance of the SOAP solution and may be less capable (i.e. it is enough that the RESTful approach solves front channel use cases).

#### A.2.6 Message Bus Approach

- SHOULD support AMQP [AMQP06]

#### A.3 Authorization Systems

This section addresses Reqs. D1.2-2.19-AzCredi and D1.2-2.20-Az.
A.3.1 Authorization Queries

1. MUST support XACML 2.0 [XACML2] request-response contexts for authorization queries
2. MAY support other versions of XACML
3. MAY support XACML policy language
4. MUST support XACML SAML Authorization Query extension [XACML2SAML] in order to allow
   policies to be dynamically passed to the PDP.

All communication between the PEP and PDP will be using SOAP based XACML SAML profile. This
profile is mostly independent of rules language. Thus the PERMIS and trust and reputation language
specificity will be mostly contained within the PDPs themselves. The only exception is the obligation
vocabulary which must be understood by the PEP and therefore needs to be standardised. This is a
major effort that will be started in the TAS\textsuperscript{3} project. On the other hand, the sticky policies, which will be
passed over the wire in the protocol exchange, will be engineered such that they transparently pass
from the data store to the appropriate field of the XACML request without the PEP proper really having
to understand them.

A.3.2 Policy Languages

TAS\textsuperscript{3} does not mandate any specific policy language. However, consider following possibilities:

1. PDP SHOULD support XACML 2.0 policy language [XACML2]
2. PDP MAY support PERMIS x.x policy language
3. PDP MAY support P3P x.x policy language
4. PDP MAY support PrimeLife x.x privacy policies

A.4 Trust and Security Vocabularies

Usage of ontologies in TAS\textsuperscript{3} is thoroughly addressed in [TAS3D2UPONTO], which will map some of
these vocabularies.

A.4.1 Vocabularies for Authorization

Some work has been done in RADIUS [RFC2138] and Diameter [RFC3588].
[SAML2context] is mainly about authentication, but authorization is also touched.
This section will be expanded in a future version of this document.

A.4.2 Vocabularies for Basic Attributes (PII)

Use of following vocabularies of PII is RECOMMENDED:

- LDAP inetOrgPerson [RFC2798]
A.5. REALIZATION OF THE DISCOVERY FUNCTION

- Liberty Personal Profile specification [IDPP]
- X.500 standards, such as [X520] and [X521]. See also [RFC2256].

This section will be expanded in a future version of this document.

A.4.3 Discovery Vocabularies

Main vocabulary for discovery is the Service Type taxonomy described in [Disco2]. This taxonomy is complemented by discovery options that further describe the service. This vocabulary SHOULD be used when applicable.

Each Liberty service specifies its own Service Type value as well as a number discovery options. For example, see [IDDAP], [IDPP], or [DST21].

This section will be expanded in a future version of this document.

A.4.4 Security and Trust Vocabularies

See [SAML2context] and [SecMech2] for a vocabulary of security mechanisms that MUST be used when applicable.

This section will be expanded in a future version of this document.

A.4.5 Audit Vocabularies

Audit events from RADIUS [RFC2139] and Diameter [RFC3588] are RECOMMENDED for use where applicable.

This section will be expanded in a future version of this document. As audit is active research topic, we benefit from the research during the TAS³ project to specify this section in detail in the final version of this document.

A.5 Realization of the Discovery Function

- MUST support Liberty ID-WSF 2.0 Discovery Service specification [Disco2]
- MAY support [Disco12]
- MAY support UDDI, however this may require significant extentions to UDDI. Such extentions would need to be profiled.

See [NexoIRA09], section 5.4 “The Overview-Model”, fig 18, for a view of the interaction between service registration and service discovery. Unfortunately the referred document fails to recognize the need for per-identity service registrations, unless the oblique reference, where no difference is made between service requester entity and the data subject, in section 5.4.4 “Service Discovery” counts.

A.6 Realization of the Trust and Privacy Negotiator Function

The protocol to realise the trust negotiation functionality has yet to be finalised. Candidate protocols are:
A.7. REALIZATION OF THE AUDIT AND DASHBOARD FUNCTION

i. the one used by TrustBuilder2 [TrustBuilder2]

ii. one based on the Web Service Profile of XACML [Anderson07] as enhanced by [Mbanaso09]

iii. one based on an enhanced Liberty Discovery Service [Disco2]

Whichsoever protocol is finally chosen it must be able to support a ceremony to gaining incremental levels of mutual trust. The Web GUI of the Front End MUST support the ceremony.

Trust and Policy Negotiation generally takes authentication and identification of all parties for granted, but then computes a trust score which typically governs the access control decisions.

A.7 Realization of the Audit and Dashboard Function

A.7.1 Audit Event Bus

Tentative protocol choice (in order of preference):

1. AMQP [AMQP06]
2. Liberty Accounting Service [AcctSvc] with subscriptions and notifications [SUBS2] and [DesignPat].
3. Diameter [RFC3588]
4. RADIUS [RFC2138]

A.7.2 Audit Event Ontology

- Enumeration of mandatory edit events according to some standard
  - RADIUS and Diameter communities have defined at least some messages

- ZXID logging documentation [ZXIDREADME] provides an idea, at least applicable to SSO

A.7.3 Dashboard Function

- Dashboard should also realize the "PII Consent Service" or "Privacy Manager" at large.

- SHOULD support Liberty Interaction service [Interact2]

A.8 Realization of Delegation Function

The model and protocol to realize the delegation function is still to be determined. Candidates are:

i. Mandate Tokens [Peeters09]

ii. People Service [PeopleSvc] and Cross Principal Referenceing [SOAPAuthn2]

iii. The use of a delegation issuing service [ChadwickEA09]
iv. Other approaches (ID-ToK / WS-Trust) [WSTrust]

v. Ad-hoc and proprietary approaches

The People Service approach supports invitations as described in [TAS3D42Repo] under Section 5.3 Secret URLs. It also supports the whole flow described in 3.3.1. The People Service can be seen roughly equal to Delegation Service specified in TAS³ architecture and somewhat similar to Delegation Issuing Service specified in [ChadwickEA09]. Therefore TAS³ shall refine the People Service until it meets all the requirements.
Annex B: Resilient Deployment Architecture (Non-normative)

This section addresses Req. D1.2-2.8-Aval.

For TAS³ services to be dependable, they need to be deployed so that they are resilient to system and network failure. Resiliency and efficiency are the first lines of defense against Denial of Service attacks that try to attack simple catastrophic vulnerabilities or overwhelm the system on the point where it is most inefficient. Resiliency needs to be considered at several layers, namely on the Front Channel and on the Back Channel.

![Layering of resilience features for Front Channel, Back Channel, and data center Back End services.](image)

Note that the virtual IP address is hosted either in hardware load balancer, or one member of a cluster. Fail-over of the virtual IP is arranged using Virtual Router Redundancy Protocol (VRRP) [RFC3768].

B.1 Zero Downtime Updates

This section addresses Req. D1.2.7.19-DynaUpd.

For continued availability of the system, Zero-Downtime-Update (ZDTU) technology SHOULD be implemented throughout. If horizontal scaling path and failure recovery have been implemented, then ZDTU can be implemented easily by taking out of farm one server at a time and updating it. Downside of this approach is that the farm will temporarily be in an inconsistent state.

If consistency of the farm is at all times a requirement, no easy ZDTU approach exists. One approach is to bring up new "hot standbys" along side of the old configuration and then do instantaneous switch. As the switch over is less than 1 second, this could be considered ZDTU.

Never-the-less, as TAS³ is business process driven and as business processes can take long time to complete (if human interaction is required, this could easily mean days or weeks), thus consistent
ZDTU is infeasible in practise and the business process modelling should explicitly foresee handling of upgrade situations, i.e. how old processes are handled after the general upgrade.
Annex C: Compliance Requirements

C.1 Other Work

- [SAML2conf]
- [IDWSF2SCR]

C.2 General Compliance Requirements

C.2.1 Legal and Contractual Compliance Requirements

CR21-Lawful All legal requirements MUST be satisfied. Members MUST operate within the law.

CR22-Arch All normative requirements of [TAS3ARCH] MUST be satisfied.

CR23-Proto All normative requirements of [TAS3PROTO] MUST be satisfied.

CR24-File Each member MUST be registered on the file at the Trust Guarantor. The filing MUST include details appropriate for the jurisdiction to identify the entity to the extent needed to raise a law suit and/or coordinate investigation with the tax authorities. Typically this means at least

a. Entity name
b. Address
c. Company registration or VAT number
d. Version of Governance Agreement signed and date signed (Req. D1.2-6.13-Contract)

Whenever this information changes, the member MUST promptly inform the Trust Guarantor.

CR25-Policy Each member MUST conspicuously publish a Privacy Policy and Terms of Use for their services on the internet. Member must make available a registry description and offer consultation, rectification, and/or removal of PII.

The Policy and the Terms MUST address at least

a. Entity name and contact for inquiries
b. Data retention policy
c. How is User identified (database keys, properties, such as pseudonymity, of identifier, etc.)
d. With whom data is exchanged and why
e. Whether the policy may change and how existing customers are handled upon the change.

A member MUST adhere to its own Policy and Terms.
C.2.2  General Technical Compliance Requirements

CR26-SSL  All transactions that have monetary value or pass authentication credentials MUST run over encrypted (e.g. TLSv1, SSL or VPN) or physically secure network connections. Alternately the payload itself may be encrypted to similar strength, e.g. using [XMLENC].

For a network to be considered secure, it must achieve a security level equivalent to using any of the following cipher suites (assuming safe and sound key management practises):

a. DSA1024-SHA1-AES128-CBC
b. TLS_RSA_WITH_AES_128_CBC_SHA

This compliance requirement satisfies Reqs. D1.2-2.21-DataProtLaw and D1.2-6.11-Confid.

CR27-Sig  All digital signatures MUST achieve at least the security level equivalent to using any of the following cipher suites (assuming safe and sound key management practises):

a. RSA1024-SHA1
b. DSA1024-SHA1

See threat T141-AltSig.

CR28-Vfy  When data is signed, the intended recipient (see Audience) MUST verify the signature and MUST reject the operation if the verification fails. Verification of the signature MUST include in addition to the actual crypto operations, establishing that the signature was generated by the claimed trusted source.

For each verification, whether failed or successful, audit trail items MUST be generated, documenting at least

a. Signed data or its message digest (e.g. SHA1)
b. Who signed and how his trustworthiness was established
c. Date of signature and verification and the credibility of both
d. Outcome of the verification
e. In case of verification failure due to failed message digest, the input to the message digest function
f. In case of verification failure due to failed public key crypto operation, the input to the operation (e.g. the message digest of the signed data).

See threat T141-AltSig.

CR29-Revoc Whenever long lived or revocable credentials are used (e.g. public key in signature verification), a revocation list or online status service (e.g. OCSP) SHOULD be consulted. If credential is SAML assertion, then long lived means more than 60 seconds. The revocation check SHOULD be done using Assertion Query Profile described in [SAML2prof].

The result MAY be cached for efficiency for duration indicated in relevant protocol and architecture specifications, but lacking clear indication, it should not be cached for longer than risk assessment dictates (if you are confused, do not cache for more than 10 seconds).
C.2. GENERAL COMPLIANCE REQUIREMENTS

**CR210-Rnd** All signature and crypto operations MUST use a secure source of cryptographically strong random numbers. Acceptable sources include

a. Hardware approaches based on electric noise
b. `/dev/random`
c. `/dev/urandom` on busy machines and when seeded from strong source
d. Pseudo random number generator with at least 128bit cycle, when seeded from a strong source (such as user input as in PGP).

Unacceptable sources include

i. Any predictable source
ii. Only seeding with current time and/or process identifier
iii. Less than 128bit cycle or search space

The random number pool should be consulted whenever new randomness is needed, but at the same time care should be taken to make sure that the pool is not unduely depleted of entropy. This is especially a risk when using `/dev/urandom`.

Care should be taken to not to leak the random numbers except as strictly mandated by the protocols.

**CR211-Uniq** Whenever unique identifiers are called for, uniqueness must either be absolute (within specified namespace) or statistical with at least 128bits of search space.

See threat T61-Replay.

**CR212-Trail** Audit trail, including logs, MUST be digitally signed or otherwise tamper proof. Tamper-proofness MUST achieve at least the security level equivalent to using any of the following cipher suites (assuming safe and sound key management practises):

a. RSA1024-SHA1
b. DSA1024-SHA1

depending on circumstances, such as hosting of services in a untrusted data center, the logs SHOULD also be encrypted to achieve a security level equivalent to using any of the following cipher suites (assuming safe and sound key management practises):

i. RSA1024-SHA1-AES128-CBC
ii. DSA1024-SHA1-AES128-CBC

See threat T142-Tamper.

This compliance requirement addresses Reqs. D1.2-2.17-AuditUntamp, D1.2-2.15-Resp, D1.2-6.10-Redress, D1.2-6.17-TechBind, D1.2-4.4-CourtProof.

**CR213-Backup** All backups or batch data transfers MUST be in encrypted form ensuring security level equivalent to using any of the following cipher suites (assuming safe and sound key management practises):
C.3. COMPLIANCE REQUIREMENTS FOR GOVERNING AGREEMENTS

a. RSA1024-AES128-CBC
b. DSA1024-AES128-CBC

See threat T101-LeakBackup and Req. D1.2-2.21-DataProtLaw.

CR214-CertSAML If SAML assertions are involved the software implementation MUST have passed the relevant SAML certification administered by the Liberty Alliance certification program.

CR215-CertIDWSF If Liberty ID-WSF is involved the software implementation MUST have passed the relevant certification administered by the Liberty Alliance certification program.

CR216-EntAn When Systems Entities are required to authenticate each other or assymmetrically one party, HTTPS MUST be supported and other X509v3 certificate based methods (PKI) MAY be supported. HTTP Authentication header based methods MAY be supported.

Authentication requirement CAN be satisfied at VPN, SSL, or application layer (e.g. application layer credentials or trusted digital signature over data). In any case, the authentication MUST be part of the audit trail in a cryptographically strong way and SHOULD be referenced by the summary audit events.

This satisfies Req. D1.2-7.3-An.

CR217-CertCert Certificates used for entity authentication and digital signatures MUST be obtained from a trustworthy authority. Designation of acceptable authorities MUST be made in the Governance Agreement of the Trust Network.

CR218-PrivKey Private Keys MUST be adequately protected. In the minimum this should mean procedural protections against exposure during generation, certification, install, and backup, as well as operational protection using file system permissions. Disclosure of private keys MUST be on strictly need to know basis.

C.3 Compliance Requirements for Governing Agreements

CR30-GA Governing Agreement should at least address

a. Governance structure, such as advisory and audit boards
b. Criteria to join and stay on the network, including certification and audits (Req. D1.2-6.14Compat)
c. Process for removal from the network
d. Process for complaints, arbitration, and disciplinary action (Req. D1.2-6.9-Complaint)
e. Commercial liability and its fair appropriation
f. Liability due to negligence in criminal cases and its fair appropriation
g. Privacy protection
h. Redress for users that have suffered unwarranted disclosure (Req. D1.2-6.10-Redress)
i. Minimal mandatory security practises and policies (Reqs. D1.2-6.11-Confid and D1.2-6.15-MinPolicy)
j. Acceptable use for Service Providers
C.4. COMPLIANCE REQUIREMENTS FOR TRUST GUARANTORS

k. Acceptable use for Users

l. Requirement to be legally bound (Reqs. D1.2-6.16-Bound and D1.2-6.17-TechBind)

CR31-CheckList Any prospective Trust Network member should document the answer to the following questions:

a. Are you collecting or using PII as part of the service?

b. Do you have a Privacy Policy that you are bound to follow?

c. Do you use PII for any purpose other than providing the service?

d. Do you get User's consent or let him opt out before his information is used for other purposes than providing the specific service?

e. Do you share PII beyond your company or family of companies?

f. Do you get user's consent or let him opt out before your share his information with any other company not needed to provide the specific service?

g. Do you allow user to manage these preferences over time and change my options?

C.4 Compliance Requirements for Trust Guarantors

CR41-Col Trusted Guarantor MUST NOT have a conflict of interest with any of the parties that are designed to trust it.

CR42-Records Trust Guarantor MUST maintain credible business records, including:

a. Members of the Trust Network (see CR24-File).

C.5 Compliance Requirements for Service Providers

CR51-DNSpub Service Provider MUST use DNS to publish its network addresses in a symbolic form. This requirement facilitates reconfigurations of the network. It is a well accepted "best practise".

CR52-BPM Service Provider's business processes MUST be modelled.

CR53-DontLogTok Service Requester SHOULD NOT log, even in encrypted form, the the tokens destined to the Service Responder or other parties if threat T107-LogTokLeak is a concern. If audit trail requires logging tokens, then the tokens must be blinded so that the correlatable part is not visible or the token MUST be encrypted such that legitimate viewers of audit trail can decrypt it, but SP itself can not.

Compliance with this requirement is established with audits.

CR54-CorrConsent Service Provider MUST have user's consent before leaking a correlation handle of any kind.

CR55-MDExp Service Provider MUST implement Well-Known Location (WKL) method of metadata export, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method.
C.6. COMPLIANCE REQUIREMENTS FOR SERVICE REQUESTERS

CR56-MDImp  Service Provider MUST implement Well-Known Location (WKL) method of metadata import, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method. The Import MUST NOT unintentionally lead to a trust relationship.

CR57-VfyAn  Service Provider MUST authenticate the Service Requester according to CR216-EntAn.

CR58-An  Service Provider MUST authenticate itself to the Service Requester according to CR216-EntAn.

C.6  Compliance Requirements for Service Requesters

CR61-DNS  Service Requester MUST use DNS to resolve names. This requirement facilitates configuration and provides a load balancing method (round robin DNS) for the SPs. DNS query results MUST NOT be cached beyond their TTL.

CR65-MDExp  Service Requester MUST implement Well-Known Location (WKL) method of metadata export, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method.

CR66-MDImp  Service Requester MUST implement Well-Known Location (WKL) method of metadata import, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method. The Import MUST NOT unintentionally lead to a trust relationship.

CR67-VfyAn  Service Requester MUST authenticate the Service Provider according to CR216-EntAn.

CR68-An  Service Requester MUST authenticate itself to the Service Provider according to CR216-EntAn.

C.7  Compliance Requirements for Databases Storing PII

Since Databases Storing Personally Identifiable Information (PII) usually are SPs, the requirements for SP also apply.

A future version of this document will specify in detail

- Special encryption concerns
- Special privacy protection
- Record keeping and audit

C.8  General Compliance Requirements for Trusted Third Parties

CR81-CoI  Trusted Third Party MUST NOT have a conflict of interest with any of the parties that are designed to trust it.
C.9 Compliance Requirements for Identity Provider

CR91-CoI Identity Provider MUST NOT have a conflict of interest with any of the Service Providers or Users. In general, IdP functions cannot be performed by a SP.

CR95-MDExp Identity Provider MUST implement Well-Known Location (WKL) method of metadata export, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method.

CR96-MDImp Identity Provider MUST implement Well-Known Location (WKL) method of metadata import, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method. The Import MUST NOT unintentionally lead to a trust relationship.

C.10 Compliance Requirements for Discovery Providers

CR101-CoI Discovery Providers MUST NOT have a conflict of interest with any of the Service Providers or Users. In general, the discovery and token mapping functions cannot be performed by a SP.

CR105-MDExp Discovery Service MUST implement Well-Known Location (WKL) method of metadata export, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method.

CR106-MDImp Discovery Service MUST implement Well-Known Location (WKL) method of metadata import, see [SAML2meta] section 4.1 "Publication and Resolution via Well-Known Location", p.29, for normative description of this method. The Import MUST NOT unintentionally lead to a trust relationship.

C.11 Compliance Requirements for Trust and Reputation Provider

CR111-CoI Trust and Reputation Provider MUST NOT have a conflict of interest with any of the Service Providers or Users to which it provides trust scorings.

C.12 Compliance Requirements for Audit Provider

CR121-CoI Audit Provider, Audit Event Bus operator, or shared Event Bus Operator MUST NOT have a conflict of interest with any of the Service Providers or Users. In general, apart from SP internal audit, the audit functions cannot be performed by a SP.

C.13 TAS\(^3\)-Lite Compliance Profile

The compliance requirements have been drafted to ensure true security and accountability. However we recognize that some of the compliance requirements are quite onerous and could be a hindrance to TAS\(^3\) adoption in some low value situations. Therefore we define in this section a TAS\(^3\)-Lite profile that can be used in low value situations as long as the risks are recognized and the deployment is not misrepresented as fully TAS\(^3\) compliant. The TAS\(^3\)-Lite relaxations are as follows:
C.13. TAS³-LITE COMPLIANCE PROFILE

1. CR24-File and CR25-Policy are dropped. Informal means should be used to achieve the same end result. Dropping these requirements seriously compromises the ability of the Trust Network and the Users to hold parties accountable.

2. CR214-CertSAML and CR215-CertIDWSF are dropped due to financial cost of the certification. Attending cheaper informal interop events is still highly recommended.

3. CR217-CertCert is dropped. Self-certification is allowed.

4. CR30-GA is dropped. Informal governance structure is allowed. The consequence of this is most likely that parties can not be held responsible in case of serious violations.

5. CR52-BPM is dropped. Informal modelling is still recommended.

Subject: RE: Status D4.1?? From: "Danny De Cock" <decockd@esat.kuleuven.be> Date: Sun, May 31, 2009 20:53 To: Sampo Kellomäki <sampo@symlabs.com> Cc: "Montagnon, Gilles" <gilles.montagnon@sap.com> (more) Priority: Normal Options: View Full Header | View Printable Version | Download this as a file

sure...

d.

http://godot.be

godot:at:advalvas:dot:be http://godot.studentenweb.org

On Sun, 31 May 2009 sampo@symlabs.com wrote:

Danny De Cock wrote: > On Sun, 31 May 2009 sampo@symlabs.com wrote: » Danny De
Cock wrote: » > dear sampo, » » > I am going through your document now, provide you
with feedback this » » evening, and ship the stuff tomorrow early afternoon. » » Thanks.
I'll standby late tonite to revise. » » Regarding arch Annex C and the MD5 problem » »
CR26-SSL: I will remove RSA1024-MD5-3DES-CBC, so that the requirement » w/David's
contribution, will read: » » new applications should no longer use deprecated algorithms,
e.g., md5 > and des. 3des has not yet been formally deprecated, but should no more
be > promoted... » » cu later, d. » » therefore you might wish to restrict the list to »
TLS_RSA_WITH_AES_256_CBC_SHA > TLS_RSA_WITH_AES_128_CBC_SHA

> They share SHA1 so SHA1 compromise would compromise both?

> They share also RSA.

> The AES is shared, albeit with different bit lengths. Is this really enough diversity?
C.13. TAS³-LITE COMPLIANCE PROFILE

> –Sampo

> » a. DSA1024-SHA1-AES128-CBC » b. RSA-AES-128-SHA » c. TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA

» » Does this sound agreeable? » » In general we need always two mandatorily supported suites in case one is compromised. I agree MD5 should be considered compromised. Wide deployment of compromised algorithm is not really an excuse. » » CR212-Trail: I will remove the MD5 variant, but I need a second cipher suite. For reference, it read now: » » (REMOVED: i. RSA1024-MD5-3DES-CBC ) » ii. DSA1024-SHA1-AES128-CBC

» » What would you suggest? » » There was a mention of MD5 in some example. This has already been removed. » » Cheers, » –Sampo » » > cu later, d.
Annex D: Generalized Use Cases

Non-normative. The simulated user interface screenshots in this section are NOT normative. They serve merely to illustrate one feasible way of designing the user interface. The user interface flows are also non-normative, for example the IdP detection or already-logged-in detection may follow different paths. Every step of the way, confirmation questions, wizards, and other user interface devices may be inserted. Depending on business model and branding choices of the Trust Network, there may be some graphical guidelines and restrictions, see [TAS3BIZ] and Governing Agreement of the Trust Network.

This section addresses Req. D1.2-2.13-Easy, among others.

These Use Cases deal with User Interaction, therefore they do not illustrate the rather large Web Services proportion that TAS³ architecture mainly aims to address. Never-the-less, in a User Centric system, we must start with the user - without his impulse (direct or indirect) the back-end Web Services should never happen.

A general assumption has been that Single Sign-On (SSO) will be used, though some other approaches are foreseen as well. Long tail services should especially use SSO as it is unreasonable to ask for user registration for one-off service request.

Methodology. In the Story Boards that follow, the sequence describes user’s perception. It does NOT describe protocol flow, which can at times be quite different from User’s perception. For example, many SSO protocols call for HTTP redirects, so technically speaking any transfer between screens should pass via User Agent. A big circle in diagram means a protocol step that usually is optimized so that no page is shown to the user (but astute users may notice some flicker). When the optimization for some reason does not work out, the regular user interface screen will be shown. We apply Cognitive Walkthrough method [Wharton94] to elaborate the story boards.

Figure D.1: User accesses Front Ends using Single Sign-On.
D.1 User Uses Service (First Time in the Session)

The first time use of a service in a session consists of

- First the User interacts with the Front End (FE)
- The User is redirected to IdP (cf. Req 3.1 Existing Accounts)
- The User logs in at IdP
- The User is redirected back to the protected content

This means minimum three steps, but there could be more if there are confirmation questions.

**Trust Seals.** As can be seen, the user interface is expected to display trust seal of the Trust Network and may display TAS\(^3\) seal as well. These are intended as visible indicators that public associates with trust. Their exact design and realization, including the possibility of not displaying them at all, will depend on the particular Trust Network.

![Figure D.2: Story board: Using service for 1st time in a session.](image)

**Cognitive Walkthrough**

1. **Choice of IdP**

   **Motivation** User has taken initiative to perform a task he thinks can be accomplished using a web site. He realizes that some form of authentication or authorization will be required. When the User navigates to the task, a dialog is presented asking for authentication so that authorization can be granted. User will consider engaging in this dialog because they feel the system is trustworthy, based on the Trust Seals and based on past successful experiences.
D.2. ALREADY-LOGGED-IN OPTIMIZATION (SSO)

**Available and understandable** User will be guided by modality of the interaction to a situation where he will either have to proceed with selection of an IdP or will have to abandon the task. Choosing another task that does not require authentication is also an option. The interaction should be structured such that the requirement for authentication will become evident early on, so that User avoids performing work only to find out that he is unable to proceed.

**Feedback** The available IdP choices that are presented should be as narrow and relevant as possible. Federated SSO research recognizes IdP selection as a major problem. Once IdP is chosen and button is pressed, clear feedback is provided that User has landed on the IdP web site. The IdP screen should provide contextual information about the task which motivated the authentication (such feedback is lacking in step 2 of Fig-D.2).

2. **Login**

**Motivation** User is in the mind set of completing a task and will perform this step if he reasonably can. This mind set is reinforced by IdP providing feedback as to what task requires the authentication.

Biggest challenge and inconvenience for the User will be the necessity to present authentication credentials. This inconvenience can be mitigated by use of Single Sign-On.

**Available and understandable** Availability of the logon and the acceptable forms of credentials should be self-evident from the first screen of the IdP. First screen should lay visible all options and avoid any hierarchical navigation to arrive to the desired option.

**Feedback** Successful authentication will lead to User being returned to the Front End web site. This in itself is a form of feedback, but it should be reinforced by the web site providing a clear welcome greeting, stating that the User has been authenticated (and possibly authorized as well).

3. **Login complete.** This use case ends here, but an application specific use case will start here.

D.2 Already-Logged-in Optimization (SSO)

Same as above, but without IdP authenticating the user again. The flow does not need to stop at IdP at all. Optimized SSO use case, showing the full convenience of SSO, leading to 2 step process.

**Cognitive Walkthrough**

1. **Choice of IdP:** Same cognitive walkthrough as in previous section.
2. **Login:** No cognitive walkthrough needed as no user interface will be presented.
3. **Login complete.** This use case ends here, but an application specific use case will start here.

D.3 User Uses Dashboard

This use case addresses Reqs. D1.2.2.11-Transp and D1.2.3.3-Dash.

In this use case the user interacts with the TAS3 Dashboard in order to determine the status of a business process he is engaged in. It consists of the following steps:
D.3. USER USES DASHBOARD

Figure D.3: Story board: Using further services after logging in at IdP - Single Sign-On (SSO).

- The user logs into the Dashboard (possibly using SSO)
- The user sees a page with an overview of the transactions
- The user drills down to visualise a particular business process.
- The user views a particular audit trail and discovers a suspect item.
- The user requests a legally binding audit statement about the transaction.
- Competent authority requests further information about the transaction from the Service Provider that holds the detailed audit trail.

Cognitive Walkthrough

1. Engaging Dashboard and Choice of IdP

Motivation User has taken initiative to find out about the state of some business process or the handing of his PII. User understands, due to training or awareness campaigns, or because a noise was given in the beginning of the business process, that this is possible. User may have found out about the possibility by surfing the web or through a search engine. The mere possibility may spark the User's interest and get him to try the Dashboard out. User may also have noticed an irregularity or complained to some instance and been told to consult his Dashboard.

Available and understandable Since User is assumed to take initiative, a major hurdle will be how the user finds out about the Dashboard and how to contact it. Some possibilities
D.3. USER USES DASHBOARD

Welcome, Alice!

1. Currently open processes
   - Competencies certification at Job Agency
   - Life-long learning at Alumni Portal

2. Recent completed processes
   - M.Sc. degree

3. Archive

(1) Dashboard
IdP is detected and User automatically redirected to it.

(2) Identity Provider 1
Session already active, no need to login again (SSO).

(3) Dashboard
Welcome, Alice!
1. Currently open processes
   - Competencies certification at Job Agency
   - Life-long learning at Alumni Portal
2. Recent completed processes
   - M.Sc. degree
3. Archive

(4) Dashboard
Competencies Certification at Job Agency
Active Business Process Visualization
You Are Here

(5) Dashboard
Detail of request to Alumni Portal
Req ID X23AD of 30.3.2009 by Job Agency
* Requested Information: Name, DoB
* Policy pledges: PL334 (non commercial)
* Returned Information: Name, YoB
* Obligations: O765 (retention)
* Audit trail pointers: M123, R343, T225

Figure D.4: Story board: Using Dashboard to audit a business process.

a. A link to the Dashboard is provided as part of the user interface of each business process.
b. A link to Dashboard is provided in every web site that participates in the Trust Network.
c. Trust Network operates some sort of a portal and the link can be found there.
d. Dashboard engages in Search Engine Optimization (SEO) so that User is sure to find the Dashboard through a popular search engine.

Once the user has found out about the Dashboard, the problem shifts to the IdP selection and...
D.3. USER USES DASHBOARD

authentication. In Fig-D.4 we have assumed that IdP can be detected and User is already logged in, as the case typically would be immediately after engaging some Front End (e.g. the Job Agency).

However, if time has passed, user may need to choose explicitly an IdP and explicitly authenticate, as in Section “User Uses Service (First Time in the Session)”. A confusing situation can arise where user has tried to access the Dashboard, but the first screen he sees is the IdP authentication screen (because IdP detection worked, but user was not logged in yet). This situation should be mitigated either by IdP providing enough context about the operation that is motivating the authentication, or by the Dashboard imposing a splash screen even when IdP choice is already known.

Feedback If IdP was detected and user was already logged in, the first feedback will be Dashboard logged in welcome screen. If authentication is needed, then the IdP context message or the splash screen solutions should be adopted, as described in the previous paragraph.

2. Login: no specific cognitive walkthrough requirements. See discussion in First Time use case.

3. Choose Business Process to Audit

Motivation User set out on his quest to perform this task.

Available and understandable The list of the business process instances should be structured so that all business process instances are reachable while at the same time the processes user is most likely to be interested in are presented first or more prominently. Due to potentially large number of processes, we may need to resort to hierarchy or search functions. An ontology of business processes will help in setting up the hierarchy and search. The business processes should be titled and described in language that the User can relate to. In particular, while codes can be provided for accuracy and reference, every business process should have a human readable name. The resultant translation issues will have to be recognized and addressed.

Feedback Choice of a business process instance will lead to its visualization where User can clearly identify What, Who, When, and similar information so that user can confirm he has made the right choice. If choice was wrong, User should easily be able to choose another instance.

4. Choose Detail of Business Process Instance to Audit

Motivation Once user sees visualization of the business process instance, he will need to drill down to relevant detail. This may be driven by User’s curiosity or perceived notion of culpable part.

Available and understandable The visualization has to be structured so that it honestly depicts the essence of the business process without cluttering the view with details that can be reached later. Every step that User is expected to perform (or has already performed) should be visible as well as major processing steps that are not in User’s control, especially those that involve transfer or manipulation of PII.

All descriptions of the steps should be succinct and in human language, with translation issues addressed. Codes and references for the instance and steps can be provided for accuracy, but these should never supplant the human description.
To assist User in drilling into detail, the user interface should make it patently evident where this possibility exists, e.g. by using high-lighting techniques.

**Feedback**  User is assisted in contemplating the choice of drill-in by high-lighting of available options. Once a step is chosen for scrutiny, user will see visualization of that step in great detail. The visualization will be titled in such a way that it is evident to the User that it pertains to the step he chose in the business process instance overview.

5. **View detail and request audit item from Front End**

   **Motivation**  User needs to get evidence about a step of a process

6. **View audit item**  (not depicted in the figure)

7. **Escalate**  (not depicted in the figure)  (Req. *D1.2-6.9-Complaint*)

**D.4  IdP Detected-Optimization (SSO)**

This flow, see Fig-D.5, can further optimize the already logged in case by allowing the Job Agency to detect that the user has already chosen IdP and therefore use the IdP to log the User in automatically. Essentially the ceremony becomes a one step process.

**D.5  User Uses Service, Identity Selector Case**

In the Identity Selector flow, see Fig-D.6, the User never interacts with the IdP directly. Instead, the Identity Selector provides a user interface (step 3) for the IdP to query authentication credentials. User
Figure D.6: Story board: Identity Selector provides IdP User Interface.

experience is entirely managed by the "ceremony" that the Identity Selector presents.

Figure D.7: Story board: Using services with local login (not recommended).
D.6 User Uses Service, Local Login Case

N.B. This use case is not recommended. You should use SSO based approaches instead. We document it here only to illustrate the problems associated with multiple logins.

The assumption is that the user will use more than one service. This highlights the inconvenience of user having to authenticate separately to each service. There are further complications under the hood, not least of which are privacy threats. This scenario could be called explicit account linking.

While we consider supporting this scenario to be in scope, we do not recommend it unless there is no alternative, or as temporary solution.

D.7 User Uses Service, Proxy IdP Case

This sequence, see Fig-D.8, illustrates the experience of a user logging in to SP that does not directly trust his IdP. The trust is mediated by the "middle" IdP that SP trusts.

This sequence can be further optimized if the middle IdP can somehow automatically detect which IdP is the home IdP (similar to Section IdP Detected Optimization SSO) and, of course, if the User is already logged in the SSO optimization of Section Already Logged-in Optimization SSO.

D.8 Consenting to PII Release or Manipulation

This section addresses Reqs. D1.2-6.3-What, How, Why, Who, D1.2-6.6-Consent, D1.2-6.7-Reconsent, D1.2-4.1-Enf, UCPol.
D.8. CONSENTING TO PII RELEASE OR MANIPULATION

D.8.1 Interaction on Front Channel

The obvious choice of having the requesting SP collect User’s consent has an obvious conflict of interest issue. In some legal contexts this may be acceptable, but in general we need a way for either the releasing party or some Trusted Third Party to collect the consent.

Alternatively, not shown here, the user may explicitly provide his consent by authenticating to the releasing party and authorising it to release the PII to the SP. Further user cases for accessing releasing parties who are repositories and authorising third party access to repository contents are provided in [TAS3D42Repo].

Cognitive Walkthrough

1. IdP choice as usual
2. Authentication as usual
3. User triggers action, as usual
4. Consent to release of PII

**Motivation** User will be motivated to take action because it is imposed to him by the modal flow of the interaction. User will be pleased to take action because asking consent is in his protection, but Users do get annoyed if you ask too often - to solve this we would need Privacy Agent, whose Use Cases are to be elaborated later (M30 D2.1?).

**Available and understandable** Presentation of the consent question is a major challenge. It needs to be succinct, yet comprehensive and legally binding. Some Users will want high degree of detail and control, while others will be confused by too many options. Fig-D.9 depicts a dummied-down interface. This may not be appropriate for some users.

**Feedback** Once consent is given, User lands on page that uses the consented information. This may be sufficient in its own right, but could be enhanced by highlighting the information on the page the user just consented to.

5. Business process continues with the PII as usual

D.8.2 Interaction on side channel

This Use Case is similar to the previous one. Only difference is that the consent is asked using a Side Channel, such as mobile phone or instant messaging. The side channel provides an independent means of communication, a type of second factor to the consent.

The Side Channel approach can also be convenient when consent needs to be asked deep in SOA Web Services calls where Front Channel is not available.

In User-not-present transaction the Side Channel may be the only option for asking user’s consent, or else the business process needs to be stopped until user provides consent via Dashboard.

D.8.3 Interaction via Dashboard

In User-not-present transaction the business process may stop until user provides input or consent via Dashboard. This alternative will be covered in a future version of this document.
D.8. CONSENTING TO PII RELEASE OR MANIPULATION

Figure D.9: Story board: Presenting a PII consent question in Front Channel interaction.
D.8. CONSENTING TO PII RELEASE OR MANIPULATION

Welcome, Applicant!
Here are your competencies.
... (protected content)
You have requested protected content, please login.
Using: IdP 1 Login

Identity Provider 1
Session already active, no need to login again (SSO).

Job Agency
User
Welcome, Applicant!
Here are your competencies.
... (protected content)
You have requested protected content, please login.
Using: IdP 1 Login

Job Agency wants to know your competencies. Please consent to release of:
[x] High school grades
[x] University diploma
Reply to this SMS with code x98sd1 if you agree.
[Reply]

SMS
C
TAS3
TAS3
TAS3
TAS3
TN
TN
TN
TN

Job Agency

Job Agency

Job Agency

Job Agency

Job Agency

Please wait while your consent is asked via mobile phone.

Figure D.10: Story board: Presenting a PII consent question using Side Channel interaction.
D.9 Using Linking Service

1. The Linking Service should be user friendly. It may be the only interface that users see for linking their attributes together (other approaches are possible, see "pull model").

2. A welcome screen explains the purpose of the Linking Service and guides the user through the process of attribute linking. It has:
   a. Picking list for choosing IdP
   b. "Connect" button
   c. "View linked accounts" button
   d. "Make linked accounts available to services" button
   e. Notice or pledge about respecting User's privacy

3. When the user selects the "Connect" button, the linking service will redirect the user to the selected IdP, allowing the user to login. After login, the user will be redirected back to the linking service welcome screen.

4. When the user selects "View my linked accounts" he will be presented with the screen with:
   a. A table containing two columns, labelled "Organisation" and "Temporary Account Identifier" and at the left hand side by each table entry will be a tick box that the user can tick to remove the linked account. Above the column of tick boxes will be the word Delete.
   b. "Delete" button, which will remove the chosen accounts from the table and return the user to this page
   c. "Home Page" button, which will take the user to Welcome screen
   d. "Make my linked accounts available to services" button, which will take the user to the next screen.
   e. Notice or pledge about respecting User's privacy

5. When the user selects the "Make my linked accounts available to services" button he will be presented with a screen containing:
   a. An explanation about opt-in in the linking (if you do not make accounts available, the default will be no linking).
   b. A table with 3 columns and a delete tick box beside each row of the table. The table columns are "Service", "Organisation" and "Temporary Account Identifier". The table will always be empty for new users when they first approach this screen.
   c. A picking list of all the services in the federation, obtained from the metadata of the federation. The first entry in the list will be "All Other Services".
   d. Once the user selects a service provider or "All Other Services" from the picking list, a picking list of all the IdPs that are currently linked together and that appear in the table of the My Linked Accounts Screen, minus the IdPs that have already been paired with the selected service provider is displayed.
D.10. CHOOSING AMONG MULTIPLE SERVICE PROVIDERS

The first row of this picking list will be "All My Linked Accounts". The user will then pick one of his linked accounts or "All My Linked Accounts". If the user picks "All My Linked Accounts" a wild card will be inserted into the third column. If the user picks one of his accounts then the third column will be automatically completed with the account Persistent ID unless the user has two or more accounts at the same IdP, in which case the third column will contain a picking list of Persistent IDs sent from that IdP, minus any already selected for this service provider.

It is important that the table always lists the service providers in alphabetical order so that the user can easily see which links he has set up for which SPs, and for every SP, the linked IdPs are in alphabetical order.

e. "Delete" button, which will remove the chosen accounts from the table and return the user to this page

f. "Home Page" button, which will take the user to Welcome screen

g. "View my linked accounts" button, which will take the user to the screen referred to in step (4), above.

D.10 Choosing among Multiple Service Providers

Sometimes user will have choice of multiple possible providers for a given service. In this situation Trust and Privacy Negotiation function can be used to narrow down the list. If after narrowing down more than one choice still remains, it may be reasonable to ask the user to make the choice.

D.10.1 Simple Choice of Provider

Cognitive Walkthrough

1. IdP choice as usual

2. Authentication as usual

3. User triggers action, as usual

4. Choose Service Provider

Motivation  The decision point will be imposed to the user through modal user interaction. User will be motivated to make the choice as he may guard different information, e.g. different personae, at different Attribute Authorities.

Available and understandable  User's choice should only be solicited if there is genuine choice. System should implement automatic discovery and detection as much as possible.

The choices should be formulated in human language, with translations as appropriate.

Feedback  Once User makes his choice, he will land on the requestor's page. This in itself may be sufficient feedback, but indicating on the page where the information came from is recommended.
Job Agency

You have requested protected content, please login.

Using: IdP 1 Login

Identity Provider 1

Session already active, no need to login again (SSO).

Job Agency

Welcome, Applicant!
Here are your competencies.
... (protected content)

Job Agency

Your competencies are available from multiple sources. Please choose:

- University
- Employer

Get Cancel

Your competencies are available from multiple sources. Please choose:

- University
- Employer

Get Cancel

Job Agency

Detail of competency provided from University.

Figure D.11: Story board: Choice of Service Provider.

D.10.2  Trust and Privacy Negotiation Assisted by User Interaction

Cognitive Walkthrough

1. IdP choice as usual
D.10. CHOOSING AMONG MULTIPLE SERVICE PROVIDERS

Figure D.12: Story board: Trust and Privacy Negotiation with User Interaction.

2. Authentication as usual

3. User triggers action, as usual

4. Negotiate appropriate supplier for service or information
D.11. USER-NOT-PRESENT TRANSACTION

**Motivation** User will be forced to the decision point by modal user interface flow. User will be motivated to make a choice either because he has no prior relationship with proposed SPs and he needs to rely on trust preceptions, or because user wants to be in control and avoid machine deciding for him.

**Available and understandable** Presenting complex trust based decision is not easy. This topic will be further researched during TAS$^3$ project.

**Feedback** Once User makes his choice, he will land on the requestor's page. This in itself may be sufficient feedback, but indicating on the page where the information came from is recommended.

Further Use Cases depicting complex Trust and Privacy Negotiations will be elaborated in other project deliverables.

D.11 User-Not-Present Transaction

User-not-present scenario can be driven in three ways:

1. User has been present in some earlier time and authorized, indirectly, the transaction. Audit trail MUST show this authorization.

2. There is an over-arching legal or legitimate business requirement. Existence of such requirement MUST be demonstratable from the audit trail.

3. "Break the glass" scenarios. Again audit trail MUST capture legitimate reason why the scenario was invoked and the audit trail should be especially detailed about the actions performed under the break the glass authority.

Actual triggering of the event will depend on a business process. To gain acute authorization to execute the operation, the business process will have to declare its intent and show evidence why it should be authorized (see (1) and (2), above). Then, the operation MUST be thoroughly recorded in the audit trail.

User’s only contact point with User-Not-Present transaction is to audit it after the fact using the Dashboard.

D.12 User Present Delegation

See Fig-D.13.

- Problem of choosing to whom to delegate, buddy list visualization
  - How to obtain human readable names without violating privacy of the buddies?

Delegation of permissions to access repositories is addressed more fully in [TAS3D42Repo].
D.13 User-Not-Present Delegation

This will cover situations such as administrative or judicial decisions that result in delegation without the User necessarily wanting the delegation to happen.
We will explore these use cases in more detail in a future deliverable (M30 D2.1).

D.14 Other Use Case Work

[TAS3D42Repo] has an extensive section on use cases, which should be viewed as a complement or extension of what is presented here. [TAS3D14DESIGNREQ] has some usage scenarios, especially relating to the pilots, although they are not refined into use cases.

D.15 Future Use Case Work

Some other User Cases we may elaborate on, or that will be elaborated in other TAS³ deliverables, include:

- Full elaboration of the Trust and Privacy Negotiation Use Case(s)
- SP BPel4People UI
- Trust Guarantor UI
- SP registration process UI
- Bulletin board UI's
- Statistical services from anonymised data UI
- Situation where additional data request deep in the recursive Web Services or business process requires Step-Up authentication
- Processes that may take long time and have start stop states taking longer than a web service call can be reasonably expected to take.

BPEL engine can monitor this: any timeout is service failure and recorded as such. All service providers must agree to terms SLA on sign up to TAS³ network and a key element of this will be service reliability and performance.

- Human steps in process flow can be slow (e.g. process can be waiting sometimes for days / weeks)

- Use case: User wants to audit and complain
  - like on ebay give negative feedback and influence reputation of Service Provider
  - Complaining to wrong entity
  - Misidentifying probable cause
  - Ability trace all the way to the legal evidence

- 3rd party wants to audit or demonstrate that something happened,
  - nonrepudiation
D.15. FUTURE USE CASE WORK

- articulation to proof in law suits

- Registering a new service to the trust network
Annex E: TAS³ Business Model (Non-Normative)

This Annex has been moved to the end of the PDF document. In future the business model will be maintained and published as a separate document.
Annex F: Threats (Non-Normative)

This section addresses Reqs. D1.2-2.7-Safe and D1.2-2.8-Avail. Also Req. D1.2-2.9-Correct is touched from secure programming perspective.

F.1 Other work

- [SAML2security]
- [IDWSFSecPriv]
- [NIST-SP800-30]
- [Meier09] provides a check list and [Meier08] a short list

F.2 Business threats

T21-IDTheft Identity Theft
T22-Ilegal Illegal transactions

F.3 Trust model threats

T31-OverPrivil Over-privileged process and service accounts.
T32-UnTTP Untrustworthy Third Party.

F.4 Architectural threats

T41-BPMflaw Business process modelling flaws. How to audit or validate the model?
T42-BPMdel Accidental deletion of process steps such as authorization or validation
T43-Berserk Dynamically adaptive business process gone wild

F.5 Trust misconfiguration threats

T51-TNins Insertion of illegitimate members in trust network.
T52-Mole Moles in trust network. A previously trusted member turns into rogue. How to detect? How to rectify?
T55-PolluteRep Pollution of reputation of other party
T56-Augment Illicit augmentation of reputation by party itself
F.6  Protocol misconfiguration threats

T57-Deface  Defacing Front End website can cause damage to reputation of the Front End. Defacing attacks can happen through same means as phishing (T111-Phish) and also through break-ins, either through network, or physical.

T58-WrongCAs  Insertion of wrong CAs into a trust network

T59-WrongAAs  Wrongly assigning source of authority to an attribute authority

T510-WrongLOA  Wrong assertion of LOA value by an IDP

F.6  Protocol misconfiguration threats

T61-Replay  Replay attack. See CR211-Uniq.

T62-UnEncLink  Unencrypted links e.g. SSLnull cipher suites.

T63-UnAnLink  Unauthenticated links

F.7  Authorization misconfiguration threats

T71-WrongGrant  Returning grant instead of deny leading to unauthorised access

T72-WrongDeny  Returning deny instead of grant leaving to DOS.

T73-WrongObs  Returning wrong obligations

T74-MissingObs  Missing obligations from authz decisions

T75-OKAC  Attribution of wrong access rights to an otherwise trusted member

F.8  Ontology threats

T81-SameButDiff  Same term meaning different things to two parties leads to illicit access, due to wrong rule inadvertently matching.

F.9  Exposure threats

T91-Eavesdrop  Exposure of data due to network sniffing or eavesdropping. Counter measure: encrypt data and manage keys right.

T91-DBLeak  Exposure of data due to database files or transaction logs. Counter measure: encrypt data and manage keys right.

T92-TamperNet  Modification of data in transit. Counter measures: signing or verification of a hash over the data.

T93-TamperDB  Modification of data in database.
T94-ExptLeak Error condition or exception reveals too much data or system details (usually to aid debugging). Exception output that appears in user interface or over the network is especially damning. Leakage to logs is also a threat.

T95-CoreLeak Error condition or exception causes a core dump that reveals too much data or system details (usually to aid debugging).

F.10 Privacy threats

T101-LeakBackup Eavesdropping on backups (see CR213-Backup)

T102-Correlation Database correlation by colluding entities (solution: do not leak correlation handles, i.e. use pseudonyms - see Architecture, Core Security Architecture, Access Credentials, Pull Model)

T103-TAIdP IdP collects traffic analysis (and then sells or illicitly use it). Some counter measures:

- TN wide data retention policy, audit this: add compliance requirement
- Pure play IdP operator vs. mixed functions
- Centralized IdP well managed may be a good idea

T104-TADI Disco collects traffic analysis (and then sells or illicitly use it)

T105-TA3rd Traffic Analysis by Third Party

T106-CorrAudit Correlation handles of audit trail will also become correlation handles.

T107-LogTokLeak If WSC parties keeps log of User's pseudonym along with encrypted form of User's identifier at WSP, then WSC and WSP can correlate and collude using the encrypted form. However this threat is acute only between directly interacting parties. In a chain of web services calls longer than 3, the nonneibouhiring parties are not in position to collude using this attack. Current solution is to forbid logging the tokens, see CR53-DontLogTok.

T108-PhishPII Tricking user to reveal PII through phising attack that poses a real looking web page to solicit PII. See also access version of the threat: T111-Phish.

T109-SocEngPII Social Engineering, talking users to revealing PII. See also access version of the threat: T112-SocEng.

T1010-SnoopPII Network eavesdropping to record PII.

T1011-KbdLogPII Keyboard logger or other malware to record credentials.

T1012-MalwarePII PII theft, e.g. copy private contact book, using malware.

T1013-PIITheft Physical theft of PII.
F.11  Authentication threats

T111-Phish  Tricking user to reveal his authentication credentials through phising attack that poses a real looking web page to solicit user’s access credentials. This could be created through

a. DNS manipulation  
b. Cross site scripting  
c. Inappropriate insertion of content in legitimate site  
d. Containment of legitimate site in illegitimate frame

See also PII version of threat: T108-PhishPII.

T112-SocEng  Social Engineering, talking users to revealing access credentials.

See also PII version of threat: T109-SocEngPII.

T113-SnoopCred  Network eavesdropping to record credentials.

T114-KbdLog  Keyboard logger or other malware to record credentials.

T115-Malware  Credential theft, e.g. copy private key, using malware.

T116-Theft  Physical credential theft.

T117-Dict  Dictionary attack on password

T118-Brute  Brute force attacks of simply trying out all credentials.

T119-Cookie  Cookie replay attack. Use previously recorded cookie in context where authentication did not happen. Also arises if expired session cookie is allowed as a factor in authentication, resulting stronger factor not being demanded.

T1110-Lure  Luring users to do stupid things like

a. Visit web sites that phish or contain malware  
b. Install malware and troians  
c. Voluntarily give out credentials or PII

F.12  Impersonation threats

T121-Gift  Users giving away their credentials to others

T122-Loss  Users losing their credentials

T123-Careless  Users not protecting their credentials properly e.g. posting passwords on post-it stick- 

ers

T124-Delegation  Authentication delegation models in which the delegate assumes the user’s original ID instead of using their own (e.g. as in Shibboleth see https://spaces.internet2.edu/display/ShibuPortal/Solution+Proposal
F.13 Repudiation threats

T131-Repud Repudiation of events by any party (system entity or user).

F.14 Unauditability threats

T141-AltSig Alteration of signed message (see CR27-Sig and CR28-Vfy)
T142-Tamper Alteration of audit trail (see CR212-Trail)
T143-LeakLogs Eavesdropping on logs (see CR212-Trail)
T144-NoAud Insufficient auditing in the first place

F.15 Software bug threats

T151-Overflow Buffer overflow attack, to gain injection and execution of code, usually leading to compromise of the machine. Also heap overflow.
T152-Inject SQL Query Injection attack, causing database engine to execute unauthorized database statements. This threat also covers similar attacks on other databases or downstream services like shell scripts (command injection).
T153-NI Access control, signature verification, or other security feature not implemented. Testing only positive cases can easily ignore this type of threat. It is imperative that the test suites also include negative testing.
T154-Input Input MUST at all times be validated to conform to the expected syntax, and input in general should never be trusted unless there is a cryptographical or structural guarantee of trustworthiness. Some particular perils

a. Explicit inputs
b. Environment variables
c. URL and query string
d. CGI form fields
e. Cookies
f. HTTP headers
g. SOAP headers
h. Processing contexts of all sorts passed between software modules
i. Fields of certificates (from untrusted source, but you would not know until you manipulated the certificate to find out if it is trusted).
j. Metadata
k. Messages from event bus
F.16. SERVICE AVAILABILITY THREATS

Data coming from database (even if database is supposed to be trusted, it may have been compromised, thus checking for proper format will help to detect the breach and contain it).

Deserialization of any data. This is particularly acute when using eval to deserialize JSON data.

Perl(1) tainted feature provides a way to track untrusted user input. All untrusted input MUST be scrutinized for attack vectors, such as directory climbing ("." or " " = home directory) as a path component. Where relative path is expected, an absolute path - one starting by "/" - MUST NOT be allowed. All shell(1) or SQL escape characters (see also T152-Inject) are of highest suspicion until proven to be secure.

When validating input, preferred strategy is to test if input is good and anything that does not pass is bad. The alternative strategy of looking for known bad things (like ".: " in path) is much weaker and prone to errors.

**T155-Mem** Code MUST NOT

- Dereference a Null Pointer
- Use Freed Memory
- Doubly Free Memory

**T156-Fmt** Format string mismatch. In printf(3) format string it is easy to get the format specifiers and actual arguments mixed, resulting inappropriate format specifier being used for a given piece of data.

**T157-Trunc** Truncation of value. Sometimes truncated value can have different meaning.

**T158-Signed** Signed conversion of a value that may not have been identified.

**T159-CliValid** Reliance on client side validation is no good for server as an alternate client will not perform the validation. Server MUST at all times perform all security critical validations. Client side validation has value in (i) improving user experience and feedback, (ii) reducing network traffic by not submitting obviously invalid inputs, (iii) protecting client side processing like AJAX applications. Such applications MUST be suspicious of what is sent to them by server, in particular if they use JSON and plan to use eval() for processing it.

**T1510-XSiteScript** Cross Site Scripting.

**T1511-Stack** Stack overflow. Similar to T151-Overflow, but specifically applied to the argument and call stack of most compiled programs.

**T1512-HTML** Using non-validated input in the HTML output stream. This could lead into insertion of JavaScript on the generated page.

**T1513-PtrSub** Improper Pointer Subtraction

**T1514-EqEq** Assignment (=) instead of comparison (==).
F.16 Service Availability Threats

**T161-DoS** Denial of Service by massive network load.

**T162-DNS** Poisoning DNS or taking DNS down will prevent legitimate players from communicating with each other.

**T163-Expt** Using program flaw or exception to trigger excessive resource consumption (spinning process, writing all logs full, and leaking memory) or to cause a service to crash.

**T164-GC** Feeding service request pattern (e.g. monotonically increasing input size so that previously freed memory is never enough to satisfy the new request) that causes fragmentation of memory, excessive or ineffective garbage collects, or memory leaks.
Annex G: Enumeration of Audit Events

To understand the wealth of audit trail data we start by enumerating them all:

1. Session Events Channel:
   a. Session creation (possibly even an anonymous session)
   b. Session upgrade (e.g. SSO on an anonymous session, step-up auth)
   c. Session refresh
   d. Session termination
   e. Session expiry
   f. Session revival (if appropriate, could be used as a factor in authentication)

2. User Authentication Events Channel:
   a. Positive
   b. Failure with Retry
   c. Definitive Failure

3. Token Issuing Channel:
   a. Tokens issued with:
      i. Issuer
      ii. Subject
      iii. Audience
      iv. Policy constraints
      v. Validity time and/or usage count
      vi. General content of the token
   b. Token validation at relying party
   c. Token use, to the appropriate extent
   d. Token revocation when applicable

4. Authorization Channel:
   a. Az request parameters
   b. Az decision returned

5. Service Requester Channel:
   a. Choice of Service Provider
      i. Discovery
      ii. Hardwired choice of Service
      iii. Automated or algorithmic Choice of Service
iv. Choice of Service solicited from the User

b. Trust negotiation steps

c. Consent to send data

d. Service Call event

i. Signature preparation, including choice of signing key

ii. Log of content of the message

iii. Peer authentication

iv. Success or failure to send message

e. Service Call exception

i. Redirect or end point change

ii. Recredentialing

iii. Interaction requested

iv. Replay after interaction

v. Dry-run

f. Service Call Response

i. Log of content of the message

ii. Peer authentication (usually by Request-Response pattern)

iii. Success or failure to receive message

g. Service Call Response exception

i. Failures, as detailed on the Faults Channel

ii. Application layer success or failure

h. Obligations processing

i. Presence of obligation

ii. Specific processing steps

iii. Failure to process obligation

6. Service Responder Channel:

a. Trust establishment and trust negotiation steps

b. Request Acceptance

c. Response filtering and authorization decision

d. Attachment of obligations

7. PII Collection Channel

8. PII Release Channel

9. User Registration Channel:

a. Register

b. Modify
c. Deregister

10. SP Registration Channel:
   a. Register
   b. Modify
   c. Change of Control
   d. Deregister

11. User Reputation Channel:
   a. Explicit complaint or praise
   b. Other events that affect reputation

12. Service Reputation Channel:
   a. Explicit complaint or praise
   b. Other events that affect reputation

13. Browsing Event Channel (usually not shared)

14. Faults Channel:
   a. Malformed protocol message
   b. Insufficient sec mech
   c. Signature verification fault
      i. Malformed
      ii. Crypto (public key or hash)
      iii. Certificate validity (missing CA trust chain)
   d. Inappropriate use
      i. Audience
      ii. Constraints
   e. Expired tokens
   f. Replay of message or token
   g. Unsolicited message
   h. Missing database entry
      i. Explicit fault report

15. DoS Channel:
   a. Invocation frequency alert
   b. Data volume alert
   c. Explicit DoS report (e.g. from monitoring organizations)
16. Intrusion Detection System and Firewall ACL Channel:
   a. Scan alert
   b. Attack fingerprint alert
   c. Firewall deny rule triggered

17. Operations monitoring Channel:
   a. Server / Service
      i. Up
      ii. Down
      iii. Scheduled downtime
      iv. Congested
      v. Retry
      vi. Fail Over

18. Audit Operation Channel (very restricted circulation):
   a. Undertaking audits
   b. Outcomes of audit


20. Customer Care Event Channel
Annex H: Example Protocol Messages (Non-Normative)

These XML blobs, taken from [ZXIDREADME], are for reference only. They are not normative. They have been pretty printed. Indentation indicates nesting level and closing tags have been abbreviated as "</>". The actual XML on the wire generally does not have any whitespace.

H.1 SAML 2.0 Artifact Response with SAML 2.0 SSO Assertion and Two Bootstraps

Both bootstraps illustrate SAML assertion as bearer token.

```xml
<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="REvg0I1kzTmk-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">
      <sp:Status>
      </sp:Status>
    </sp:ArtifactResponse>

    <sp:Response
      xmlns:sp="urn:oasis:names:tc:SAML:2.0:protocol"
      ID="RCCzu13z77SiSXqsFp1u1"
      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">
      <sp:Status>
      </sp:Status>
    </sp:Response>
  </soap:Body>
</soap:Envelope>
```
H.1. SAML 2.0 ARTIFACT RESPONSE WITH SAML 2.0 SSO ASSERTION AND TWO
BOOTSTRAPS

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">
  <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:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>r8OvtNmq5LkYwCNg6bsRZAdT4NE="</ds:DigestValue>
      </ds:Reference>
      <ds:SignatureValue>GtWVZzHYW54ioHk/C7zjDRThohrpwC4="</ds:SignatureValue>
    </ds:SignedInfo>
  </ds:Signature>
  <sa:Subject>
    <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:SubjectConfirmationData>
    </sa:SubjectConfirmation>
  </sa:Subject>
</sa:Issuer>

<sa:Conditions>
  NotBefore="2007-02-10T05:32:42Z"
  NotOnOrAfter="2007-02-10T06:37:42Z">
    <sa:AudienceRestriction>
    </sa:AudienceRestriction>
  </sa:Conditions>

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

<sa:Assertion ID="CRED0TGAKvhlNoPlaiTq4xBg" IssueInstant="2007-02-10T05:37:42Z" Version="2.0">
  <sa:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
  <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="#CRED0TGAKvhlNoPlaiTq4xBg">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>r8OvtNmq5LkYwCNg6bsRZAdT4NE="</ds:DigestValue>
      </ds:Reference>
      <ds:SignatureValue>GtWVZzHYW54ioHk/C7zjDRThohrpwC4="</ds:SignatureValue>
    </ds:SignedInfo>
  </ds:Signature>
</sa:Assertion>
H.1. SAML 2.0 ARTIFACT RESPONSE WITH SAML 2.0 SSO ASSERTION AND TWO BOOTSTRAPS

<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="#CREDOTGAkvhNoP1aiTq4bXBg"/>
<ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-sign"/>
<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>dqq/28hw5eEv+ceFyiLImeJ1P8w="/>
<ds:SignatureValue>UKlEgHKQwuoCE="/>
<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:AudienceRestriction>
</sa:Conditions>
<sa:AuthnStatement>
  AuthnInstant="2007-02-10T05:37:42Z"
  SessionIndex="1171085858-4"/>
<sa:AuthnContext>
  <sa:AuthnContextClassRef>
</sa:AuthnContext>
<sa:AttributeStatement>
  <!-- Regular attribute -->
  <sa:Attribute Name="cn" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
    <sa:AttributeValue>Sue</sa:AttributeValue>
  </sa:Attribute>
  <!-- ID-WSF 1.1 Bootstrap for discovery. See also the Advice, above. -->
  <sa:Attribute Name="DiscoveryResourceOffering" NameFormat="urn:liberty:disco:2003-08">
    <sa:AttributeValue>
    </sa:AttributeValue>
  </sa:Attribute>
</sa:AttributeStatement>
<di12:ServiceType>urn:liberty:disco:2003-08</di12>
<di12:Description>
<di12:CredentialRef>CREDOTGAKvNh0PaiTq4bXBg</di12>
</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-wssec
d-version=2.0"
notOnOrAfter="2007-02-10T07:37:42Z"
wsu:Id="EPRIDcjP8Ob09In47SDjO9b37">
</wsa:EndpointReference>
<wsa:Metadata xmlns:id="urn:liberty:disco:2006-08">
<di:Abstract>SYMfiam Discovery Service</di:Abstract>
<di:ServiceType>urn:liberty:disco:2006-08</di:ServiceType>
<di:SecurityContext>
</di:SecurityContext>
<sec:Token
xmlns:sec="urn:liberty:security:2006-08"
usage="urn:liberty:security:tokenusage:2006-08:SecurityToken">
<sa:Assertion
ID="CREDV6ZBMyicmyvDq9pL1oSR"
IssueInstant="2007-02-10T05:37:42Z"
Version="2.0">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#"
<ds:Reference URI="#CREDV6ZBMyicmyvDq9pL1oSR">
<ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
H.2. ID-WSF 2.0 CALL WITH X509V3 SEC MECH

<ds:DigestValue><h2gbuK14e0dQ0TwiyqXr/8Y=</h2></ds:DigestValue>
<ds:SignatureValue><hHDUKa2Jc78UYJxvTReNU=</h2></ds:SignatureValue>
<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">
    <s:3372 9my93vkp3sxEOIb3c9vjlpm0pa6av3yFXioWX-Tz2I=</s:3372
  </sa:NameID>
  <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:AudienceRestriction>
    <sa:AuthnStatement AuthnInstant="2007-02-10T05:37:42Z">
      <sa:AuthnContext>
        <sa:AuthnContextClassRef>
          urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        </sa:AuthnContextClassRef>
      </sa:AuthnContext>
    </sa:AuthnStatement>
  </sa:Conditions>
</sa:Subject>

N.B. The AttributeStatement/Attribute/AttributeValue/EndpointReference/Metadata/
SecurityContext/Token/Assertion/Conditions/AudienceRestriction/Audience is the same
as the IdP because in many products the IdP and Discovery Service roles are implemented by the
same entity. Note also that the audience of the inner assertion is the discovery service where as the
audience of the outer assertion is the SP that will eventually call the Discovery Service.

H.2 ID-WSF 2.0 Call with X509v3 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: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">
  <e:Header>
    <wsa:MessageID wsu:Id="MID">123</wsa:MessageID>
    <wsa:To wsu:Id="TO">...</wsa:To>
    <wsa:Action wsu:Id="ACT">urn:xx:Query</wsa:Action>
    <wsse:Security mustUnderstand="1">
      <wsu:Timestamp wsu:Id="TS">2005-06-17T04:49:17Z</wsu:Timestamp>
      <wsu:SecurityToken anonymous="0">
        <wsu:BinarySecurityToken
          ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
          wsu:Id="X509Token">
          MIIB9zCCAWSgAwIBAgI...</wsu:BinarySecurityToken>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xxmldsig#"/>
H.3. ID-WSF 2.0 CALL WITH BEARER (BINARY) SEC MECH

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 conveys 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).

H.3 ID-WSF 2.0 Call with Bearer (Binary) Sec Mech
H.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:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:xm:="http://www.w3.org/2001/04/xmmlenc#">
  <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</e:Header>
    <wsse:Security mustUnderstand="1">
      <wsu:Timestamp wsu:Id="TS">2005-06-17T04:49:17Z</wsu:Timestamp>
      <wsu:Created>2005-06-17T04:49:17Z</wsu:Created>
    </wsse:Security>
    <sa:Assertion xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
Version="2.0"
ID="A7N123"/>
<sa:Issuer>http://idp.symdemo.com/idp.xml</>
<ds:Signature>...</>
<sa:Subject>...
<sa:EncryptedID>
<xenc:EncryptedData>U2XTCNvRX7B11NK182nmY0OTEk==</xenc:EncryptedData>
</sa:EncryptedID>
<sa:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
<sa:Conditions>
  NotBefore="2005-04-01T16:57:20Z"
  NotOnOrAfter="2005-04-01T21:42:43Z">
  <sa:AudienceRestrictionCondition>
  </sa:AudienceRestrictionCondition>
  <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:AuthnContextClassRef>
    </sa:AuthnContext>
  </sa:AuthnStatement>
  <sa:AttributeStatement>
    <sa:EncryptedAttribute>
      <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
        mQEMAzRniWkAAAEH9RbzqXdgcX8fpEqSr1v4==
      </xenc:EncryptedData>
    </sa:EncryptedAttribute>
  </sa:AttributeStatement>
  <wsse:SecurityTokenReference xmlns:wsse11="..."
    wsu:Id="STR1"
    wsse11: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</wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
</sa:Conditions>
<ds:Signature>
<ds:SignedInfo>
  <ds:Reference URI="#MID">...<ds:Reference URI="#T0">...<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"></ds:Reference>
      </ds:CanonicalizationMethod>
    </wsse:TransformationParameters>
    <ds:Reference URI="#BDY"></ds:Reference>
  </ds:Transform>
</ds:Reference>
<e:Body wsu:Id="BDY">
  <xx:Query/></xx:Query></e:Body>
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.
Annex I: Glossary

**Abstract Business Process**  Abstract business processes are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete operational details. Abstract Processes serve a descriptive role, with more than one possible use case, including observable behavior and process template.

- **Source:** Quentin (Quentin.Reul@vub.ac.be)

**Accessibility**  Accessibility means any condition, disease or disability that requires special employment measures or is eligible for positive action.

- **Source:** Ingo (dahn@uni-koblenz.de)

**APL**  See Accreditation of Prior Learning

**Accreditation of Prior Learning**

- **Source:** Dries (dries.pruis@kenteq.nl)

**Activity**  An activities an agent wishes to undertake in order to fulfill his/her goals.

- **Source:** Ingo (dahn@uni-koblenz.de)

**Action**  An operation on a resource.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** PERMIS Glossary

**Address**  An address is the identifier for a specific termination point and is used for routing to this termination point.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** ITU-T Y.2091 - Terms and definitions for Next Generation Networks

**Affiliation**  Membership of learned, professional, civic and recreational organisations.

- **Source:** Ingo (dahn@uni-koblenz.de)

**Agent**  A person (or service) entitled to act on behalf of another.

- **Source:** Ingo (dahn@uni-koblenz.de)

**Anonymity**  Ability to allow anonymous access to services, which avoid tracking of user's personal information and user behaviour such as user location, frequency of a service usage, and so on.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** ITU-T X.1121 (04), 3.2.1
**ADPDP**  See Application Dependent PDP

**Application Dependent PDP**  Application Dependent PDP. Apply specific rules that relate to the application roles. Typically communicates with ADPEP, but may also proxy requests in relevant special cases to outside PDPs or gather Information for its decisions from outside, including from Reputation Providers.

- Source: David (d.w.chadwick@KENT.AC.UK)

**ADPEP**  See Application Dependent PEP

**Application Dependent PEP**  Apply specific rules that relate to the application roles. Typically communicates with ADPDP.

- Source: David (d.w.chadwick@KENT.AC.UK)

**AIPDP**  See Application Independent PDP

**Application Independent PDP**  Application Independent PDP, more properly TAS3 Network PDP or External PDP Aggregator (cf. Architecture: Anatomy of PEP)

- Source: David (d.w.chadwick@KENT.AC.UK)

**AIPEP**  See Application Independent PEP

**Application Independent PEP**  Application Independent PEP, typically communicates with AIIPDP (cf. Architecture: Anatomy of PEP)

- Source: David (d.w.chadwick@KENT.AC.UK)

**AP**  See Asserting Party

**Asserting Party**  *** TBD

**Assertion**  A collection of one or more statements about an entity (e.g. Authentication statement or Authorisation statement).

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: OMA - The Open Mobile Alliance

**Asset**  Anything that has value to the organization, its business, its operations and its continuity.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.2701 - Security requirements for NGN release 1

**Assurance Level**  A quantitative expression of Authentication Assurance agreed between a Relying Party and an Identity Provider.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec
**Asymmetric Authentication Method**  A method of authentication, in which not all authentication information is shared by both entities.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** ITU-T Y.IdMsec, X.811

**Attribute**  A distinct characteristic of an object. An object’s attributes are said to describe the object. Objects’ attributes are often specified in terms of their physical traits, such as size, shape, weight, and color, etc., for real-world objects. Objects in cyberspace might have attributes describing size, type of encoding, network address, etc.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** WSIA Glossary - Glossary for the OASIS WebService Interactive Applications (WSIA/WSRP)

**AA**  See Attribute Authority

**Attribute Authority**  Trusted authorities, which assign roles to users. Normally this is also done by the SOA.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** PERMIS Glossary

**AAPML**  See Attribute Authority Policy Markup Language

**Attribute Authority Policy Markup Language**  AAPML is a XACML profile designed to allow attribute authorities to specify conditions under which information under management may be used (and possibly modified) by other applications.

- **Source:** Liberty Alliance Project

**AC**  See Attribute Certificate

**Attribute Certificate**  Attributes that are certified (digitally signed) by an Attribute Authority as belonging to a particular object. As an analogy, if a PKC corresponds to a passport, an AC corresponds to a visa.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** PERMIS Glossary

**ACRL**  See Attribute Certificate Revocation List

**Attribute Certificate Revocation List**  List of revoked ACs issued by and AA.

- **Source:** David (d.w.chadwick@KENT.AC.UK)
- **Reference:** PERMIS Glossary
Attribute Type That component of an attribute which indicates the class of information given by that attribute.

- Source: David (d.w.chadwick@KENT.AC.UK)

Attribute Value A particular instance of the class of information indicated by an attribute type.

- Source: David (d.w.chadwick@KENT.AC.UK)

Audit An independent review and examination of system records and activities in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy and procedures.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T X.800 - Security architecture for Open Systems Interconnection for CCITT applications

Audition Aiming at providing only high quality service to the users, the provider of a directory service can be interested in testing that the services asking for registration are of "good" quality. For this purpose, the directory could submit the service under registration to a verification step before granting the registration. The implementation of such process with respect to the technical assessment is called Audition (Automatic Model-Based Interface Testing In Open Networks).

- Source: Antonia (antonia.bertolino@isti.cnr.it)

Authenticated Identity A distinguishing identifier of an entity that has been assured through authentication.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.811

Authn See Authentication

Authentication The provision of assurance of the claimed identity of an entity.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.811

Authentication Certificate A security certificate that is guaranteed by an authentication authority and that may be used to assure the identity of an entity.

- Source: David (d.w.chadwick@KENT.AC.UK)
• Reference: ITU-T Y.IdMsec, X.811

**Authentication Exchange**  A sequence of one or more transfers of exchange authentication information (AI) for the purposes of performing an authentication.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.811

**Authentication Information**  Information used to establish the validity of a claimed identity.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.800

**Authentication Initiator**  The entity that starts an authentication exchange.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.811

**Authentication Insurance**  A measure of confidence that the security features and architecture of the Identity Management capabilities accurately mediate and enforce the security policies understood between the Relying Party and the Identity Provider.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec

**Authorisation**  The granting of rights, which includes the granting of access based on access rights.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.800

**Authorization Decision**  The result of evaluating applicable policy, returned by the PDP to the PEP. A function that evaluates to "Permit", "Deny", "Indeterminate" or "NotApplicable", and (optionally) a set of obligations

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

**Authoritative**  In the context of IdM, the Identity Provider which possesses the authority under law, contractual agreement, or customary practice to definitively answer queries concerning a specific identity for which it is responsible.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec

**Authority Number**  Agents may receive an authority number from a registered authority to be uniquely identified within the authority’s jurisdiction or system. Among authority numbers are included: social security numbers, enrollment numbers, etc. An authority number typically consists of a name, a scheme and a code.
• Source: Ingo (dahn@uni-koblenz.de)

Authority Provider  A provider of authentication numbers, the uniqueness of numbers and their meaning are defined by the authority provider's jurisdiction or system. Examples are governments, who can supply social security numbers, driving license numbers etc.

• Source: Ingo (dahn@uni-koblenz.de)

Availability  Availability is the goal that assets have to be available to authenticated and authorised agents when needed.

• Source: Quentin (Quentin.Reul@vub.ac.be)

Behavioural Factors  Aspects of feedback used in defining a reputation. For example for a helpdesk one could consider politeness, responsiveness, usefulness of supplied information, etc. These factors may be combined into the reputation differently depending on the needs of the user.

• Source: Sampo (sampo@symlabs.com)

BTM  See Behavioural Trust Management

Behavioural Trust Management  Class of trust management systems that use information on past performance to build trust. RTM and KPITM are examples.

• Source: Jerry (j.d.hartog@tue.nl)

BGP  See Breaking-the-glass Policy

Breaking-the-glass Policy  A term used to describe an access control policy that allows users who would not normally have access to a resource, to gain access themselves by "breaking the glass" in the full knowledge that they will have to answer for their actions later to their management.

• Source: David (d.w.chadwick@KENT.AC.UK)

BMO  See Business Management Ontology

Business Management Ontology  The Business Management Ontology (BMO) represents an integrated information model, which helps to better align IT with business. It brings together business process design, project management, requirements management, and business performance management (in the form of balanced scorecards). As such, it forms the basis for an integrated, vendor-neutral, Business Management Knowledge Base, from which various artifacts can be generated.

• Source: Quentin (Quentin.Reul@vub.ac.be)
• Reference: Ontology-Based Business Process Management - The Vision Statement

Business Process  *** TBD
**Business Process Engine**  The Business Process Engine orchestrates entities that control how FEs and SPs work together to achieve the objectives of the business process.

- Source: Sampo (sampo@symlabs.com)

**BPEL**  See Business Process Execution Language

**Business Process Execution Language**  WS-BPEL provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

- Source: Jutta (muelle@ipd.uka.de)
- Reference: Web Services Business Process Execution Language Version 2.0

**BPEL4People**  See Business Process Execution Language Extension for People

**Business Process Execution Language Extension for People**  BPEL4People addresses human interactions in BPEL. It defines a new type of basic activity which uses human tasks as an implementation, and allows specifying tasks local to a process or use tasks defined outside of the process definition.

- Source: Jutta (muelle@ipd.uka.de)
- Reference: WS-BPEL Extension for People (BPEL4People), Version 1.0

**BPM**  See Business Process Modelling

**Business Process Modelling**  Using a formal methodology to describe a business process. Such formal model will usually allow some of the configuration details for implementing the business model to be automatically derived.

- Source: Sampo (sampo@symlabs.com)

**BPMN**  See Business Process Modeling Notation

**Business Process Modeling Notation**  The Business Process Modeling Notation (BPMN) is a graphical notation that depicts the steps in a business process. BPMN depicts the end to end flow of a business process. The notation has been specifically designed to coordinate the sequence of processes and the messages that flow between different process participants in a related set of activities.

- Source: Quentin (Quentin.Reul@vub.ac.be)
- Reference: http://www.bpmn.org/

**BPO**  See Business Process Ontology

**Business Process Ontology**  *** TBD

- Source: Quentin (Quentin.Reul@vub.ac.be)
Certificate  A set of security-relevant data issued by a security authority or a trusted third party, together with security information which is used to provide the integrity and data origin authentication services for the data.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T X.800 - Security architecture for Open Systems Interconnection for CCITT applications

CA  See Certification Authority

Certification Authority  Issues digital certificates.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

Choreography  A choreography description is a multi-party contract that describes from global viewpoint the external observable behavior across multiple clients (which are generally Web Services but not exclusively so) in which external observable behavior is defined as the presence or absence of messages that are exchanged between a Web Service and it’s clients.

- Source: Guglielmo (guglielmo.deangelis@isti.cnr.it)

CoT  See Circle of Trust

Circle of Trust  See Trust Network.

- Source: Sampo (sampo@symlabs.com)

Claim  An assertion made by a Claimant of the value or values of one or more Identity Attributes of a Digital Subject, typically an assertion which is disputed or in doubt.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Gang of Identity Commons

Claim Authentication Information  Information used by a claimant to generate exchange AI needed to authenticate an entity.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.811

Client  While general meaning as in “customer” is acknowledged, in protocol contexts “Client” is taken to mean requester of a service. Thus Client is the counter part of a Service Provider. Client is a business entity and quite different from a User. A Service Provider can be a Client towards other entities that it calls.

- Source: Sampo (sampo@symlabs.com)

CARML  See Client Attribute Requirements Markup Language
Client Attribute Requirements Markup Language  
Client Attribute Requirements Markup Language is a specification that allows applications to define their attribute requirements as it relates to identity. CARML can be used to automate configuration of identity attribute services and to expose the set of identity-related data consumed by a specific application or groups of applications.

- Source: Liberty Alliance Project

Competency  
A competency is a demonstrated ability of a natural person to apply knowledge, skills and attitudes to achieve observable results.

- Source: Ingo (dahn@uni-koblenz.de)

Confidentiality  
Confidentiality is the goal that data should be readable to agents with appropriate permissions.

- Source: Quentin (Quentin.Reul@vub.ac.be)

Context  
A property that can be associated with a user attribute value to specify information that can be used to determine the applicability of the value.

- Source: David (d.w.chadwick@KENT.AC.UK)

Credential  
Authentication and Authorization data that can be used to authenticate the claimer is what it claims to be and authorize the claimer's access rights. What AA needs from the SOA to be able to issue ACs.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ETSI TS 132 372 V7.0.0
- Comments: CONFLICT:

CTM  
See Credential based Trust Management

Credential based Trust Management  
System which builds trust on structural rules which are exchanged in the form of credentials.

- Source: Jerry (j.d.hartog@tue.nl)

Credential Chain  
A tree (or sequence) of credentials which ensures trustworthiness of the statement in the root credential. Each node is validated by its children and the leaf credentials are issued by trusted entities (e.g. AA).

- Source: Jerry (j.d.hartog@tue.nl)
CIS  See Credential Issuing Service

Credential Issuing Service  The service of issuing a digitally signed attribute assertions provided by an authoritative source of subject attributes

  * Source: David (d.w.chadwick@KENT.AC.UK)

CVS  See Credential Validation Service

Credential Validation Service  The service of validating digitally signed attribute assertions and determining which are trusted and which are not.

  * Source: David (d.w.chadwick@KENT.AC.UK)
  * Reference: PERMIS Glossary

Data Origin Authentication  Corroboration that the source of data received is as claimed.

  * Source: David (d.w.chadwick@KENT.AC.UK)
  * Reference: ITU-T Y.1IdMsec, X.800

Data Protection  *** TBD

  * Source: Joseph (joseph.alhadeff@ORACLE.COM)

DB  See Dashboard

Dashboard  A web GUI for viewing audit records, work flow status, and/or viewing and manipulating privacy settings and permissions.

  * Source: Sampo (sampo@symlabs.com)

DTM  See Decentralised Trust Management

Decentralised Trust Management  DEPRECATED - see CTM

  * Source: Jerry (j.d.hartog@tue.nl)

Decision Request  The request by a PEP to a PDP to render an authorization decision

  * Source: David (d.w.chadwick@KENT.AC.UK)
  * Reference: PERMIS Glossary

Delegatee  The entity receiving a privilege though a delegation.

Delegation  Conveyance of privilege from one entity that holds such privilege, to another entity.

  * Source: David (d.w.chadwick@KENT.AC.UK)
  * Reference: ITU-T X.509 (00), 3.3.46 - Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks
Delegator  The entity that holds and conveys a privilege to another entity though a delegation.

Digital Identity  The digital representation of the information known about a specific individual, group or organization

- Source: David (d.w.chadwick@KENT.AC.UK)

Digital Identity Provider  An Agent that issues a Digital Identity.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Gang of Identity Commons

Digital Subject  An Entity represented or existing in the digital realm which is being described or dealt with.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Gang of Identity Commons

Directed Identity  A unifying identity metasystem must support both "omni-directional" identifiers for public entities and "unidirectional" identifiers for private entities

- Source: David (d.w.chadwick@KENT.AC.UK)

Discovery  Finding entities/services/objects matching a set of criteria, e.g. Service Discovery.

- Source: Jerry (j.d.hartog@tue.nl)

DNS  See Domain Name System

Domain Name System  The scheme for attributing alphanumeric, human readable "web addresses". DNS will map the human readable string to an IP address. Sometimes a /etc/hosts file replaces the function of the DNS, but this solution, while allowing more local control, is generally very burdensome to maintain.

Electronic Identity  The information about a registered entity, that the Identity Provider has chosen to represent the Identity of that entity. The eID includes a name or an identifier for the entity that is unique within the domain of the Identity Provider.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: TF-AACE - Terena Authentication and Authorisation

Employability  *** TBD

- Source: Dries (dries.pruis@kenteq.nl)

Enrolment  The process of adding a Permission to an Identity.
Entity  Anything that has separate and distinct existence that can be uniquely identified. In the context of IdM, examples of entities include subscribers, users, network elements, networks, software applications, services and devices. An entity may have multiple identifiers.


XACML  See eXtensible Access Control Markup Language

eXtensible Access Control Markup Language  The OASIS Extensible Access Control Markup Language (XACML) TC was chartered “to define a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML. There are many proprietary or application-specific access control policy languages, but this means policies cannot be shared across different applications, and provides little incentive to develop good policy composition tools. XACML enables the use of arbitrary attributes in policies, role-based access control, security labels, time/date-based policies, indexable policies, ‘deny’ policies, and dynamic policies - all without requiring changes to the applications that use XACML.”

Federation  A federation is a collection of realms that have established a producer-consumer relationship whereby one realm can provide authorized access to a resource it manages based on an identity, and possibly associated attributes, that are asserted in another realm. A federation requires trust such that a Relying Party can make a well-informed access control decision based on the credibility of identity and attribute data that is vouched for by another realm.

Federated Identity  A single user identity that can be used to access a group of services or applications that are bounded by the ties and conditions of a federation.

FIM  See Federated Identity Management
**Federated Identity Management**  The communal services provided by a group of organisations which have set up trust relationships between themselves, so that they can send each other digitally signed attribute assertions about their users’ identities in order to grant each others’ users access to their resources.

- Source: David (d.w.chadwick@KENT.AC.UK)

**FE**  See Front-end

**Front-end**  In this context, it means web site, i.e. SP

- Source: Sampo (sampo@symlabs.com)

**GA**  See Governing Agreement

**Governing Agreement**  Legal document that every member of Trust Network MUST agree to. This can be seen as the charter of the Trust Network.

- Source: Sampo (sampo@symlabs.com)

**Identification**  The process of verifying the identity of a user, process, or device, usually as a prerequisite for granting access to resources in an IT system.

- Source: David (d.w.chadwick@KENT.AC.UK)

**Identifier**  An identifier is a series of digits, characters and symbols or any other form of data used to identify subscriber(s), user(s), network element(s), function(s), network entity(ies) providing services/applications, or other entities (e.g., physical or logical objects).

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.2091 - Terms and definitions for Next Generation Networks

**Identity**  An identity is a uniquely identifiable representation of an agent. An agent can have multiple identities that do not necessarily interrelate.

- Source: Ingo (dahn@uni-koblenz.de)

**IAF**  See Identity Assurance Framework

**Identity Assurance Framework**  *** TBD

- Source: Liberty Alliance Project

**Identity Agent**  It manages and supports a consistent user experience (and in some cases other kinds of interactions) with a Service Provider.
Identity Attribute  A property of a Digital Subject that may have zero or more values.

Identity Availability  The availability of an identity gives an indication of the how much an agent can spend on a particular job.

Identity Based Security Policy  A security policy based on the identities and/or attributes of users, a group of users, or entities acting on behalf of the users and the resources/objects being accessed.

Identity Context  The surrounding environment and circumstances that determine meaning of Digital Identities and the policies and protocols that govern their interactions.

IGF  See Identity Governance Framework

Identity Governance Framework  It is an open initiative to address governance of identity related information across enterprise IT systems. This initiative includes key initial draft specifications contributed by Oracle to the community. These specifications provide a common framework for defining usage policies, attribute requirements, and developer APIs pertaining to the use of identity related information. These enable businesses to ensure full documentation, control, and auditing regarding the use, storage, and propagation of identity-related data across systems and applications.

Identity Information  All the information identifying a user, including trusted (network generated) and/or untrusted (user generated) addresses. Identity information shall take the form of either a SIP URI (see RFC 2396) or a "tel" URI (see RFC 3966).
Identity Layer  An identity layer attempts to develop convergence and interoperability regarding identity, can draw from multiple data stores, selectively exposing, or concealing data and attributes, according to policy.

- **Source**: David (d.w.chadwick@KENT.AC.UK)

IdM  See Identity Management

Identity Management  The management by trusted providers of trusted attributes of an entity such as: a subscriber, a device or a provider.

- **Source**: David (d.w.chadwick@KENT.AC.UK)
- **Reference**: ITU-T Y.IdMsec

IdMAA  See Identity Management, Authentication and Authorisation Infrastructure

Identity Management, Authentication and Authorisation Infrastructure  Application independent middleware responsible for authenticating and authorizing entities.

- **Source**: David (d.w.chadwick@KENT.AC.UK)

Identity Pattern  A structured expression derived from the behaviour of an entity that contributes to the recognition process; this may include the reputation of the entity. Identity patterns may be uniquely associated with an entity, or a class with which the entity is associated.

- **Source**: David (d.w.chadwick@KENT.AC.UK)
- **Reference**: FG IdM Use Case WG - ITU-T Focus Group for Identity Management

IdP  See Identity Provider

Identity Provider  An entity that specializes in identifying (collecting identity information or PII), and authenticating users. IdP is usually, and in SAML case especially, charged with the role of facilitating Single Sign On (SSO). IdP often with the role of facilitating Single Sign On (SSO). IdP often also conveys PII when authenticating the User. IdP has prime visibility to the usage patterns of a User and is therefore especially vulnerable or in need of special business or administrative protections. IdP function is often associated with ID Service Discover and Token Mapping functions. Core of an IdP is a federation database where mappings between several pseudonymous identities and relationships with the service providers are evident. This database constitutes a fat target when an identity system is attached.

- **Source**: Sampo (sampo@symlabs.com)
- **Comments**: CONFLICT:

Identity Registration  The process of making a person's identity known to the (Personal Identity Verification) system, associating a unique identifier with that identity, and collecting and recording the person's relevant attributes into the system.

- **Source**: David (d.w.chadwick@KENT.AC.UK)
Inactive Status  A service is inactive if it needs to use services that are not yet available according to the Registry.

Integrity  Integrity is the goal that data and information should not be altered if not explicitly allowed.

- Source: Quentin (Quentin.Reul@vub.ac.be)

IDL  See Interface Description Language

Interface Description Language  For example within the standards of the family WS*, WSDL is an IDL.

- Source: Sampo (sampo@symlabs.com)

Interoperability  Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged. In particular, it envisages the ability for loosely-coupled independent systems to be able to collaborate and communicate; the possibility of use in services outside the direct control of the issuing assigner.

- Source: Quentin (Quentin.Reul@vub.ac.be)

KPI  See Key Performance Indicator

Key Performance Indicator  Key Performance Indicators are combinations of different Business Performance factors such as Time to deliver, or number of patent application, etc.

- Source: Sampo (sampo@symlabs.com)

KPITM  See Key Performance Indicator Trust Management

Key Performance Indicator Trust Management  System which builds trust on economical factors such as performance on delivery times, number of patents filed, etc.

- Source: Jerry (j.d.hartog@tue.nl)

Layer Network  A "topological component" that represents the complete set of access groups of the same type which may be associated for the purpose of transferring information.

- Source: David (d.w.chadwick@KENT.AC.UK)
Level of Assurance  A metric which is used to measure the confidence (or assurance) that a relying party can have, that an authenticated user is really who they say they are. One scale, devised by the US National Institute of Science and Technology, ranges from 1 to 4, with 4 being the highest.  

- Source: David (d.w.chadwick@KENT.AC.UK)

LDAP  See Lightweight Directory Access Protocol

Lightweight Directory Access Protocol  *** TBD

- Source: Jutta (muelle@ipd.uka.de)

LTS  See Long Tail Service

Long Tail Service  It means that half of the volume of the internet use can be in myriad of low use services (the other half is in few high volume services).

- Source: Sampo (sampo@symlabs.com)

MS  See Message Signer

Message Signer  Digitally signs request.

- Source: Danny (decockd@esat.kuleuven.be)

MV  See Message Verifier

Message Verifier  Verifies digital signature and other constraints of a request.

- Source: Danny (decockd@esat.kuleuven.be)

NOC  See Network Operation Center

Network Operation Center  *** TBD

Non-Repudiation  The ability through historical logs and logical analysis to prevent or discourage an Entity from denying that it had acted as an Identity in a given transaction, especially in a legal sense.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Dictionary

Object  A well-defined piece of information, definition, or specification which requires a name in order to identify its use in an instance of communication and identity management processing.

- Source: David (d.w.chadwick@KENT.AC.UK)
Obligation  An operation specified in a policy that should be performed by the PEP in conjunction with the enforcement of an authorization decision

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

Offline Testing  Testing phase that includes the activities that are performed while no user ("paying customer") is using the service. Hence, off-line validation of a system implies that it is tested in one or more artificially evolving environments that simulate possible real interactions.

- Source: Seda (sguerses@ESAT.KULEUVEN.BE)

Online Testing  Testing phase that concerns a set of methodologies, techniques, and tools to monitor a system after its deployment in its real working context.

- Source: Seda (sguerses@ESAT.KULEUVEN.BE)

Ontology  an ontology is commonly defined as: "a [?, ?] conceptualization" (Gruber, 1993). More specifically, an ontology explicitly defines a set of entities (e.g. classes, relations and individuals) imposing a structure on the domain that is readable by both humans and machines.

- Source: Quentin (Quentin.Reul@vub.ac.be)

Orchestration  The process of coordinating the sequence and data flow during (web) services interaction.

- Source: Jutta (muelle@ipd.uka.de)

Owner  The registered Entity for an Identity.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Dictionary

Panopticon threat  A especially pertinent risk in running a Trust Guarantor is that it may gain excessive knowledge to the operations of the Service Provider members or the Users and their business processes. It can be mitigated by careful division of responsibilities using externally contracted Trusted Third Parties, each of which operates in its own isolated, regulatory scheme.

Pending Status  A service is in a pending status if it is registered to a directory service, but has not yet been tested by Audition.

Persistent  Existing, and able to be used in services outside the direct control of the issuing assigner, without a stated time limit.

- Source: David (d.w.chadwick@KENT.AC.UK)
PCP  See Personal Competency Profile

Personal Competency Profile  *** TBD

PDS  See Personal Data Store

Personal Data Store  *** TBD

  ●  Source: Luk (luk@synergetics.be)

PII  See Personally Identifying Information

Personally Identifying Information  Information that may allow identifying a User, or impersonation of the User.

  ●  Source: Sampo (sampo@symlabs.com)

PUPPET  See Pick UP Performance Evaluation Test-bed

Pick UP Performance Evaluation Test-bed  It is an approach for the automatic generation of test-beds to empirically evaluate the QoS characteristics of a Web Service under development. Specifically, the generation exploits the information about the coordinating scenario, the service description (WSDL) and the specification of the agreed QoS properties.

  ●  Source: Sampo (sampo@symlabs.com)

Policy  A set of rules, an identifier for the rule-combining algorithm and (optionally) a set of obligations. May be a component of a policy set.

  ●  Source: David (d.w.chadwick@KENT.AC.UK)
  ●  Reference: PERMIS Glossary

PAP  See Policy Administration Point

Policy Administration Point  *** TBD

  ●  Source: David (d.w.chadwick@KENT.AC.UK)

PDP  See Policy Decision Point

Policy Decision Point  The (application independent) part of an access control system that can answer access control requests with a granted or denied decision.

  ●  Source: David (d.w.chadwick@KENT.AC.UK)
  ●  Reference: PERMIS Glossary

PEP  See Policy Enforcement Point
Policy Enforcement Point  The (application dependent) part of an access control system that is responsible for enforcing the authorization decisions returned by the PDP.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

PIP  See Policy Information Point

Policy Information Point  *** TBD

- Source: Sampo (sampo@symlabs.com)

PMS  See Policy Management Service

Policy Management Service  Handles the management of user policies and ‘organization wide’ policies. Moreover it will have a functionality to attach policies to a request respectively a response. This is an ongoing task in WP8 under the name of ‘Aggregating Policies’.

- Source: Sampo (sampo@symlabs.com)

Principal  See Entity.

- Source: Jerry (j.d.hartog@tue.nl)

Privacy  *** TBD

- Source: Joseph (joseph.alhadeff@ORACLE.COM)

Privacy Policy  A set of rules and practices that specify or regulate how a person or organization collects, processes (uses) and discloses another party's personal data as a result of an interaction.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: W3C Glossary and Dictionary

Private Identifier  A Claimed Identifier that is intended to be private information used only the context of the End User's relationship with one or more specific Relying Parties (typically one or a small number). The use of Private Identifiers reduces or eliminates the ability of multiple Relying Parties to do correlation of an End User.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: OpenID

Privilege  A right to carry out a particular permission (act) that is assigned to a role with some constraints or conditions. A role is (can be) associated with multiple privileges.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: FG IdM Use Case WG - ITU-T Focus Group for Identity Management
**PMI**  See Privilege Management Infrastructure

**Privilege Management Infrastructure**  A highly scalable infrastructure, based on digitally signed attribute assertions, which allows subjects to be authorised to use the resources of relying parties based on their mutual trust in Attribute Authorities. A component of FIM.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

**PVS**  See Privilege Verification Subsystem

**Privilege Verification Subsystem**  Decision Engine consisting of PEP and PDP.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

**PMF**  See Process Modelling Framework

**Process Modelling Framework**  *** TBD

- Source: Intalio

**Provisioning**  Automatically providing an Identity with access to a role, resource or service, or automatically changing or removing that access, based on the life cycle of events or work requests or changed attributes.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Dictionary

**Pseudonym**  A fictitious identity that an Entity creates for itself, whereby the Entity can remain pseudonymous, or perhaps even fully anonymous, in certain contexts.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Dictionary

**Pseudonymity**  See Pseudonym.

**PKC**  See Public Key Certificate

**Public Key Certificate**  An electronic document that using a digital signature binds together a public key and an identity. As an analogy, if an AC corresponds to a visa, a PKC corresponds to a passport.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

**PKI**  See Public Key Infrastructure
Public Key Infrastructure  A highly scalable infrastructure, based on public key cryptography, which allows subjects to authenticate to relying parties based on their mutual trust in Public Key Certification Authorities (a type of TTP). A component of FIM.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

Public Private Partnership  *** TBD

- Source: Luk (luk@synergetics.be)

QoS  See Quality Of Service

Quality Of Service  *** TBD

RTM  See Real-Time Trust Management

Real-Time Trust Management  DEPRECATED

- Source: Jerry (j.d.hartog@tue.nl)

Registry  *** TBD

RP  See Relying Party

Relying Party  A Party that makes known through its Agent one or more alternative sets of Claims that it desires or requires, and receives through this same Agent a Digital Identity purportedly including the required Claims from a Digital Identity Provider or other Agent of another Party.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: Identity Dictionary

Repository  *** TBD

Repudiation  Denial by one of the entities involved in a communication of having participated in all or part of the communication

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: X.800 (91), 3.3.44

Reputation  The reputation of an entity is the view on that entity based on its past performance. Reputations are computed based on recommendations and on feedback on interaction with the entity.

- Source: Jerry (j.d.hartog@tue.nl)

RTM  See Reputation based Trust Management

Reputation based Trust Management  System which builds trust on past performance expressed in feedback and recommendation.
• Source: Jerry (j.d.hartog@tue.nl)
  Comments: CONFLICT: Same acronym as Real Time Trust

PDP-R  See Requester Policy Decision Point

Requester Policy Decision Point   *** TBD
  Source: David (d.w.chadwick@KENT.AC.UK)

PEP-R  See Requester Policy Enforcement Point

Requester Policy Enforcement Point   *** TBD
  Source: David (d.w.chadwick@KENT.AC.UK)

Resource  Data, service or system component
  Source: David (d.w.chadwick@KENT.AC.UK)
  Reference: PERMIS Glossary

RS  See Response Signer

Response Signer  Digitally signs request
  Source: Danny (decockd@esat.kuleuven.be)
  Comments: CONFLICT: Shouldn't it be 'response' instead of 'request'.

RV  See Response Verifier

Response Verifier  Verifies digital signature and other constraints of a response.
  Source: Danny (decockd@esat.kuleuven.be)

Risk  A Risk is defined as a triplet consisting of a targeted model element, a related security requirement and a threat that potentially undermines the requirement, including an assessment of its severity. Moreover, every risk is evaluated in the context of the currently implemented security controls.
  Source: Quentin (Quentin.Reul@vub.ac.be)

Role  Type of attribute that is typically used to signify the position that someone has in an organisation.
  Source: David (d.w.chadwick@KENT.AC.UK)
  Reference: PERMIS Glossary

RBAC  See Role Based Access Control
Role Based Access Control  A model for controlling access to resources where permitted actions on resources are identified with roles rather than with individual subject identities.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

S&T  See Science and Technology

Science and Technology  *** TBD

SAML  See Security Assertion Markup Language

Security Assertion Markup Language  It is an XML-based framework for communicating user authentication, entitlement, and attribute information. As its name suggests, SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application.

- Source: Quentin (Quentin.Reul@vub.ac.be)

Security Control  A Security Control is any managerial, operational, and/or technical measure or safeguard that has been put in place to mitigate identified risks.

- Source: Quentin (Quentin.Reul@vub.ac.be)

Security Domain  A set of elements, a security policy, a security authority and a set of security-relevant activities in which the elements are managed in accordance with the security policy. The policy will be administered by the security authority. A given security domain may span multiple security zones.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.2701 - Security requirements for NGN release 1

Security Domain Authority  A security authority that is responsible for the implementation of a security policy for a security domain.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.810

SO  See Security Officer

Security Officer  A job function or role at Trust Guarantor. Similar function, with the same name, may also exist at Trusted Third Parties, and Service Providers. Security Officer’s job is to on continuing basis verify and validate that the members of a Trust Network adhere to the rules. To do this Security Officer usually operates and monitors automated auditing and systems monitoring tools. If discrepancies are found, or complaints are reported, the Security Officer will
investigate manually in more detail. Security Officer also participates in approving new members to the network and in taking disciplinary action, such as removal from the network, against the offenders.

**Security Objective**  A Security Objective describes a general security goal to the system. Security Objectives in many cases originate in legal requirements and general availability, integrity and confidentiality requirements.

- **Source:** Quentin (Quentin.Reul@vub.ac.be)

**Security Policies**  A Security Policy realises a specific Security Objective (or a combination thereof). A Security Policy is defined as a statement of what is and what is not allowed.

- **Source:** Quentin (Quentin.Reul@vub.ac.be)

**Security Requirement**  A Security Requirement is a detailed context-dependent explanation of a Security Objective. It breaks security objectives down in several more detailed descriptions. The context of a Security Requirement is derived from the model element for which it is defined.

- **Source:** Quentin (Quentin.Reul@vub.ac.be)

**Semantics**  Semantics provide a (semi-)formal meaning to concepts in a domain of discourse. It allows computer programs and humans to understand what is meant by a concept.

- **Source:** Quentin (Quentin.Reul@vub.ac.be)

**SoD**  See Separation of Duties

**Separation of Duties**  A security procedure whereby a high risk task is split into at least two sub-tasks which have to be carried out by different people.

- **Source:** David (d.w.chadwick@KENT.AC.UK)

**Disco**  See Service discovery

**Service discovery**  Service discovery, sometimes specifically identity enabled service discovery such as Liberty ID-WSF Discovery Service. Discovery service corresponds to one of the bulletin boards in Danny’s “snake” diagram.

- **Source:** Sampo (sampo@symlabs.com)
- **Comments:** CONFLICT:

**SOA**  See Service Oriented Architecture
**Service Oriented Architecture**  A conglomeration of web services, or in a broader sense any kind of
services. SOA paradigm attempts to abstract the services so that they are reusable components
that can be composed in different arrangements at will. Parallel to the orchestration, there is
identity propagation infrastructure and authorization infrastructure, which in its turn relies on
trust infrastructure. Real life SOAs are much less generic and recomposing the components in
any reliable way remains a dream.

- Source: Sampo (sampo@symlabs.com)

**SP**  See Service Provider

**Service Provider**  An entity that provides a kind of electronic service to users. In TAS3 context the
service is foreseen to be provided over a network, usually the Internet.

- Source: Sampo (sampo@symlabs.com)

**PDP-P**  See Service Provider Policy Decision Point

**Service Provider Policy Decision Point**  *** TBD

- Source: David (d.w.chadwick@KENT.AC.UK)

**PEP-P**  See Service Provider Policy Enforcement Point

**Service Provider Policy Enforcement Point**  *** TBD

- Source: David (d.w.chadwick@KENT.AC.UK)

**SPPE**  See Service Provider Process Engine

**Service Provider Process Engine**  Controlling logic of the Service Provider.

- Source: Danny (decockd@esat.kuleuven.be)

**SRPE**  See Service Requester Process Engine

**Service Requester Process Engine**  Controlling logic of the Client.

- Source: Danny (decockd@esat.kuleuven.be)

**STM**  See Session Trust Management

**Session Trust Management**  System which builds trust from session parameters such as authenti-
cation parameters used. Establishing a given LoA is an example of STM.

- Source: Jerry (j.d.hartog@tue.nl)

**SOAP**  See Simple Object Access Protocol
Simple Object Access Protocol SOAP is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) as its message format, and usually relies on other Application Layer protocols (most notably Remote Procedure Call (RPC) and HTTP) for message negotiation and transmission. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built.

- Source: Quentin (Quentin.Reul@vub.ac.be)
- Reference: http://www.w3.org/TR/soap/

SLO See Single Log-Off

Single Log-Off The converse of SSO, whereby a user is simultaneously logged out of all the services that he is currently logged into via SSO.

- Source: David (d.w.chadwick@KENT.AC.UK)

SSO See Single Sign-On

Single Sign-On The process whereby a user can sequentially gain access to a number of computer services by only providing his login credentials once to the first service he contacts.

- Source: David (d.w.chadwick@KENT.AC.UK)

SoA See Source of Authority

Source of Authority Root of trust, issues ACs and may have subordinate AAs.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary
- Comments: CONFLICT: Potential problem with acronyms

StTM See Structural Trust Management

Structural Trust Management Class of trust management systems which use formally expresses trust facts and relations to establish trust. CTM and STM are examples.

- Source: Jerry (j.d.hartog@tue.nl)

Structural Trust Rules It can be simple trust statements as Provider X is trusted to supply Job Vacancies and the combinations trust relations for example when the party trusted to issue credentials is itself determined by trust rules; Provider X is trusted to supply Job Vacancies if a trusted Accreditation agency certifies them. An Accreditation agency is trusted to certify Providers if it is registered at a national registry and has a good reputation, etc.

- Source: Sampo (sampo@symlabs.com)

Subcontinent *** TBD

- Source: Sampo (sampo@symlabs.com)
**Subject**  An actor who wants to perform an action on a target.

**Symmetric Authentication Method**  A method of authentication in which both entities share common authentication information.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: ITU-T Y.IdMsec, X.811

**Target**  A resource on which a subject tries to perform an action.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Reference: PERMIS Glossary

**TAS3 Trust Network**  See Trust Network.

- Source: Sampo (sampo@symlabs.com)

**Test Driver**  A dedicated software service that is able to run test suites on a service under test.

- Source: Seda (sguerses@ESAT.KULEUVEN.BE)

**Test Storage**  A dedicated software repository that stores test suites to be used by Audition.

- Source: Seda (sguerses@ESAT.KULEUVEN.BE)

**TAXI**  See Testing by Automatically generated XML Instances

**Testing by Automatically generated XML Instances**  A tool by CNR that generates XML instances from an XML Schema automatically. The methodology is largely inspired by the Category Partition testing technique.

- Source: Antonia (antonia.bertolino@isti.cnr.it)

**Threat**  A threat is the description of an adverse event that is considered as potentially having a negative impact. A Threat by itself is not interesting, but becomes relevant when associated with a targeted model element and a security requirement.

- Source: Quentin (Quentin.Reul@vub.ac.be)

**TTL**  See Time-To-Live

**Time-To-Live**  Parameter that indicates how long a cache entry is valid. Generally a cache entry will not be re-fetched until TTL expires. This concept is especially used by the DNS.

- Source: Sampo (sampo@symlabs.com)

**TLG**  See Top Level Guarantor
Top Level Guarantor  
See Trust Guarantor.

Trail  
A "transport entity" which consists of an associated pair of "unidirectional trails" capable of simultaneously transferring information in opposite directions between their respective inputs and outputs.

- Source: David (d.w.chadwick@KENT.AC.UK)

Transmission Media Layer Network  
A "layer network" which may be media dependent and which is concerned with the transfer of information between transmission media layer network "access points" in support of one or more "path layer networks".

- Source: David (d.w.chadwick@KENT.AC.UK)

Transport  
The functional process of transferring information between different locations.

- Source: David (d.w.chadwick@KENT.AC.UK)

Transport Entity  
An architectural component which transfers information between its inputs and outputs within a layer network.

- Source: David (d.w.chadwick@KENT.AC.UK)

TLS  
See Transport Layer Security

Transport Layer Security  
*** TBD

- Source: Sampo (sampo@symlabs.com)

Transport Network  
The functional resources of the network which conveys user information between locations.

- Source: David (d.w.chadwick@KENT.AC.UK)

Trust  
Is used to refer to both the subjective notion of trust, i.e. a perceived likelihood that an entity/system will behave/perform as required as well as to a formalization of this notion into a measurable quantity.

- Source: Jerry (j.d.hartog@tue.nl)

TPN  
See Trust and Privacy Negotiator

Trust and Privacy Negotiator  
*** TBD
T&S  See Trust and Security

Trust and Security  DEPRECATED

- Source: Sampo (sampo@symlabs.com)

Trust Consortium Convener  See Trust Guarantor.

- Source: Joseph (joseph.alhadeff@ORACLE.COM)

Trust Ecosystem  The users, members, suppliers, and stake holders of a Trust Network.

- Source: Sampo (sampo@symlabs.com)

Trust Information Collector  A point which gathers feedback information needed to calculate reputations (see also WP2 D2.1 deliverable).

- Source: Sampo (sampo@symlabs.com)

TG  See Trust Guarantor

Trust Guarantor  Governing entity of a Trust Network. The top level Trusted Third Party that administers the Trust Network.

- Source: Sampo (sampo@symlabs.com)

TM  See Trust Management

Trust Management  An approach to making decisions about interacting with something or someone we do not completely know. It formalizes the subjective notion of trust into measurable trust levels by quantifying and combining sources of trust such as credentials expressing trust statements by entities, recommendations and feedback on performance, etc.

- Source: Jerry (j.d.hartog@tue.nl)
- Reference: Deliverable TAS3D5.1
- Comments: CONFLICT:

Trust Negotiation  The process whereby two entities negotiate a trusting relationship between themselves, by sharing their credentials that were issued to them by TTPs that both of them trust.

- Source: David (d.w.chadwick@KENT.AC.UK)
- Comments: CONFLICT: Same acronym as Trust Network suggested by David.

TN  See Trust Network

Trust Network  An online business environment where parties can interact with each other securely. While the network does not warrant honest behaviour of the members in the network, it does ensure that everybody adheres to some basic principles especially in non-repudiation, data security, communications security, and IT security. Thus a Trust Network promotes trust between its members.
Trust Network Domain  *** TBD

TO  See Trust Operator

Trust Operator  See Trust Guarantor.

T-PDP  See Trust Policy Decision Point

Trust Policy Decision Point  A Policy Decision Point specialised in evaluating trust policies. Will answer authorization request with a granted (trusted) or denied (insufficient trust established) decision by combining different trust management techniques.

Trust Seal  A seal awarded by a proprietary company, usually a certification authority, to business web sites to display in an attempt to boost consumer confidence in the site. Seals are often awarded when the web sites purchase SSL certificates from the CA. The seals are usually trademarked or copyrighted to prevent them from being copied illegally.

Trust Service  A webservice which evaluates trust using a trust management approach such as CTM, RTM, STM, KPITM.

TAS3  See Trusted Architecture for Securely Shared Services

Trusted Architecture for Securely Shared Services  EU FP7 Project.

Trusted Entity  An entity that can violate a security policy, either by performing actions which it is not supposed to do, or by failing to perform actions which it is supposed to do.

TTP  See Trusted Third Party

Trusted Third Party  An entity that is technically trusted by the infrastructure to assure correctness of some transaction or relationship. TTP is generally subordinate to Trust Operator, the latter being responsible for the overall oversight.

Source: Sampo (sampo@symlabs.com)

Source: Jerry (j.d.hartog@tue.nl)

Source: David (d.w.chadwick@KENT.AC.UK)

Reference: ITU-T Y.IdMsec, X.810

Comments: Current definition is of an entity that is apparently trusted because it is not prevented from doing wrong things, i.e. it is an entity that needs to be trusted for the system to be secure. In a perfect world this would be the same as actually trusted entities but of course the world is not perfect. I think using this definition is bound to be confusing. I suggest using Trusted Entity for entities which are actually trusted. Perhaps use a term such as entrusted entity for the current definition if this term is needed. David?
Trusted Zone  From the viewpoint of a NGN provider a security domain where a NGN provider’s network elements and systems reside and never communicate directly with customer equipment. The common characteristics of NGN network elements in this domain are that they are under the full control of the related NGN provider, are located in the NGN provider premises (which provides physical security), and they communicate only with elements in the “trusted” domain and with elements in the “trusted-but-vulnerable” domain.

Trustworthiness  The amount in which an entity is worthy of being trusted. Is used for both the subjective notion of trust; i.e. is the entity likely to behave as expected and the formalized notion, i.e. has a sufficiently high trust level been established for the entity.

User  Human that uses the Trust Network. In Liberty and SAML contexts User is synonymous with Principal.

User Identifier  Identifiers that represent users in their interactions with other parties. Users may present their identifiers verbally, on paper, on plastic cards, or in any other appropriate manner. Electronic user identifiers are electronically presented over data communication channels by user-operated computing devices (client devices) such as PCs, laptops, mobile phones, and smartcards.

User Identity  A code or string uniquely identifying a user across a multi-user, multi-service infrastructure.

Verification  The process of confirming a claimed Identity. For example: any one-to-one precise matching of an identity’s registered credentials, such as in a logon or any non-AFIS process. Usually performed in real-time, with a yes/no outcome.

Verification Authentication Information  Information used by a verifier to verify an identity claimed through exchange Authentication Information.
Verifier  An entity which is or represents the entity requiring an authenticated identity. A verifier includes the functions necessary for engaging in authentication exchanges.

Vulnerability  A Vulnerability is a flaw in a system’s design or its implementation. It is a weakness that might be exploited to cause a system to malfunction, ultimately resulting in some harm or loss.

X.500  Series of computer networking standards covering electronic directory access. Similar to LDAP.

X.509  A joint standard by the ITU-T, ISO and IEC which describes both PKI and PMI. X.509 public key certificates are ubiquitously used on the web for SSL/TLS communications with web servers.

WS  See Web Service

Web Service  Web Service is SOAP based machine to machine communication. Sometimes specifically Identity enabled web service, e.g. Liberty ID-WSF based WS.

WSDL  See Web Service Description Language

Web Service Description Language  WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate.
**Source:** Quentin (Quentin.Reul@vub.ac.be)

**Reference:** http://www.w3.org/TR/wsdl20/

**WSP** See Web Service Provider

**Web Service Provider** *** TBD

- **Source:** Sampo (sampo@symlabs.com)

**Workflow** *** TBD
Bibliography


[AcctSvc] "Liberty ID-WSF Accounting Service Specification"


[CardSpace] InfoCard protocol (aka CardSpace) from Microsoft


[DesignPat] "Liberty ID-WSF Design Patterns", liberty-idwsf-dp-v1.0.pdf


[Disco12] Liberty ID-WSF Discovery service 1.1 (liberty-idwsf-disco-svc-v1.2.pdf)

[DST11] Liberty DST v1.1


[FF12] Liberty ID Federation Framework 1.2, Protocols and Schemas


[OAUTH] http://oauth.net/


[Peeters09] Roel Peeters, Koen Simoens, Danny De Cock, and Bart Preneel: "Cross-Context Delegation through Identity Federation", KUL 2009 (To be published?)


[RFC3548] S. Josefsson, ed.: "The Base16, Base32, and Base64 Data Encodings", July 2003. (Section 4 describes Safebase64)


[SAML11core] SAML 1.1 Core, OASIS, 2003


BIBLIOGRAPHY


[SWIG] Simplified Interface and Wrapper Generator by Dave Beazley. www.swig.org


[TAS3CONSOAGMT] "TAS3 Consortium Agreement", TAS3 Consortium, 2008. (Not publicly available.)

[TAS3D12DESIGNRAR] David Chadwick (Kent), Seda Gürses (KUL), eds.: "Requirements Assessment Report", TAS3 Consortium, 20090102. Document: TAS3_D1p2_Requirements_Assesment_Report_1_V1p0.pdf


[WSTrust] "WS-Trust 1.3.", CD 6, OASIS, Sept 2006. (** WS-Trust, STS, etc.)


[XACML2SAML] "SAML 2.0 Profile of XACML, Version 2, Working Draft 5", 19 July 2007, OASIS. (** instead of "SAML 2.0 profile of XACML v2.0, ERRATA, Working Draft 01, 17 November 2005" which is the version that the profile is currently based on; XACMLContextProfile1.1.doc from Open Grid Forum - OGF)
BIBLIOGRAPHY

[XML] http://www.w3.org/TR/REC-xml


Document ID tas3-deliv-2_1-arch-v17.pdf

URL path https://portal.tas3.eu/arch/review/tas3-deliv-2_1-arch-v17.pdf
The following part of the D2.1 document will become a stand-alone document in the future.

Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Luk Vervenne, editor</td>
<td>Synergetics</td>
</tr>
<tr>
<td>2 Sampo Kellomöki</td>
<td>Eifel</td>
</tr>
<tr>
<td>3 Tom Kirkham UoN</td>
<td>Univ. Of Nottingham</td>
</tr>
<tr>
<td>4 Joseph Alhadeff</td>
<td>ORACLE</td>
</tr>
<tr>
<td>5 David Chadwick</td>
<td>Kent</td>
</tr>
<tr>
<td>6 Antonia Bertolino</td>
<td>CNR</td>
</tr>
<tr>
<td>7 Ingo Dahn</td>
<td>Koblenz</td>
</tr>
</tbody>
</table>

Legal Notice
All information included in this document is subject to change without notice. The Members of the TAS3 Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the TAS3 Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.
Executive Summary

In the last decade the Internet has consistently pushed the balance of power between the organization and the individual towards more user-empowerment. Today most services are being reengineered to be demand-led and user-centric.

While empowering the user/consumer/customer does create a more dynamic, agile economy, enhances consumer choice and spurs service innovation, the 180° switch from provider-led to demand-led services does come at a price.

Agile and efficient demand-led services, in fact require that users and service providers can present and exchange coherent and trustworthy information. Until now that information was collected, managed, stored and mostly kept if not shielded or stolen by the service providers. Today, at the height of using these antiquated business models in an Internet age, the truth is that your personal information sits in a 1000 databases. At best you know 10 of them.

TAS³ has set out to provide an answer to the user-controlled, trusted sharing of personal information in a user-centric, demand-led services economy.

This document explores the various Business Models for the Trust Networks (TNs) that will need to assure the trust for user-controlled Personal Information sharing. We acknowledge that there will be multiple Trust Networks. A given Service Provider may belong to more than one. This document also explores the role of Trust Guarantor in the Trust Network and its relation to the concept of Trusted Third Parties (TTP) in general. TTPs can be Identity providers, authentic (data) sources, etc. It is recognized that Trusted Third Parties will be used to link trust and identity across networks.

The Trust Guarantor’s role is to provide overall coordination of the operation of the Trust Network and to ensure that all Trusted Third Parties as well as the Service Providers are performing to their obligations. Governance aspects and stakeholders of a Trust Network are examined. Some acute privacy threats are examined. Costs and sources of revenue are identified. Some recommendations for government policy are highlighted.
TAS³ Business Model

TAS³ (Trusted Architecture for Securely Shared Services), aims at the creation of a secure and effective means for individuals to online monitor and control their personal information, when this is produced, used, or requested by service providers. TAS³ business models will have to provide an answer to the social innovation trends that underpin it. As such the TAS³ technological development will have to proceed alongside non-technological innovations and innovations based more on the notion of a demand-led services economy. A key task of the Demand-led Innovation is the promotion of dialogue between users and service providers. User-led innovation is promoted in traditional business sectors, in private and public services, and in sectors which generate new demand.

TAS³ aims to achieve this vision either via a distributed or central approach, however emphasizing the sharing and componentization of services and user-centricity in terms of service provision and data storage. Our working assumption is that the only robust and practical way to achieve this goal is to create a so-called “Trust Network”. Within a Trust Network information exchange and transactions are supported by guarantees in terms of both quality and the various trust & security components (authentication, authorizations, data privacy and trust management). Underpinning the Trust Network is a set of services called the Trust Network Infrastructure Services (TNIS) providing a core trust infrastructure supporting information exchange based on user control in the trust networks.

Central to the operation of the network based trust infrastructure is the use of specific Trusted Third Parties (TTPs) for mechanical & legal validation of services (providers + requesters) and (end-) users in the networks. The trusted third parties also interface with a higher level definition of trust metrics overseen by a top level Trust Guarantor. It is envisaged that cross Trust Network communication will be enabled by co-operation between Trust Guarantors. This eventually will result in a Trust Ecosystem.

![Figure 1: Main Components of a Trust Network](image-url)
Technically the top level Trust Guarantor has a fundamental role in (1) introducing, (2) monitoring, and (3) auditing the end2end assurance of trust between the transacting parties.

1. All parties in the Trust Network (i.e. (1) end-users, (2) service providers & requesters, including (3) TTPs) will be represented by tangible legal entities that agree to the terms of the trust network before participating in it. The top level Trust Guarantor will set these terms and therefore define the laws / nature of the trust network and has to fulfill both technical and legal audit roles in the trust network.

2. All parties including end-users, service providers/requesters of application specific services (i.e. eHealth tools) and service providers/requesters of TTP services (i.e. authentication) will be monitored by the overall TAS³ Trust Network Infrastructure Services (TNIS) that spans the specific trust network, its trusted services and the related and information exchange. Any party breaching of the terms of the Trust Network ecosystem and/or trust network will be reported to / monitored by the Trust Guarantor.

3. Any breaches of terms or failures of the Trust Network are the responsibility of the Trust Guarantor. Therefore the body has to act upon such cases and eject / penalize members of the ecosystem who break its rules. For example this could be done (1) via reduction of trust ranking of specific services, (2) by legal action on the basis of the TAS³ legal contract, including the expulsion of the involved party. This level of support will lead to an a-priori assumption that it is basically safe to transact with any member of the network, promoting trust in the whole network which leads to its acceptance and widespread use. This brings down (compliance) costs of doing business, reduces fraud and abuse of information, and ultimately leads to new innovative ways of transacting.

This vision raises some fundamental questions that need to be addressed:

1.1 What should constitute a Trust Network?

TAS³ is developing a generic architecture for securely shared services related to personal information. This Architecture is to be implemented in four parts:

- business requirements
- technical requirements
- policy requirements
- legal requirements.

The parties covered by the TAS³ architecture fall into three main categories:

- Trust (infrastructure or service) entities: trust guarantors and its certified Trusted Third parties such as authentic sources or identity providers)
- Application specific Service Providers (in TAS³ these are either employability or eHealth related)
- End-users (individuals)

All parties should consider the technical, policy and legal requirements to be the minimum requirements of the architecture.

- In the case of technology requirements, there may be limited flexibility in some implementation parameters to ensure interoperability:
- In the case of policies, they can be used in 2 ways. Participants to the TAS³ consortium can either adopt the policies promulgated as their own, or can map
their policies to the model policies to assure that they meet the minimums required. The policy blocks presented by TAS³ can be used to create policies or are to use existing possible legacy policies and map onto the TAS³ definitions. Participants may need to provide evidence of the policy gap analysis if they rely on existing policies for compliance and also may need a mapping tool.

- Lastly, the legal requirements are reduced into contracts tailored to the role that each participant is playing, so they must be completed and adhered to. When registering to the Trust Network, the contract terms will bind organizations to technical and policy requirements, both in terms of those expressed at the intra- and inter-organizational level as well as in terms of using the appropriate trust technologies to honor the preferences and choices of users as to use and sharing of personal information.

Trust verification and assurance are essential elements of the TAS³ Infrastructure, and thus the organization and cooperation of trust enablers in the operation and oversight of trust is essential. The co-operation is hierarchical in terms of the use of Trust Network level definitions, down to user and service definitions.

**TAS³** Trust Networks are monitored & trust assured by (1) an independent Trust Guarantor supported by one or more (2) TAS³ certified Third Trusted Parties (TTPs). Both must be in an absolute position that provides them with an oversight role without having to provide services that place them in a position of conflict with data subjects.

The TAS³ Consortium members will periodically review the architecture requirements from a trust and oversight perspective or may engage in more frequent reviews as a result of changes in legal requirements, regulatory requirements, or cases that require new terms.

A **TAS³ Trust Network** therefore will usually consist of:

a) **Trust Network Governing Board.** The board manages the affairs of the trust network. Depending on the domain this might be a public-private-partnership (PPP) consisting for instance of the main societal eHealth or employability domain stakeholders.

   *For instance, in the Limburg province in the Netherlands, such PPP is currently in the making for a regional employability platform. It involves representatives of the employers, labor unions, educational sector, local governments, industry sectors,*

b) **Trust Guarantor.** The trust guarantor is the technical operator of the trust network and its **Trust Network Infrastructure Services (TNIS)**

c) **User Representation.** The Trust Network will need some form of end-user representation. Depending on the type if Trust network this can range from existing end-user representatives such as labor unions, industry sector federations, government, grass roots user groups, ...) In essence this boils down to “organizing the communality”

d) **Governments.** We notice that governments in UK and the Netherlands (both at local and national level) are gearing up to help initiate / organize / facilitate the needed communality around user-centric data and services and the trust this requires.

The TAS³ Trust Network Governance Agreement as monitored and audited by the Trust Guarantor therefore has the following tasks:

- Monitor the governance structure of the Trust Network
- Register, certify, oversee and audit of the TTPs active in the Trust Network.
- Register, certify, oversee and audit the Service Providers
The certified Trusted Third Parties, working in framework set by the Trust Network as executed by the Trust Guarantor will typically be existing actors possibly performing the following functions:

- Identity Providers (IdPs), to be certified by the TG
- Discovery and registries, to be certified by the TG
- Reputation Providers, to be certified by the TG
- PKI (q.b.) to be certified by the TG
- Authentic attribute sources, to be certified by the TG
- Etc...

Service Providers, may participate in multiple TNs (having a non-exclusive relationship), and choose their TTPs from available choice within each TN.

- Service Providers that act as data requestors
- Service Providers that act as data providers (running trusted repositories)
- Service Providers that act as data originators

End-Users can expect the following services from the Trust Network:

- Certification and audit procedures
- Branding (might/should be reputation based on live tangible metrics. Problem with brand is that it can take on a life of its own
- Secure and dependable technical infrastructure assuring trusted sharing of his personal information.

A Trust Network is built around an accountable legal entity, the **Trust Guarantor (TG)**. Accountability implies both oversight and (legal) responsibility. The issues of obligations and liability must be clarified in the agreements that bind the party and must be appropriate to both the role and risk assumed by each party. The TG organizes and charters multiple technical Trusted Third Parties (TTPs) in order to perform specific and partial trust functions of the Trust Network. The sum of the delegated TTP functions may or may not cover ALL the operational functions of the Trust Guarantor. If it does, the remaining responsibility of the trust guarantor consists of overall management, certification and auditing.

A TTP will typically NOT have an exclusive relationship with TN and can operate in several TNs. TTPs should be leveraged to gain faster take-up and market acceptance of the Trust Network. Often this is also both inevitable and necessary because:

- the TG does not have all the skills needed
- there are already players in the market like:
  - Certification Authorities
  - issuing certificates
  - credit check operators
  - providing reputation

Other participants of a Trust Network are Service Providers who transact with each other, and Users who use the services of the Service Providers. Users also have a special role in that they may commit into the network data that needs special protection. The Trust Network and its Infrastructure Services only exist for the benefit of its users, not to enslave them and will be reflected in the way the user data policies are respected.

**Public vs. Private networks**

The Trust Network is generally foreseen to be a public and nonexclusive entity: anyone, User, Service Provider, or even Trusted Third Party operator, willing to be certified can
participate. Trust Networks may compete on issues such as cost, trust level, terms of use and even competence of members (i.e. specialists). That being said, TAS³ Trust Networks do not exclude the possibility to run private exclusive networks. Enabling such private networks however, is a non-goal, but is not an anti-goal either.

From that perspective a Trust Ecosystem (consisting of several Trust Networks) becomes possible that are made up of component TN systems. This would allow some parties to seek to develop private, closed or exclusive networks that are compatible with the TAS³ infrastructure but not subject to it. In itself this may enable some information transfers across providers that are both in public and private networks in order to service particular customer needs, but would not necessarily imply that such private providers were under the TAS³ Governance model or direct oversight. Thus TAS³ may also be considered a portable standards-based business model, but those wishing to use it for that purpose will need to appropriately adapt it and develop their own oversight models.

Each Trust Network is governed by a Governance Agreement to which all parties agree. <<ignore: footnote: David suggests an alternative model where some (or all?) aspects are left open, for the players to define. The members could, for example, define their own TTPs for some functions. In a way this can be seen as sub-network within a network. It may also lead to diminished brand value for the whole network.

**PS:** In implementing the TAS³ requirements, two equally valid models may be used simultaneously. Some players may wish to adopt whatever criteria are created by TAS³, where others may map their existing criteria to TAS³ to demonstrate equivalent compliance and interoperability. As we work on the development and pilots of TAS³ we shall seek to maintain whatever flexibility is possible that is consistent with governance, oversight and end-to-end security.

Trust Network participants will be subject to a general framework contract. This covers the overall rules of engagement for any user (end-user or service provider) of the Network and creates the needed relationships for obligations to be enforced against service providers. For these service providers this general framework agreement is then supplemented with role and transaction based contracts, covering not only what is allowed within the Trust Network, but also how data acquired for specific purposes should be handled beyond the reach of the TNIS monitoring capabilities (read: behind the service provider firewall).

**Branding and reputation**

Depending on the constitution of the Trust Network Governing Board the notion of branding the Trust Network may or may not be effective. If the Trust Network is merely an organization that operates the technical/legal Trust Infrastructure Services branding may or may not work.

In fact, when not backed up by a community of practice, Trust brands in some cases have shown to fail, e.g. some of the websites carrying a certification brand have never-the-less been fraudulent (even more so than sites without certification). I.e. a brand is not a guarantee in itself. This argument could go as far as recommending that no brand should be used in order to avoid inducing users into the false belief that a brand guarantees something. Users are not able to remember the historical track record of a brand and will instead trust it on basis of first impression or recent marketing.

If however the Trust Network/Guarantor is foremost setup to trust-enable a public-private-partnership, (covering for instance the employability services in a region) and
this PPP is acting as the Trust Networks’ Governing Board, then such a community of practice and/or its TN may well become a solid brand. While branding is foremost a user perception related term, more tangible trust is to be expected from real time reputation. In fact the TAS³ type of Trust Networks are based on trust which has to be user defined and real time, while brand is not defined by users nor is it real time. Nevertheless we would argue that TAS³ - where possible - should try to combine both approaches and in fact both are needed to produce end2end trust:

- The notion of BRAND has a connotation with user trust perception. Users are expected to perceive the brand as trusted, though if the network is mismanaged and the trust is not earned, the opposite may happen. The brand will also be used in certification: only valid participants are allowed to display the brand.
- REALTIME REPUTATION however requires measurable trust metrics. TAS³ therefore builds in reputation into the system of transaction guarantee, i.e. 100% compo if you use a gold star ranked service provider or 50% if you use a grey star ranked service provider and the SLA of TAS is breached.

Finally, to build a Trust Network, build its brand and promote its adoption, technologies and products implementing the technologies will be needed. In a mature market these may be available off-the-shelf. However in an early innovator market, such as TAS³ type of TNs, ensuring that these are available and of high enough usability and quality can be instrumental to the success of the network. The Trust Guarantor therefore needs to work with all system participants and technology developers to ensure this is the case. Similarly, even user friendly and user centric applications require some basis of familiarity or necessity for uptake among consumer/citizen users, which will have to be addressed.

Users trust perception

Users should be able to join trust networks by agreeing to terms and conditions of use. The User can then allow his personal information to be shared within the network in order to become part of distributed composite applications & services. It is the central focus of TAS³ that when users present their personal information to a TAS³ Trust Network, they can trust that it will be not used out of context of the terms that they agreed when joining the network and the policies set out for the actual transaction. The trust is based on the end2end assurance provided by the Trust Guarantor, and relies on a combination of technical monitoring and enforcement capabilities and legal contracts signed by all involved parties. More specifically, legal contracts extend the reach of enforcement beyond the TN perimeter and beyond the service providers’ firewall.
The ultimate guarantee that the data will not be misused is presented by the trusted guarantor who effectively takes on the liability for their respective trust networks. When joining the network the user will agree that in the case of breach the guarantor will compensate / take action to rectify. Service providers in the domain are tied in by the same rules. The action taken might include legal actions, insurance claims, service provider depreciations of exclusion, etc...

Beyond the brand perception of a TAS³ trust network, its’ trust model works foremost on the user defining policies for their own personal information when joining a network and at the time of the transactional network decisions based on the users’ information.

Quality of trust

A key business opportunity in TAS³ is the concept of user trust perception. As users present their data to TAS³ various methods of reporting can be used to keep the user in the loop. For example some users may wish to keep a close account of what applications their data is being used in or the status of their application execution. This can be achieved through the use of the trust dashboard. Trust dashboards will help the user to discover & select the appropriate trusted service providers according to specific terms and return them in trust rank order like Google.

Users could sign up for different qualities of dashboard and this will be reflected in the cost of the application. These could be scaled in terms of price. The least expensive dashboard could present users with almost a ‘fire and forget’ interface to TAS³ networks allowing application invocation and presentation of results when done.

A more expensive and top of the range dashboard could notify via a variety of means the various hops that the user’s data may have in a trust network as it is used in applications. This could be achieved via email, SMS, etc. Service providers could compete to provide new innovative ways to report on the progression of the users data through the network.
Overall the user’s perception of trust can be seen as related to their knowledge, and this needs to be reflected in the way users can interact with TAS³. It is likely that the GUI to TAS³ will not carry much weight in terms of trust perception as users will be directed to choose from reputation rankings and elements such as cost. Some users may interact through specific TAS³ interfaces whilst others may use TAS³ embedded in existing applications. In both cases the users need to sign off on their policy refinements and have a means by which they can be notified of their data use.

2. How many of Trust Networks should there be?

First of all we like to restate that an important goal of the TAS³ project is to create documentation and software so that Trust Networks will be easy to setup, whatever their application domain.

As argued on page 4, we expect TAS³ TNs to emerge around a clear business need, as perceived and made operational by Public-Private-Partnerships (PPP). Early models are likely to be government (1) initiated, (2) facilitated, (3) mediated, (4) anchored or even (5) owned (notice the increasing governmental involvement).

PS: Of course this view has its limitations: it would for instance seem unreasonably stifling to outright forbid private Trust Networks. Society and public debate should establish what distinguishes a public Trust Guarantor from a private one and what regulation should apply to each kind. On the other hand, let’s not forget that a TN represents foremost the user’s interest (and personal information).

As such a Trust Network as something similar to a bank or telecoms operator. There is room for more than one and indeed having several will promote healthy competition. However, due to the special infrastructure utility role, Trust Networks are likely to be heavily regulated and there would only be a handful of them.

Conclusion: With the TAS³ project initiated from a need for trusted sharing of personal information, we see Trust Networks arise from two angles:

- With the user as the ONLY ‘lifelong’ continuum within Trust Networks, the variety and scope of the TN is likely to be fitted around the users’ health, wealth and happiness!
- From a service providers perspective we see two orthogonal axis or attraction pools:
  - regional development, interests & communality
  - domain/industry sector specific interests.

As such we expect a bottom-up approach with smaller, local initiatives being used as reference cases and national governments overseeing the results and eventually building momentum for larger, possibly national roll-outs, where different trust networks can be interlinked into Trust Ecosystems. In fact the Trust Ecosystem level could be the goal of the TAS³ project guiding principles, standards & methods (and tools?), promoting them to new candidate Trust Networks. It may also be the correct level to discuss cross-country issues.

3. Should Trust networks interact with each other?

As TAS³ services mature, the focus will shift towards building efficient larger trust ecosystems, which means connecting different context networks and avoiding
overlapping functions. Again TAS³ is build from the ground up as a generic & domain-independent architecture. Multiple Trust Networks (say a European wide and interlinked employability trust networks) can be linked into for instance a larger European Employability Trust Ecosystem. This requires that the involved Trust Networks cooperate and have established a set of common rules of engagement, both at technical and legal level. Besides providing first insights and comments, the TAS³ project however considers the Trust Ecosystem level to be outside of the scope of the project and of its demonstrators.

We are presuming sectorial/national trust ecosystems at the outset. But these ecosystems may be comprised of trust network solar systems in galaxies that in turn make up the universe of the national sector. The business, technological, policy and legal contractual root of these interlocking players will be the architecture defined by TAS³ which will enable interoperability, where needed. Not all players will interact with each other, but rather interact as required by need. It is impossible to predict in advance all of the specific participants to any transaction type, as user needs and preferences must be factored. The idea of parallel, but disjointed Trust Networks lacks credibility in today's globalized world. The users would not be able to understand why the networks do not talk to each other and a multitude of kludgy or illicit "gateways" would spring into existence whether we want or not. Much better policy is to foresee the interaction directly in the architecture.

However roaming the trust concept from one TN to another is quite challenging and requires standardization of the different trust concepts. Nevertheless commercial systems have proven to be quite adequate in solving these types of unclean interfaces. As long as someone is willing to carry the liability for occasional mismatches and leaks, it can be made to work. Risk management is good enough if you can't prevent the risks entirely. This is for instance how the credit card system works.
4. Who should run them?

*Initially we expect the involved (innovating) project authority to govern the TN. Later on the powers that be will take over, unless the initiators manage to cross the chasm.*

**Figure 2: Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers** (1991, revised 1999), is a *marketing* book by Geoffrey A. Moore that focuses on the specifics of marketing high tech products. Moore's exploration and expansion of the *diffusions of innovations* model has had a significant and lasting impact on high tech entrepreneurship. In 2006, Tom Byers, Faculty Director of Stanford Technology Ventures Program, described it as "*still the bible for entrepreneurial marketing 15 years later*". This success resulted in several follow-up books and a consulting company, *The Chasm Group*.

Trust Network should be run by a credible and long lasting legal entity, with the necessary strong user community impact. Without government backstop, there might be a question of credibility to oversight, for instance. As such Trust Network is likely to be a non-for-profit PPP or Consortium type of organization, run by a representative governing board. The Trust Guarantor on the contrary has a specialist operational task probably best suited for a commercial entity, contracted by the TN.

Nevertheless, just for the sake of it, we list some other alternatives:

- Public-Private partnership with a user community impact
- New for profit company founded for the purpose (needs to build reputation)
- New non-profit foundation or association created for the purpose (needs to build reputation)
- Existing non-profit foundation or association
- Certification Authority
- Other major (consumer) player
- Government institution
- Government
- EU
• Charity
• Bank (they run your money, why not your personal data?)
• Insurance Brokers (you honest broker represents users)
• United Nations

A special peril would seem to be that the Users are easily left without representation (they are not foreseen to sign the Governance Agreement). Part of this problem is that there is no obvious party that could represent the users. Should they be represented by some consumer organization? We noted that Labor unions stepped up for the employability use case in the Netherlands, but on a more general level, outreach to privacy and consumer organizations would be recommended.

5. How should the governance be organized?

Since the Trust Network ‘polices’ the sharing of services and personal information exchange between multiple parties, it seems natural that the representative societal stakeholders that traditionally help manage users, providing them with a service offering, are the prime candidates to be brought onboard.

a. On the one hand will TAS³ enable them to evolve into demand-led service providers (representatives) and on the other hand they are needed create momentum for the trust network to become accepted as the 'new way forward'.

b. Overall we see national and local governments as the possible initiator and the most likely facilitator to help engage the stakeholders in adhering to the Trust Network compliance. Some caution is needed in defining the role of governments, since in a user-centric service economy they are also service providers like any other.

c. Hence the need for a society-wide Public-Private-Partnership, where all representative parties involved will need:
   1. a clear win-win for their members and constituency (why)
   2. adhere to TAS³ compliance and its common rules of engagement (how)
   3. a board seat and a responsibility in governing the Trust Network, following the Trust Network governance agreement (what)

The basic premise of Trust Networks is that transactions of monetary value will be involved. There will also be data protection issues to which liability is attached. Liabilities will eventually bring the Trust Guarantor, Service Providers, Trusted Third Parties or indeed even an end-user in court. Law and legal contracts will therefore provide the ultimate safety nets.

All parties are bound by contracts which set out rights and obligations. Users will sign documents related to terms and conditions, but they will be geared to the exercise of their rights and recourse, although it will also set out the need for them to be bound to their choices and act in ways that nobody undermines the system or attempt to defraud or otherwise injure other parties literally or figuratively. It therefore seems logically that governments sooner or later are going to regulate the Trust Networks.

To stave off excessive regulation and to delay regulation in general, Trust Guarantors have active interest to self-regulate. This places heavy emphasis on the Trust Network's Governance Agreement.

Such a Governance Agreement should address:
• Governance structure, such as advisory and audit boards
• Criteria to join and stay on the network, including certification and audits
• Process for removal from the network
• Process for complaints
• Commercial liability and its fair appropriation
• Liability due to negligence in criminal cases and its fair appropriation
• Data protection
• Minimal mandatory security practices
• Acceptable use for Service Providers
• Acceptable use for Users
• Licensing of Trusted Third Parties, and their liability

6. How are Trust Networks financed?

a. A TAS³ Trust Network represents the trust assurance for a new way of working between service providers and users. The Trust Network therefore is merely an instrument and, at best, "a conditio sine qua non" for supporting a new demand-led services economy model. The TN therefore foremost replaces (and claims to improve) the existing services and their underlying business processes by changing them into trusted, online web services. We do believe tough that by putting the user in the center, new, and innovative user-centered services will be developed, which did not exist in the old service economy.

b. Financing the Trust network and its operational Trust Network Infrastructure Services (TNIS) therefore is foremost a replacement cost & benefit issue. The two main questions then are:
   • How much money is saved by using a user-centric online trust network, compared to the scattered service provider centric services?
   • How much of these saving can be spend on financing the Trust Network?

c. There is no easy answer. Firstly, one has to calculate the REAL and often hidden cost of let’s say, the lifelong employability of a worker. Today that cost is not even considered from a lifelong perspective! It is spread over any number of service provider cost models.

d. The way forward therefore seems to be to define the specific win-win benefits for the separate main stakeholders that come onboard to found the Trust Network.

By now it is clear that Trust Networks will have operational costs:

- Procurement and maintenance of technical infrastructure
- Member acquisition costs
- Member management costs
- User acquisition costs
- User management costs
- Trust branding costs (marketing)
- Audit costs
- Legal costs
- Liability and insurance costs
- General management costs

Trust Network can be financed in a variety of ways

- Initial capital injection for
  o Procurement of technical infrastructure
  o Procurement of legal contract framework
  o Initial trust branding costs (marketing)
o Initial member acquisition
o General set up

- Fees from Service Providers: there is a need to accommodate many types of Service Providers:
  o Fixed yearly fee
  o Per transaction
  o Per number of Users
  o Per yearly business volume
  o Revenue share
  o Member management fees
  o Audit fees

- Fees from Trusted Third Parties
  o Trusted Third Parties are allowed to have a fee structure of their own
  o Need to accommodate many types of TTPs; for

- Fees from Users:
  o Fixed yearly fee
  o Usage fees from Users. (The tricky part is to collect them)
    - Per transaction
    - Per number of Users
    - Per yearly business volume
  o Revenue share
  o Member management fees
  o Audit fees

- Advertisement
  o Placement of advertisement in authentication steps
  o Right to place advertisements in Service Providers
  o Revenue share from Service Provider advertising revenue

- Proceeds from foundation grant investment portfolio
- Government subsidy, research funding, taxes in a fair way. Perhaps a model where bill for broadband Internet connection includes some fee.

The TAS³ architecture should enable all of the above forms of raising revenue, or at least not block any of them.
A win-win for **ALL** Trust Network Parties

- Authentic, coherent, dynamic, automated and up to date personal information
- User controlled sharing of personal information
- Employability self-empowerment & planning

- Authenticate & direct2process employee data
- Training & Career planning
- Meaningful matching
- Manage multiple-sourced data
- Assessment & recruitment processes

- Engage in demand-led services economy
- Support user-empowerment & LLL
- Facilitate employers & employees
- Matching of unemployed

**Figure 3:** Example of Employability Trust Network win-win benefits
7. What form of Trust Guarantor is most suited to operate and manage the Trust Network Infrastructure Services, a centralized or shared trust model?

The Trust Network Governance Board will appoint a Trust Guarantor to oversee, operate and technically & legally manage the Trust Assurance guaranteed by the Trust Network. While the Trust Guarantor will likely use a centralized model for starters, it is clear that there are already several third trusted parties guaranteeing identities, or authentic sources, etc... Today they are scattered, each having their own – often offline - trust models. The Trust Guarantor will have to incorporate and enable these existing third trusted parties all while providing the end2end trust assurance for sharing personal information. This capability will be build upon the stakeholder engagement.

The Trust Guarantor will likely be a privately held, for-profit company that holds the technical and legal and business skills to operationally oversee and promote the Trust Network, on the request and on behalf of the Trust Network Governing Board. The Trust Guarantor architecture and compliance requirements will need to be matched to Government requirements & regulation.

The Trust Guarantor tasks include:

1. **Assure Compliance to TAS³ specification.** This includes:
   a) Operate or outsource the certification program for software products to be used in the Trust Network.
   b) Operate or outsource the certification program for deployments, i.e. the participating Service Providers, and possibly others like IdPs.
   c) Operate or outsource an audit programme for the deployments
   d) Process complaints and arrange for arbitration or disciplinary action
   e) Market the network to both Service Providers and Users
   f) Maintain government compliance & endorsement as "The Trust Network"
   g) Guarantee minimal cost participation for non-profits

2. **Operate necessary technical infrastructure.**
   Depending on how the Trust Guarantor organizes (1) its business and (2) the Trust Network this may include:
   a) Execute an IdP function or arrange for others to operate IdPs in the network
   b) Authentication providers, in as far as this is not integrated into IdP.
   c) Discovery and registry functions
   d) Dashboard and audit results publication portal
   e) Possibly a certification authority of some sort - this is likely to be outsourced. Certificate or credentials validation or revocation will be a central responsibility of Trust Guarantor.
   f) Network level PDP
   g) Reputation system, or arrange for someone to run the reputation system.
   h) Where users have choice of multiple providers, the Trust Guarantor will need to ensure all in fact work and if not, may need to provide an integration solution, such as a gateway.
   i) Where interaction between networks happens, the Trust Guarantor may operate a gateway that mediates.

3. **Managing liability**
**Panopticon threat**
One especially pertinent risk in running a Trust Guarantor is that it may gain excessive knowledge to the operations of the SP members or the Users and their business processes. This is the so called "panopticon" threat. It can be mitigated by careful division of responsibilities using externally contracted Trusted Third Parties, each of which operates in its own isolated, regulatory scheme.

**Government regulation**
Governments should consider regulating sound operation practices for Trust Guarantors. For example, it might be mandatory to outsource the IdP function. It may also be that regulation will require Users to be able to choose their dashboard or audit provider from choices that are available within the network.

The Trust Guarantor should also be able to make ultimate decisions on suspensions of parties, and will be liable to the core functionality of the trust networks it is responsible for.

**Outsourcing**
Trust Guarantor is a business entity that has liability. The actual running of the Trust Network may involve several outsourced, franchised, or otherwise farmed out functions. The most obvious of these are Identity Provider (IdP), Authentication Provider (usually same as IdP), Discovery Service (DS), Reputation Provider (Rep), and Audit Function.

Thus an actual network will be configured to trust a number of IdPs, DSes, and Reps. In a strict view, all of these entities should be viewed as Trusted Third Parties (TTP), but from business perspective what matters is that they are endorsed by the Trust Guarantor. As such the Trust Guarantor is the ultimate TTP policing the other TTPs and allowing them to enter the network. A clear legal definition of shared accountability and responsibility will be the paradigm in order to foster public trust in the network.

4. **The Trust Guarantor monetary streams are:**
   - Trusted Third Parties contracted on case-by-case basis.
     - Most of these will involve cash outflow
     - Some cases cash inflows may be possible. Never-the-less, to be negotiated.
   - Government Service Providers pay a yearly fee, to be negotiated
   - Commercial Service Providers pay as negotiated, but preferred basis is revenue share or per transaction
   - Small Service Providers pay small
     - yearly fee
     - a one-off Service Provider setup fee
     - support fees once initial support package has been exhausted
   - Value added telephone & (first/second level) helpdesk support for users
   - Advertising in authentication process where feasible
   - Licensing fees or Revenue Sharing from Trusted Third Parties
   - Insurance against liability

In the above listing, there are many charter requirements to guarantee that the Trust Guarantor will operate 'within reason'. Since TAS³ is in position to license its brand and possibly some of its IPR, it should be in position to negotiate with prospective Trust Guarantor to get these charter items included.
8. How do you differentiate between parts of the trust network with oversight responsibilities and service providers that are relevant to trust but may have conflicting interests?

8.1 Kinds of Service Providers

All business actors, other than end-users, are modeled as Service Providers. Obviously there are different types of Service Providers with different legal requirements.

Requesters or Clients and agents: Provide some service to the end-user and in performing this service, will invoke other services, some to perform an action, others to store or retrieve data.

Infrastructure specific service providers: In case the Trust Guarantor does not execute these services which are core to the functionality of the trust network, they can be outsources to Infrastructure Service providers. They provide the core application functions such as accounting, monitoring etc on behalf of the trusted Guarantor. A generic TTP can also be seen as infrastructure specific. The business model / price they operate on may be fixed with the trusted Guarantor in order to guarantee availability.

Application specific services: These services provide the main functions of the network and make it domain specific. For example in an employability network you will have application specific services such as job matching services and CV translator. These types of services can offer a variety of prices and may compete in service selection brokering and negotiation; the cost may reflect a more real-time supply and demand / market place model.

Authentic Data Sources (Data Originators): These are authoritative / authentic source of data may certify its veracity. They may also wish to control where and how the data is used. An originator may use a repository to store the data, or it may act as a repository by itself.

Data (Repository) Providers: These store data on behalf of the user or service provider, but the data in itself may have originated or is referenced outside the repository. Effectively the data provider's repository is handling data on behalf of someone.

8.2 Kinds of Trust Entities

Trust entities will fall out of the TAS³ architecture, i.e. the business model should not nail them down before architecture is decided. However, we can say with fairly good degree of confidence that at least following types of trust entities will exist:

1. PKI Certification Authorities (CAs)
   - Issue SSL and signing certs to system entities
   - Potentially issue certs to users as well (we should avoid this if at all possible)
   - Handle certificate revocation and online status checks (OCSP)
   - No particular conflict of interest seen. TG could well be CA.
   - However, established players exist on market, so it makes sense to leverage them.

2. User registration authorities.
(PS: Sometimes IdPs perform the user registration function as well, but this need not necessarily be so)

3. **Identity Providers (IdPs)**
   - Authentication
   - SSO
   - Possibly some initial attributes
   - Possibly a discovery and/or token issuance service bootstrap

Main problems with IdPs are
   - Visibility to federation relationships
   - Traffic analysis, know who is who's customer and how often
   - Potential visibility to authentication credentials

Since IdP can see so much, it's best of there is no single IdP. The Trust Guarantor therefore probably should not perform this role itself. Instead it should charter or license others to do it. However, to avoid Conflict of Interest, an IdP SHOULD NOT be run by a SP.

4. **Discovery service, service registry, or token mapper**
   - Who provides what service to whom
   - Where do users keep their data
   - Indirection layer in providing end point URLs
   - Credential mapping, from original to specific use

Problems are similar to IdP. Further technical reasons usually dictate that that Discovery is operated by same entity as IdP.

5. **Relationship Service**
   - Who has invited whom to have sharing relationships
   - Social network
   - Groups
   - Delegation use cases
     - Almost certainly should be distinct from IdP to avoid accumulating too much information in one place
     - Technically easy to have multiple PS

6. **Organizational PDPs**
   - Local to organizations operating the SPs
   - Authorizations must be trusted blindly
   - The PDP gets to see quite a lot of traffic analysis info

7. **TAS³ network-wide PDP**
   - Enforces the network wide rules
   - Authorizations must be trusted blindly
   - The PDP gets to see quite a lot of traffic analysis info

8. **Reputation Providers**
   - Reputation based trust scores are computed from usage pattern data and from PII. Both sensitive, but both can also be influenced by savvy users.
   - Multiple instances encouraged to avoid accumulation of too much info of this nature in one place
9. Authoritative data sources
   • Sometimes known as Policy Information Points (PIPs)
   • PDPs and reputation scoring can rely on authoritative attribute data, thus supplier of this data has to be trusted
   • The way the data was collected in the first place has to be trusted

10. Policy Authorities
    • Where do the policies that drive various PDPs come from?
    • Are they dynamic or field upgradeable?

    • Many TAS³ components are driven by business models, so whoever programs these and has ability to update the installed base, has a lot of power
    • Dynamic BPM where the actual model itself can change and be propagated instantaneously to the consumers of the model

12. Ontology authorities
    • When trying to compare apples to apples, e.g. to make authorization decision, the authority that defines the equivalence classes of terminology can control the outcome.
    • Field upgradeable or dynamic ontologies are likely to be used, thus it matters what authority lies behind them.
    • This threat is similar to the process model one

13. The domain name system
    It may arise that in some situations integrity of DNS will affect trust. Usually we should be able to avoid relying on this by using digital signatures, but there may be special cases, e.g. error situations where signature is not applied, which could then open the door to phishing or hijack attacks. Please note that the audit dashboard, while probably trusted by the user, need not be trusted in process of making any access decisions. The dashboard can, however, be one of the channels from which the reputation system gets its information.

8.4 Principles that Trust Network Should Adopt

A TN should adopt following principles:

1. Personal Data should only be collected and/or processed for fair and legitimate business purposes.
2. The purpose(s) for collection must be clearly specified.
3. The collection related to those purposes must be relevant and non-excessive.
4. Personal data must be accurate and, where needed, up-to-date.
5. Use, and subsequent use, of personal data cannot be incompatible with the purposes specified and should be with the consent of the data subject
6. Appropriate security (technical and organizational) measures against
7. Unauthorized/unlawful/accidental access; modification, disclosure, destruction, loss or damage to personal data must be in place.
8. Controllers and processors have duties to maintain confidentiality of information.
9. Sensitive data may be subject to greater restrictions.
10. Data subjects have the right to know what types of data are being maintained and have the right to access and correct personal data.
It should be noted that consent often bears important adjectives of clear, unambiguous or explicit. From a technical point of view, this requires that the user “opt in” to the collection of personal information.

Questions a Trust Network Member Has to Answer

1. Are you collecting/using PII as part of the service?
2. Do you have a privacy policy that you are bound to follow?
3. Do you use PII for any purpose other than providing the service?
4. Do you get my consent or let me opt out before my information is used for other purposes than providing the specific service?
5. Do you share my information beyond your company or family of companies?
6. Do you get my consent or let me opt out before your share my information with any other company not needed to provide the specific service?
7. Do you allow me to manage these preferences over time and change my options?

8.4 Privacy Architecture Elements

1. Identify why information is needed
2. Provide appropriate notice and obtain consent for use of information
3. Limit information collected to that which is required for the legitimate business need
4. Limit access to information to those that need it for the business function
5. Retain information only as long as reasonably needed to complete the business function and securely delete it (or possibly anonymise it).
6. Secure information as required in a manner proportionate to its nature and sensitivity
7. Maintain the integrity and accuracy of the information
8. Provide access and possibility of correction
9. **Operational aspects**

9.1 **Searching the right service (provider)**

1. User queries the TAS3 registry on the basis of functionality she is searching.

2. TAS3 Registry returns matching Service Providers.

3. User starts TPPN with the matching Service Providers.
   3a. User delegates the data manager to run the TPPN with the Service Providers (optional).

   Step 3a. User may delegate the TPPN to the Data Manager. In which case the audit update to the External Audit comes from the data manager. The notification of the
   External Audit by either the User/Data Manager + Service Provider is part of the TPPN protocol. The consistency of the Interaction Audit Trail forwarded by the User/Data Manager and the SP needs to be verified. In case of inconsistency, red flag must be raised.

4. All parties report the Audit Trail of the TPPN interaction to the External Audit.
   4a. User keeps audit of the TPPN result with data manager.
   4b. SPs keep an audit trail of the TPPN interaction (optional)

5. User starts interacting with the selected service provider.
9.2 User populates his Personal Data Store & data manager validating data

1. The user populates the PDS which is run by the data manager. The user delegates the authority to the data manager to validate his claims.

2. The data manager puts the claims into the personal data store and logs this activity in the Audit Guard.

3. The data manager contacts the necessary institutions for the validation of the user's claims. This requires a process like the Qualifications Assessment and Certification Process developed by Kenteq. The data manager has to have some convincing evidence of its authorization to execute such a process.

4. A notification can be given to the user if necessary. (optional)
9.3 TAS³ TPPN registration / Interaction with SP

1. User queries the TAS3 registry on the basis of functionality she is searching.
2. TAS3 Registry returns matching Service Providers.
3. User starts TPPN with the matching Service Providers.
   3a. User delegates the data manager to run the TPPN with the Service Providers (optional).
4. All parties report the Audit Trail of the TPPN interaction to the External Audit.
   4a. User keeps audit of the TPPN result with data manager.
   4b: SPs keep audit trail of the result of the TPPN.
5. User starts interacting with the selected service provider.
6. Further interactions are logged in the SP Audit Guard and the External Audit

Note to Step 4:
The notification of the External Audit Entity by both the User/Data Manager + Service Provider is part of the TPPN protocol. The consistency of the two audit trails forwarded by the User/Data Manager on the one hand, and the SP on the other hand needs to be verified immediately. In case of inconsistency, red flag must be raised. Note to Step 4b: They may in first instance also keep the reasons for the result of accept/deny (i.e. the complete protocol), but only for a limited period of time. If they wish to keep this information for an extended period they may only maintain this information in a pseudonymous form.
9.4 TAS³ Connector - Enabled Data Downstream

1. Patient initializes the general practitioner with his medical history with a policy.
2. General practitioner stores the medical history in a Global Medical Dossier (GMD).
3. Patient is brought to the emergency and is unconscious.
4. Emergency SP requests from the appropriate GP the SumEHR. (Patient Summary)
5. Emergency SP receives the SumEHR with the appropriate policy.
6. Emergency sends tissue and blood samples to a LAB SP, together with an extended extract of the SumEHR and the appropriate policy.
7. The LAB SP sends to two sublabs out of the TAS3 ecosystem. For each an appropriate extract of the SumEHR and the relevant policy is sent, along with the corresponding samples.
8. The TAS3 Legacy Connector de-tassifies the information and the policies, and finally forwards the information to the legacy service provider.
9. The legacy SPs apply the relevant regulation (e.g., audits to the transferred information) and completes its tasks.
9.5 TAS³ Connector – Enabled Data Upstream

1. The legacy SPs send back their analysis results. These may include policies that need to be enforced.
2. The analysis results and policies are TASSsified and sent back to the LAB SP.
3. The LAB SP returns the aggregated results and the updated and aggregated policies are sent back to the Emergency SP.
4. The patient is treated according to analysis results.
5. The Emergency SP notifies the General Practitioner of the treatment with the corresponding policy.
6. The General Practitioner WS stores the updates and policy and logs the relevant information.
10. Limburg Employability Platform

Synergetics which in TAS³ develops a Personal Data Store (aka employabilityPortfolio™) for managing personal employability related data recently signed an agreement with the province of Limburg (NL) for a Regional Employability Platform. The platform involves multiple regional stakeholders all involved in employability processes with learners, workers, unemployed and jobseekers.

Phase I of the project consists of stakeholder engagement plan which will investigate, describe and commit the specific challenges, needs wishes and wants of the various stakeholders. The result will be fed into this document, which by M24 reporting will be considered an independent document in the New DoW.

6 april 2009
Limburg Talent Region

External
- Demand -

Employability Platform Limburg

Internal
- Supply -

Build Infrastructure

Consortium stakeholders

Public-private-Partnership

Regional Cooperation Financing

provincie limburg

Win-Win for all stakeholders

Stakeholders:

Employees/ Job-seekers Learners

• input data once, use many
• Dynamic en automated medium
• User control of information access
• Facilitates training & career planning

Employers

• Easy access to coherent information
• Facilitates training & career planning
• Meaningful matching
• Harmonised employability information
• Facilitates HR training processes
• Provides career services to employees

Education
Government
Interest Groups
(labor unions, industry sectors, ...)

• Facilitate employers and employees
• Match Job seekers
• Transfer education - workplace
Consortium (proposal phase)

- COLO – kenniscentra
- Labour Union (FNV.nl)
- Province of Limburg
- City of Maastricht
- Higher Education
- Ministry of Social Affairs
- Public Employment Services
- Employer organisations