SEVENTH FRAMEWORK PROGRAMME
Challenge 1
Information and Communication Technologies

Trusted Architecture for Securely Shared Services

Document Type: Deliverable

Title: # TAS³ Identifiers and Discovery

Work Package: WP2

Deliverable Nr: D4.1

Dissemination: Public

Preparation Date: June 30, 2011

Version: 2.1

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

	Beneficiary Name	Country	Short	Role
1	KU Leuven	BE	KUL	Coordinator
2	Synergetics	BE	SYN	Partner
3	University of Kent	UK	KENT	Partner
4	University of Karlsruhe	DE	KARL	Partner
5	Technische Universiteit Eindhoven	NL	TUE	Partner
6	CNR/ISTI	IT	CNR	Partner
7	University of Koblenz-Landau	DE	UNIKOL	Partner
8	Vrije Universiteit Brussel	BE	VUB	Partner
9	University of Zaragoza	ES	UNIZAR	Partner
10	University of Nottingham	UK	NOT	Partner
11	SAP Research	DE	SAP	Project Mgr
12	EIfEL	FR	EIF	Partner
13	Intalio	UK	INT	Partner
14	Risaris	IR	RIS	Partner
15	Kenteq	NL	KETQ	Partner
16	Oracle	UK	ORACLE	Partner
17	Custodix	BE	CUS	Partner
18	Medisoft	NL	MEDI	Partner
19	KIT	DE	KARL	Partner
20	Symlabs	PT	SYM	Partner

Contributors

	Name	Organisation
1	Brendan Van Alsenoy, Andreas Pashalidis	KUL
2	Michele Bezzi	SAP
3	David Chadwick	KENT
4	Jeroen Hoppenbrouwers	SYN
5	Sampo Kellomäki (main contributor)	SYM
6	Gilles Montagnon	SAP

Table of Contents

Identifiers and Discovery Executive Summary

This document describes identifier and token issuance considerations and services. It describes two principal categories of privacy friendly identifiers, the persistent and transient Name IDs that are difficult to guess and not shared across participants of a federation.

The data model of the federation databases is discussed and it is noted that the databases of an Identity Provider, discovery, linking service, and ID Mapper are highly similar and that a common implementation choice is to have the same system entity offer all these interfaces from a single database. However, to support separation of duties, an alternate model with separate databases and controlled synchronization is presented as well.

The issuance of tokens by an ID Mapper in various specific situations is discussed. The properties of the tokens and the necessary policy and audit safeguards are presented. We cover user-present, pre-authorized, and not-present cases as well as token based delegation. We compare delegation of attributes to token based delegation and say why the former is the preferred approach.

A conclusion about token revocations is that most short term tokens do not need a revocation mechanism. In case of the Identity Mapper (IM) bootstrap token, which due to the logistics has to be long lived, specific risk mitigation strategies are adopted. In any case all derived tokens will be short lived and authorized upon token creation, effectively providing revocation of the IM bootstrap.

The role of the Registry Server in locating per-user resources is discussed. We also discuss how the Registry Server integrates with the On-line Compliance Testing and Trust Network's partner intake process.

Finally an exposition of the Credentials and Privacy Negotiation functionality is presented, including user interface driven front channel and discovery driven back channel approaches. Gap analysis is provided to see how the two phases of the back channel approach, discovery and service call, satisfy the essential needs to communicate policy pledges and policy requirements.

1 Introduction

This document specifies the TAS³ Discovery function, see [TAS³ARCH] Fig-2.2, comprising of ID Mapper, Registry Server, Linking, and Trust and Privacy Negotiator. The discovery function aims at solving two problems: issuance of credentials, or tokens, for specific transactions such that wild card credentials can be avoided; and finding out where a given service is hosted for a given user, so that it is possible to host the same service for different users in different places, promoting competitive market place for the Service Providers.

This solution addresses Reqs. D1.2-2.3-BMs (discoverability), D1.2-2.14-Priv (pseudonymous design, attribute pull enablement), D1.2-3.11-UPAPD (the policy discovery aspect), D1.2-7.17-Increm (incremental release of credentials), D1.2-3.12-SPManifest (discovery based on privacy policy), D1.2- 3.13-BPAdapt (business process adaptation by coordinating discovery), D1.2-3.14-PIIPolicyDisco (discovery keyed on adequate policies), D1.2-3.15-SecPreserve (discovery of policies so that business process can be adapted preserving certain policy properties), D1.2-4.2-BPPrivacy (use of pseudonyms in Business process).

An important architectural property of the discovery function is that it allows pseudonymous operation, thus avoiding leakage of unencrypted correlation handles and improving privacy protection in complex, inter-calling, systems.

The discovery function also addresses user not present transactions, provides for some delegation scenarios, and acts as a registry of services playing a part in Service Provider compliance validation business process.

1.1 Format and Properties of IDs

An Identifier is a special type of identity attribute, in that on its own, it is able to unambiguously identify a single user of the federation. It is for this reason that TAS³ chooses not to use globally unique multi-directional identifiers that are shared between service providers, since this makes it very easy for them to link their databases together and correlate the user's actions. X.500 distinguished names and OpenIDs are examples of these multi-directional globally unique identifiers.

As specified in [TAS³PROTO] (also Annex A of [TAS³ARCH]), the primary token format of TAS³ is a SAML 2.0 Assertion (A7N). In SAML 2.0, the users are identified by a Name ID, which can come in several variants. TAS³ chooses to use two principal kinds, see [SAML2core] sections 8.3.7 and 8.3.8, and assigns them (at least) the following properties:

Persistent Name ID Whenever the Identity Provider (IdP), or discovery, and federation partner talk about the same user (e.g. an IdP vouching authentication of the user to an SP in a SSO transaction), they always use the same identifier, across the sessions. i.e. it can be understood that it is always the same user. Typically the SP might maintain a database and use this identifier as a key. Note that it is the SP that requests this type of identifier, otherwise the IdP will use a Transient ID in order to better protect the user's privacy and anonymity.

Additional properties are required:

ID MUST be difficult to guess, ideally it should be at least 128 bit random number.

The ID MUST be uni-relational, meaning that a different identifier must be issued by the IdP for each SP.

The ID used for the same user towards different parties MUST NOT be easily inferable (e.g. it MUST NOT be the same, statistically related, or guessable from the other ID).

A globally Unique ID or a National ID would satisfy the permanence criteria, but would not satisfy the uni-relational property, since it is multi-relational. This has adverse privacy consequences due to the ease of database linking.

The essence of our approach is that while a User may choose to (or be required to) allow one party to track his actions across sessions (hence persistent), this should not imply that this party can compare notes with other parties. The persistence property allows tracking by a legitimate party, while the "not easily inferable" property keeps the parties from colluding or comparing notes.

The e-Government sector specific ID approach, e.g. a tax number is different from a social security number is different from a health number and so forth for all the government agencies, comes quite close to what we mean by persistent and uni-relational.

Persistent Name ID is often called a pseudonym.

Transient Name ID This identifier format allows identification of the user for the duration of one session only. It has similar properties to the persistent identifier but is not persist across sessions, thus providing better privacy guarantees than the Persistent Name ID, at the cost of not allowing the SP to maintain a tracking database without the user's explicit consent for the release of a persistent attribute. Often the motivation for using a Transient Name ID is exactly to prevent such databases from being created.

The most important privacy property, which both persistent and transient ID satisfy, is that the User's identifier towards two different parties must be different and not easily inferable. This provides a technical protection against cross site collusion. The difference between the persistent and transient is that the former allows the (authorized) site to correlate the user's actions across sessions, i.e. the same user visiting the same site repeatedly, while the transient ID makes this form of correlation difficult without the user specifically releasing an attribute that can perform the same function.

It is important to understand that the persistent and transient properties alone do not provide significant privacy benefits if they are shared across web sites even momentarily or even encrypted. Even a single instance of such sharing would provide the sites with a correlation handle despite their encrypted or transient properties. Just a single occurrence of a correlation handle allows all (persistent) past and future click-trails to be linked across the web sites.

Other kinds of Name IDs are possible and allowed, depending on agreement within the Trust Network. However, it should be fully understood what the privacy and other properties of the chosen ID

1.2 Who Issues IDs

In TAS³ the Name IDs that pass over the protocol flows are issued either by an IdP or the ID Mapper (IM) of the discovery function. The internal IDs that SPs may have issued are generally not passed over the wire (but see SAML 2.0 ManageNameID protocol [SAML2core] for a possible exception).

If the authentication scheme at the IdP involves a User ID, such a User ID is considered to be part of the authentication credential. Allocation of the User IDs is a private matter of the IdP and the User Ids are never communicated to other parties or passed over the wire. It is possible that an IdP collaborates with some national eID scheme outside the scope of the TAS³ architecture. In that case the eID would probably be allocated by the national scheme, but the eID would not be communicated by the IdP to the Service Providers. The eID would only be used towards the IdP to authenticate the user and from that point onwards the IdP allocated pseudonyms or transient IDs are used.

1.3 Supported Flows

Reader should refer to [TAS³ARCH], section 3 "Core Security Architecture" for description of the supported protocol flows.

2 Federation Data Model

One of the fundamental principles of the identifier management is the use of federations. In order to implement persistent and pseudonymous federations, the IdP and IM have to keep state.

In general, a federation table for an IdP has mappings of form

```
User at IdP1 --> [ encrypted pseudonym of user at SPA,
                   encrypted pseudonym of user at SPB,
                   ...
                   encrypted pseudonym of user at SPN ]
```

If the table serves only one IdP and thus the IdP EntityID is implicitly known, then the table simplifies to have columns

```
 “User ID” “AuthN Cred” “SP EntityID” “Enc.pseudonym of user at SP”
 --------   ----------  ------------ ----------------------------
 Koerkki    salainen    A.example.com     enc_A(123)
   --’’--    --’’--      B.example.com     enc_B(456)
   --’’--    --’’--      C.example.com     enc_C(246)
   --’’--    --’’--      IM.example.com    enc_IM(789)
 Tester     secret      A.example.com     enc_A(357)
   --’’--    --’’--      IM.example.com    enc_IM(579)
```

where "enc_A", etc., means encryption such that only A can decrypt (e.g. with A’s public key or shared secret only known to A). The encryption should also include a nonce component to avoid two encryptions of the same data looking the same. This is to protect the pseudonyms against exposure at middlemen and while in the database.

The federation table for an IM needs similar mappings

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

If the table serves only one IM, then the table simplifies to have columns

```
“User’s pseudonym” “SP EntityID” “Enc. pseudonym of user at SP”
----------------  -------------- ----------------------------
  789IM           B.example.com           enc_B(456)
  789IM           C.example.com           enc_C(246)
  579IM           B.example.com           enc_B(791)
  579IM           C.example.com           enc_C(913)
```

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 the user ID and sometimes by the SP ID, or both.

The IM needs an 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 Authr	C.example.com
789IM	HR Authr	B.example.com
579IM	Role Authr	C.example.com
579IM	HR Authr	B.example.com

but other more general realisations can include data needed for Credentials and the Privacy Negotiation phase of Discovery. These will be explored later in the section on Credentials and Privacy Negotiator.

An IdP may have a limited form of this table to cover the necessity of emitting an IM bootstrap token during single sign on (SSO).

All parties - IdP, IM, and SP (Front End or Web Service) - need to maintain some metadata about each other. Such metadata may include SOAP endpoints, protocol profiles and bindings to use, etc., see [SAML2meta].

3 ID Mapper: Issuing Specific Tokens

As specified in [TAS³PROTO], the ID Mapper functionality is realised as part of the Discovery Service [Disco2]. It MAY also be realised using the Security Token Service (STS) role of [WSTrust] or the Identity Mapping Service described in [SOAPAuthn2] (this being typical in some delegation cases using the People Service [PeopleSvc]).

The tokens issued by the ID Mapper often pass through intermediaries due to the logistics of the discovery and token based delegation flows. To maintain a fully pseudonymous architecture, the tokens that as passed through intermediaries MUST be encrypted using the public key of the intended consumer of the token.

3.1 User Present Case

The "User present" scenario is the base case of the TAS³ architecture. It assumes the user is interacting with the system in (near) real time and instructs it to act according to his wishes. Such instruction simultaneously provides a command of the action and consent for it being performed, including any sub-actions that may be needed for its performance. From the audit trail perspective, it is essential that the User's manifest will and consent is captured. The audit trail becomes stronger when it can be shown that the user was tactically present and aware of the action being taken. Presence is of course relative when a web service call is made somewhere deep in the infrastructure, but if it can be shown that the user had an active front channel session and that from this session emanated a command to perform an action which caused the audited action to be performed, the presence of the user in the audited action is established.

In the user present case the token issuance is straight forward: the IM bootstrap token is generated by the IdP and included in the Single Sign-On (SSO) or web services layer authentication as an attribute in the assertion, as seen in Fig-3.1

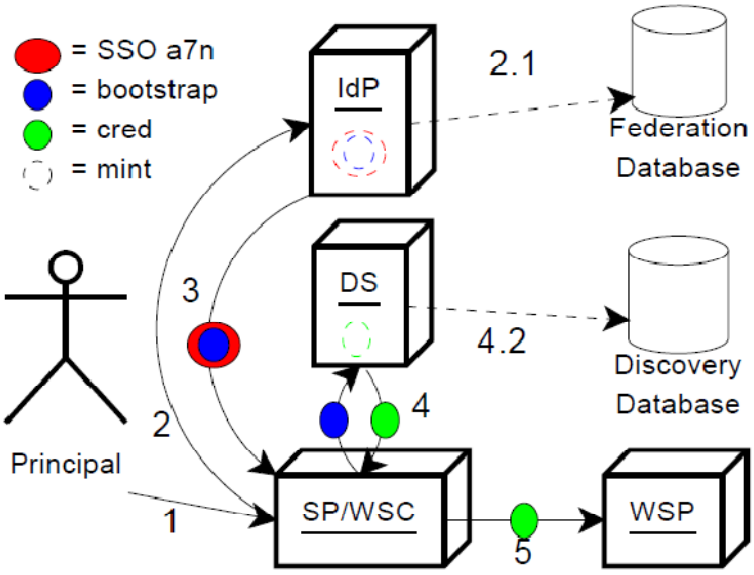

Figure 3.1: Single Sign-On (2,3), Discovery (4), and call to WSP (5). The blue ball represents discovery bootstrap.

The Service Provider consumes the SSO or authentication assertion and extracts the IM Bootstrap. When it needs to call a web service, it uses the IM Bootstrap to call the Id Mapper service, which usually is a Liberty Discovery Service, but could also be a WS-Trust Security Token Service (STS), to generate the access token which is then used towards the payload web service.

3.1.1 User Present Evidence

The "user present" aspect is captured by the fact that the IdP attests direct authentication of the user in the same session and in the not too distant past. Generally the expiry time of the tokens is set accordingly.

This works well under the assumption that the web services call happens relatively soon after the SSO, but fails to provide an adequate solution for long-lived SP sessions. For long lived sessions, the temptation is to increase the expiry time of the IM bootstrap token, but this weakens the "user present" aspect and ultimately some cut-off must be determined in the Governing Agreement of the Trust Network.

A better alternative, is to refresh the IM bootstrap token by performing a new SSO transaction with the IdP: as long as the user's session at the IdP is still valid, the refresh will not cause any user observable effect (apart from the redirection flicker), but this will return a new IM bootstrap with renewed expiry time.

3.1.2 Interaction of IdP and ID Mapper

Since the IdP and ID Mapper (IM) need to maintain federation databases and need to communicate with one-another for the purposes of arranging the IM bootstrap, it is not uncommon for the same organization to operate both the IdP and the IM so they can share a database. The Liberty Identity Mapping Service is also often co-hosted with the IdP and discovery and shares their database. In the shared database case, any suitable arrangement can be used and the standards tend to be silent on the issue.

If, however, there is desire to keep the databases of the IdP and IM separate, e.g. for separation of duties purposes, then some communication needs to happen between the IdP and the IM to arrange the bootstrap issuance - which necessitates that the IdP learns the User's pseudonym at the IM.

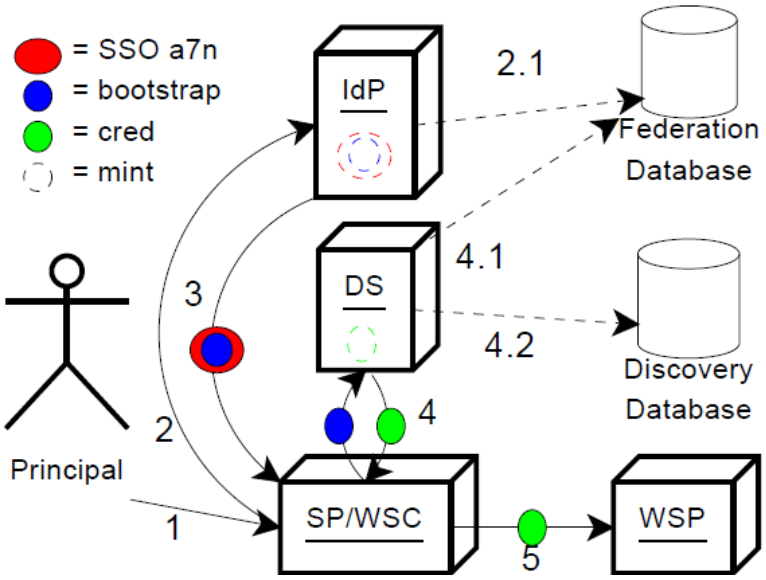

Figure 3.2: Discovery Service makes back channel query (4.1) to map the Id received in the bootstrap to a key that can be used to query other databases (4.2)

Fig-3.2 depicts a scheme where the IM updates the IdP database. This is advantageous when the IdP is a COTS software package that cannot be altered. It does require a documented database interface, though. Many IdP products use an LDAP repository as a database and work in a straight forward way. Currently there are no standards regarding the database schema.

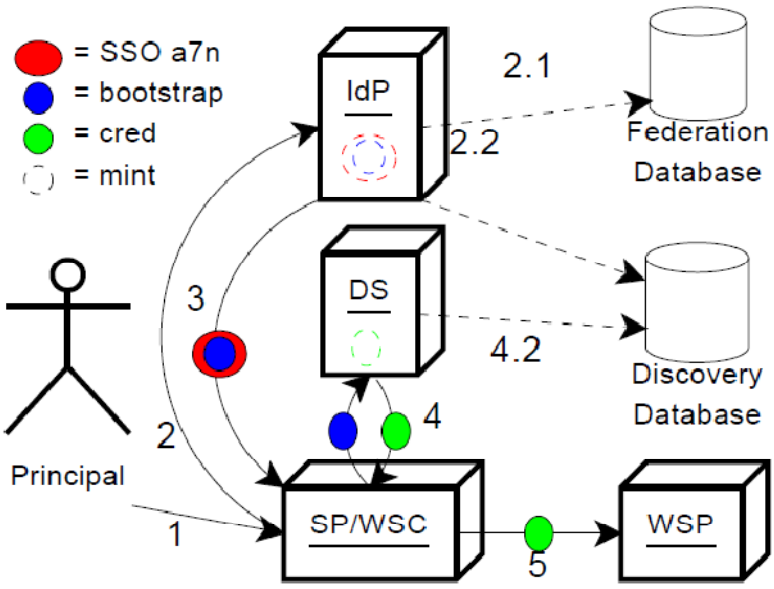

Figure 3.3: The IdP makes a back channel query (2.2) to map the Mobile Subscriber ISDN Number to the Id understood by the discovery (4.2). The Id is passed in the bootstrap.

Fig-3.3 shows the opposite arrangement where the IdP consults the IM database to obtain the IM bootstrap.

3.2 Pre-authorization by the User

The pre-authorization case involves the user being present at some point in time and indicating his will that in future some operation is performed on his behalf. Such consent and authorization is captured by technical means so that the transaction can be acted in a later time producing an audit trail that unequivocably demonstrates that the user authorized the transaction, albeit in (possibly distant) past and without being present at the moment of the transaction.

The pre-authorized case can easily be implemented by delegation of authority, whereby the user delegates a subset of his authorisation attributes to a service provider (or to another user), to act on his behalf in the future. This is the primary approach that has been adopted by the TAS³ infrastructure. When it is user to user delegation, we use the delegation by invitation method, as described in D7.1, since the user (the delegator) does not know the uni-relational identifier of the other user (the delegate). When it is user to service delegation, this does not need to be by invitation since the user can determine the permanent omni-directional identifier of the service and can delegate directly to it via the Delegation Service (again as described in D7.1). This is the preferred approach since it provides fine grained delegation of authority. The delegation can be for an extended period, as determined by the delegator, and he can revoke the delegation whenever he wants to.

An alternative approach that can also be adopted, is where the user delegates the right to use his entire set of uni-relational identifiers to other service providers via the IM. This requires the IM to issue a long-lived bootstrap token that can then be used as if the user was still present This pre-authorized case is implemented by the first SP, which already has in its possession the uni-relational identifier of the user to itself, using the IM bootstrap token to obtain another token containing the user's encrypted uni-relational identifier to another SP, just prior to the pre-authorized transaction being forwarded to the other SP. The request for the encrypted token must indicate that this is the pre-authorized situation and the discovery will check appropriate policies to see if the contemplated pre-authorized transaction can go forward. This represents a depth of defence as a similar check can and should be made at the Service Provider. However this alternative approach is not the preferred solution, since the encrypted token acts as a correlation handle between the two SPs, which is something TAS³ is trying to avoid. Furthermore the IM has to trust that all the SPs are acting honestly when they ask for one of the user's encrypted uni-relational identifiers, since they have no proof that the user actually authorised this, unlike in the delegation approach where the user has to actively perform the delegation.

3.3 User Not Present Case

The user not present case generally involves situations where legal or contractual justification can be invoked to authorize a transaction to go forward. The TAS³ authorization infrastructure MUST check policies to determine if indeed it is legally or contractually acceptable to perform any give transaction without the user's consent or presence.

An interesting identity management problem that arises in the user-not-present case is how the user can be accurately identified. In the case of the user having a globally unique multi-relational ID, this may be relatively easy, but in case of fully pseudonymous identity management, the authorized initiator of the transaction may not actually know the pseudonyms (uni-relational identifiers) that are needed to complete the transaction.

To resolve this impasse, the identity mapping functionality described in Section 3.2 above can be used to convert a pseudonym that the initiator knows to a pseudonym that he needs. As such this generic conversion ability is a serious privacy threat. The authentication of the initiator and the authorization implemented by the IM must be of the highest standards. The Trust Network should invest significant planning and audit to make sure that the identity mapping does not become dangerous backdoor. It is for this reason that the use of a delegation service is preferred, since there is no protocol specified way of obtaining a delegated attribute when the user (delegator) is not present and does not initiate this. Nevertheless lawful access requirements may require the DS to issue them with an attribute, via an administrative channel, allowing them to utilise the permissions of the user under investigation. However, this specification of this back door is outside the scope of TAS³.

If the initiator does not know any pseudonym, there are grounds to suspect that it does not have a legitimate case for performing a user-not-present -transaction. If such transaction is legitimate, none-the-less, then initiator judicially requests a pseudonym for the user to be disclosed by a party (identity discloser) that can check the legitimacy of the request and understand which user is meant. Such identity discloser could also provide search and browsing interfaces for finding the user. The caveat about grave privacy and security concerns mentioned in identity mapping, above, doubly apply to the identity discloser.

All tokens emitted via identity discloser MUST be marked as such and should not purport to be user present tokens. The authorization at the SP in this case is based on the legitimate-user-not-present access marker. The initiating party MUST be identified in each user-not-present transaction. This could be achieved by regular authentication of the initiator using authentication mechanisms specified in [SOAPBinding2] or it could be expressed using Subject Identity in token based delegation.

Once the user-not-present transaction in principle is authorized, the problem remains as to which type of token should be issued to the initiator. To keep authorization narrow, the token should be directly destined to the SP to which access is authorized. However, as there is no easy way to know if the SP in its turn needs to call on other SPs to perform its function, it is necessary to provide an IM bootstrap as well.

The problem with providing an IM bootstrap in user-not-present transaction is that it may authorize too wide access. The specification of this back door is mechanism is outside the scope of TAS³.Meanwhile, the best that can be done within TAS³ is to ensure extended audit to detect and curtail any abuse.

3.4 User Revokes Authorization/Delegation

Revocation of delegated authority is supported by the Delegation Service, and the user can revoke the attribute(s) that he has delegated at any time. The Delegation Service will not issue any more delegated attribute tokens to the delegate, immediately upon the revocation taking effect.

In the ID mapping case it is more difficult. Revocation (or suspension) of identity prevents further issuance of any tokens on behalf of the user (delegator), but it also stops the user from performing any further transactions as well. This is another reason why "delegation of identifiers" is not the favoured approach.

In either case, the problem that remains is what to do with the already existing tokens issued to the delegate and that are "in the wild". Currently TAS³ does not foresee a token revocation mechanism for these. The intent is that all issued tokens are sufficiently short lived that a revocation mechanism is not needed. Even if a conventional revocation list was issued, there is always a latency between the delegator performing the revocation, the revocation list being issued, and the relying party downloading the latest list. Consequently, if the short lifetime of the issued tokens is similar to the revocation latency, then there is no advantage in issuing revocation lists.

The notable exception is the pre-authorized case for identity mapping. As described in Section 3.2, this is solved by only tolerating long lived IM bootstrap tokens. All other tokens are issued from this bootstrap on as-needed basis and are short lived. This achieves the same net effect as a revocation list check.

It should be noted that for PKI certificates we do foresee the use of OCSP [RFC2560] or certificate revocation lists (CRLs). However as PKI is not used for end user identity (except, perhaps by an IdP to authenticate the user), this revocation processing is not in scope of this document.

4 Linking Service

The Linking Service (LS) is a component under the user's control that allows him to link together his various accounts at various IdPs. The Linking Service holds, for any user, a set of tuples containing: a uni-relational identifier to the LS, the IdP that issued this, the Level of Assurance provided by the IdP, and the attribute types (but not values) that the IdP is capable of issuing as assertions for this user. When the user wishes to aggregate his attributes from multiple IdPs in a single transaction with an SP, it is the Linking Service which requests these attribute assertions. All attribute assertions are issued encrypted to the SP, so that the LS cannot read them, and contain the same transient identifier for the current session so that the SP can validate that they all belong to the same user.

Whilst it is technically possible for the various IdPs to collude together to see which transient IDs were issued to which users in which sessions, thereby allowing them to link together the user's accounts at the various IdP, TAS³ relies on the Trust Network Agreements and the Law to forbid this type of covert activity. It provides no greater risk of account correlation than the passing of encrypted tokens between SPs described in section 3.2.

TAS³ believes that moving towards anonymous credentials, as exemplified by U-Prove and Idemix, are currently the only technical way to significantly reduce the risk of covert correlation of accounts

5 Registry Server

The registry server is part of the discovery functionality. Its purpose is to maintain a database of Service Providers (SPs), a mapping of which SP provides which service to which user, and a federation database specifying the User's (encrypted) pseudonym at each SP with which the user has a relationship. Covert use of this database will obviously allow SPs to link together a user's account with them.

Such database is needed to ensure that Users have a choice of SP or at least to ensure that different Users can get the service from disjoint sets of SPs. Such situations commonly arise when Users of one organization start using services of another organization. For example, if User uses procurement application at Supplier, the User's roles still need to be fetched from his home organization. The registry server enables the supplier to dynamically understand where the role authority is for each of the Users that use the system.

5.1 On-line Registration Steps

Given the fully pseudonymous design of the TAS³ architecture, populating the registration database is non-trivial. Fig-5.1 presents a dynamic solution where the User initiates the registration. In this case the (encrypted) pseudonym for the user at the SP is pushed to the registry by the SP itself.

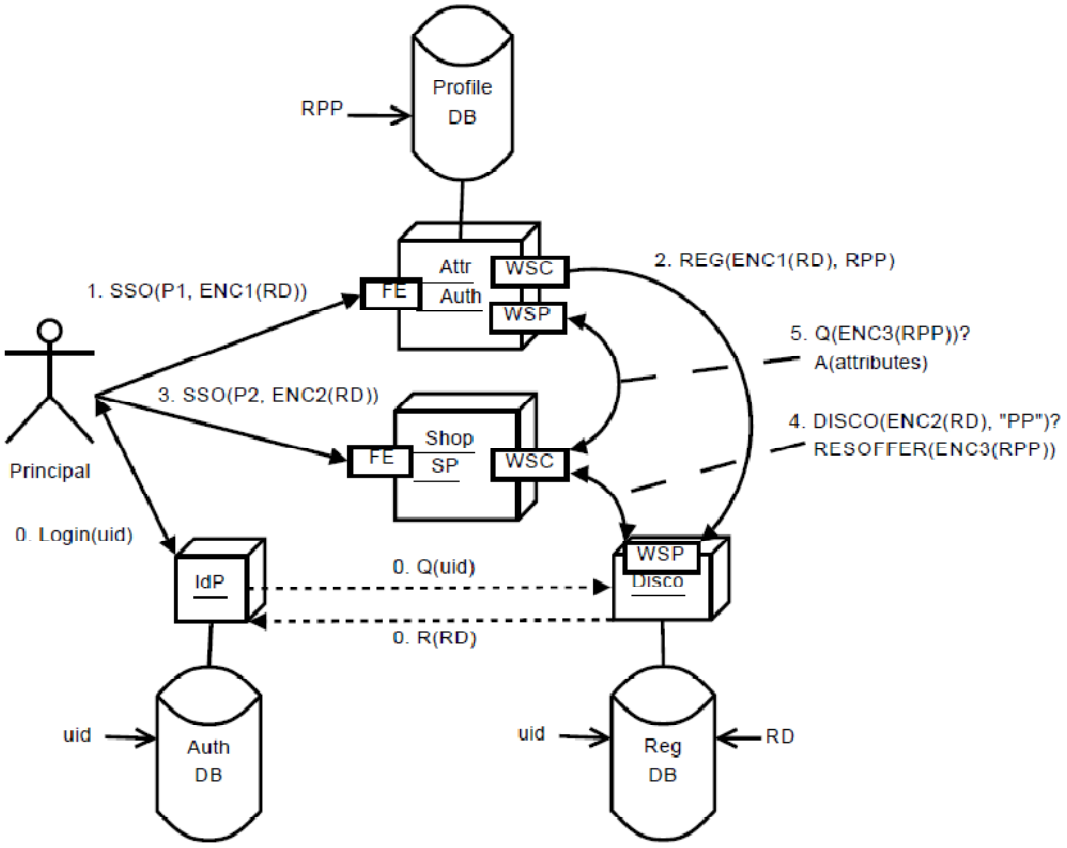

Figure 5.1: Discovery Registration Using Front Channel Interface.

The registration steps are as follows:

1. User visits a service (the Attribute Authority in Figure 5.1), perfoming a Single Sign-On (the user has to login to his IdP if he has not already done so, - this is message 0), thus establishing his pseudonymous identifier P1 at the service. The SSO message contains the discovery bootstrap token ENC1(RD)
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 the user's identifier (RPP) in a subsequent web service call.
3. User instructs the front end to perform an action that triggers a web service call to a second service (the Shop SP) where via SSO the Shop SP receives the user's pseudonymous identifier with it (P2) plus the discovery bootstrap token (ENC2(RD).
4. The Shop SP uses the disco bootstrap token to make a back channel web services call to the user's discovery service to obtain the encrypted token (ENC3(RPP)) for communicating with the Attribute Authority (known as service PP).
5. The actual back channel web service call is made from the Shop SP to the Attribute Authority with the correct identity (ENC3(RPP)) allowing the latter to return the user's attributes to the Shop.

5.2 Bulk Registration

The dynamic registration model may not be appropriate in all situations. Bulk registration offers an alternative, but presents a problem as populating the registration database will require identification of the Users, i.e. the pseudonym of the user at each SP needs to be found. The problem is similar to the user-not-present case, see Section 3.3. The technical solution is essentially the same as in Section 3.1.2.

5.3 Articulation of Registration with Compliance Validation

The Registry Server plays an important role in the On-line Compliance Testing as it is the mechanism by which the testing infrastructure finds out about new system entities to test and their relevant metadata, test cases, and declared policies.

The Trust Network level new organization intake process integrates a registration step.

6 Credentials and Privacy Negotiation

The Credentials and Privacy Negotiation (CPN) can happen via either the front channel or the back channel. In the former case it can involve a choice of Front End and mutually assuring steps shown through the user interface. The purpose of the steps is to climb a ladder of trust whereby each party progressively reveals more about itself until trust can be established between the parties. Front channel Credentials and Privacy Negotiation is a user interface intensive process and may need to be modelled at a business process level. As it is likely to be extremely implementation and deployment specific, it is not discussed any further here.

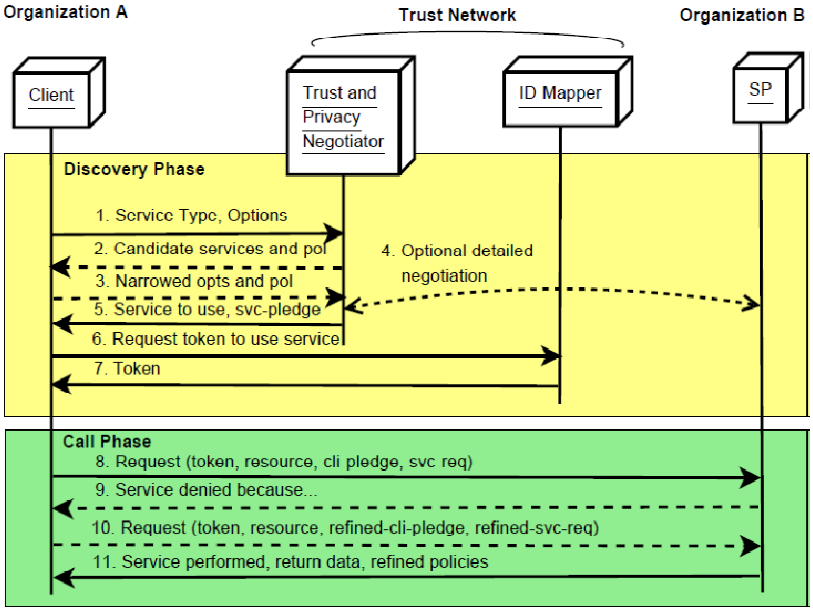

Figure 6.1: Two phase Credentials and Privacy Negotiation on back channel.

The CPN on the back channel involves stating trust and privacy requirements at both the client and service side and then discovering a service that matches the client. If a single match is found, the negotiation ends and the service provider is used. If multiple matches are found, the client may unilaterally choose the one that is most advantageous to it.

If no matches are found, the real negotiation begins. The client may relax some of its requirements and see if it can discover any SPs under the new terms. Alternatively, the discovery may return some SPs whose policies are close enough that it seems the client might relax its requirements sufficiently, with the trust and privacy parameters they accept, and the client then has to make a new query, promising to satisfy the refined parameters that were required, to obtain the access credentials.

The final part of the Credentials and Privacy Negotiation can be carried out once the client has chosen a service. The service request contains a request specific pledge by the client about the policies it promises to honour, if the request is performed, and with respect to the data that may be returned. The SP examines

this policy pledge and decides if it is acceptable given the request and the data it would return.

If policies are acceptable, the SP performs the request, attaching to the response additional policies that the client must honour. These policies can be only ones that are foreseen by (a) law, (b) contract, or (c) client's policy pledge carried in the request. For example, if the client's policy pledge promises to honour any data retention limitation above 5 seconds, then the service could set an obligation to delete the returned data in 60 seconds. Insisting on deletion in 3 seconds would be moot as the client never promised to honour that.

If the policy pledge of the Client is not acceptable, the service returns an error indicating why it was not acceptable or what would have been acceptable. The Client can then decide if it is willing to modify its policy pledge until it is acceptable and retry, or abandon the request. The Client can then return to the discovery phase and try to locate a different Service Provider candidate.

There are two key decisions in the last phase: (i) the Service Provider needs to decide whether the Client's policy pledge is acceptable given the nature of the request, the User behind the request, including any delegation, and the trustworthiness of the Client itself; and (ii) if Service Provider tries to negotiate on the policy pledge, the Client needs to decide whether it finds the proposed pledge acceptable given the trustworthiness of the Service Provider.

In both of these cases the trustworthiness of the other party needs to be established. This is primarily done using TAS³ trust establishment mechanisms, which tend to communicate whether the Trust Network considers the other party a trustworthy participant of the network. If the parties have specific trust requirements, beyond what the Trust Network is able to tell about the parties, then they need to do some extra work themselves. Often such specific trust evaluation will be specific to the business of the Client and/or the Service Provider, thus it is expected that they will establish their own business processes for it. For example, Service Provider may have a requirement that the mere Trust Network membership is not sufficient, thus it needs to invoke the Service Provider specific Client Intake business process to get the Client vetted and registered. If this happens in the background on the Service Provider side, the main flow is not affected. If, however, the Client Intake business process needs to be initiated by the Client, then the Service Provider needs to return an error code or policy requirement that triggers the Client to initiate the process. In this case there needs to be a private understanding about the meaning of these error codes between the Client and the Service Provider. TAS³ foresees this mechanism, but does not specify what the codes are.

Bibliography

[AAPML] Prateek Mishra, ed.: "AAPML: Attribute Authority Policy Markup Language", Working Draft 08, Nov. 28, 2006, Liberty Alliance / Oracle. http://www.oracle.com/technology/tech/standards/idm/igf/pdf/IGF-AAPML-spec-08.pdf

[CARML] Phil Hunt and Prateek Mishra, eds.: "Liberty IGF Client Attribute Requirements Markup Language (CARML) Specification", Draft 1.0-12, Liberty Alliance, 2008. http://www.projectliberty.org/liberty/resource_center/specifications/igf_1_0_specs

[Disco2] Cahill, ed.: "Liberty ID-WSF Discovery service 2.0", liberty-idwsf-disco-svc-2.0-erratav1.0.pdf from http://projectliberty.org/resource_center/

[PeopleSvc] "Liberty ID-WSF People Service Specification", liberty-idwsf-people-service-1.0-erratav1.0.pdf from http://projectliberty.org/resource_center/specifications/

[RFC2560] Myers et al., "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP", RFC 2560, June 1999.

[SAML2core] "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-core-2.0-os

[SAML2meta] Cantor, Moreh, Phipott, Maler, eds., "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0", Oasis Standard, 15.3.2005, saml-metadata-2.0-os

[SOAPAuthn2] "Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification", liberty-idwsf-authn-svc-2.0-errata-v1.0.pdf from http://projectliberty.org/resource_center/specifications/

[SOAPBinding2] "Liberty ID-WSF SOAP Binding Specification", liberty-idwsf-soap-binding-2.0-erratav1.0.pdf from http://projectliberty.org/resource_center/specifications

[TAS³ARCH]"TAS³ Architecture", TAS³ Consortium, 2011. Document: TAS³-arch-v24.pdf

[TAS³PROTO]" TAS³ Protocols and Concrete Architecture", TAS³ Consortium, 2011. Document: TAS³-proto-v?.pdf

[WSTrust] "WS-Trust 1.3", CD 6, OASIS, Sept 2006. (*** WS-Trust, STS, etc.)