

# Aggregation of Attributes from Different Authorities

David Chadwick


# Contents

- Conceptual Model
- Proposed Protocols
- Next Steps

# Attribute Aggregation

- Users typically have lots of different attributes from different providers
- User is usually known by different IDs at the different IdPs/AAs
- Only the user knows what these IDs are
- User might wish to benefit from using these multiple attributes at an SP, but how will SP know that all these different IDs belong to the same real world person?
  - E.g. Use IEEE membership and credit card when purchasing a book
- Furthermore, what are the privacy issues involved in attribute aggregation?

# Shintau Project

- 2 year project at University of Kent, funded by UK JISC, joint with Internet 2 with support from SWITCH, Globus, TERENA etc
- Objective: to define standard protocols for attribute aggregation and implement them as open source
- First stage was to capture user requirements
- Second stage was to define conceptual model, get feedback from community, then revise model
-  Third stage is to define standard protocols for conceptual model and get feedback
- Fourth stage is to implement system in open source code for free distribution to world

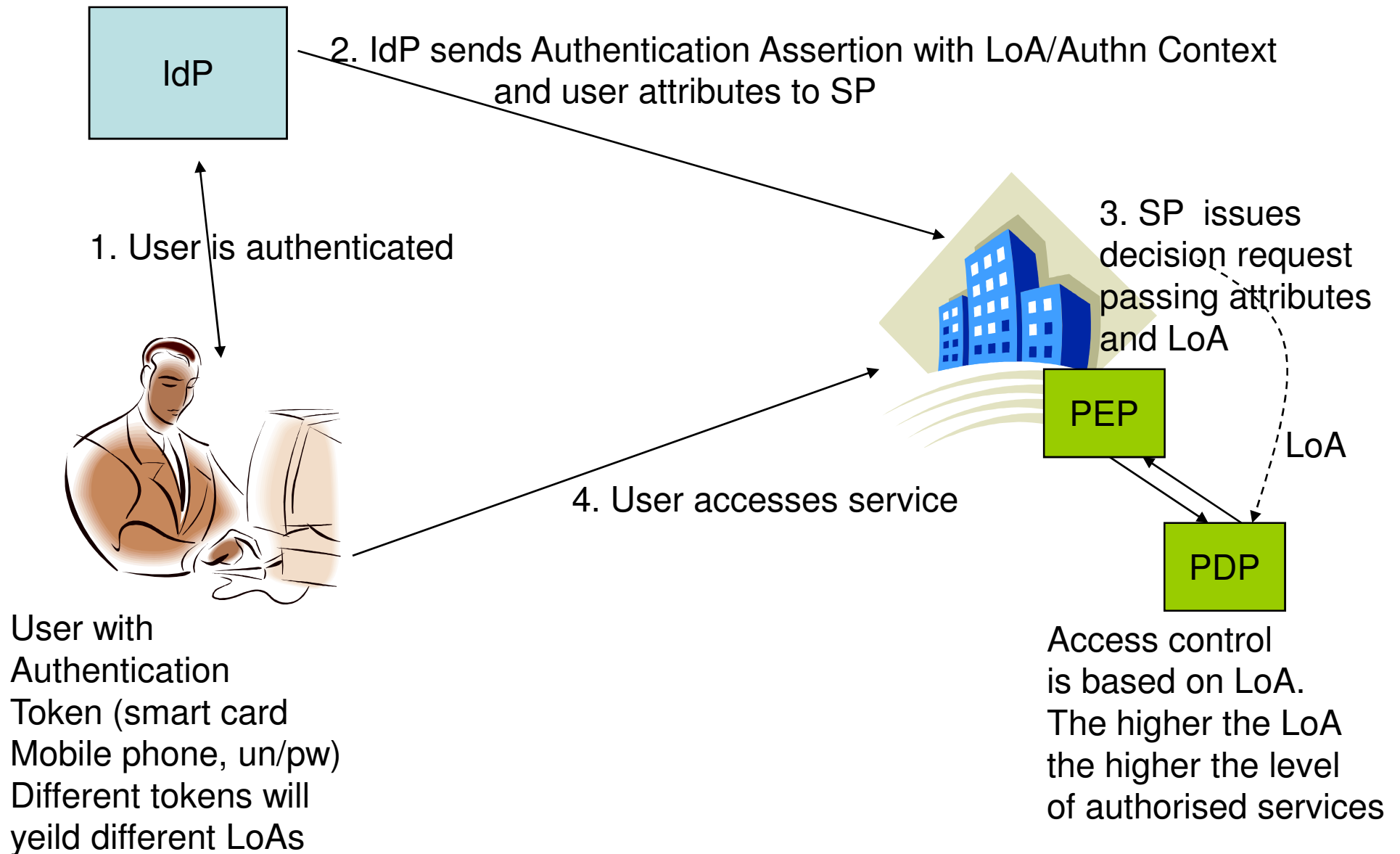
# Conceptual Model

- Introduce a Linking Service whose purpose is to hold uni-directional links between a user's attributes from different IdPs
- User will register with a Linking Service and link his IdPs together, optionally providing a Link Release Policy to say which links can go to which SPs.
- When user contacts an SP for a service, IdP returns local user attributes and Referrals to Linking Service. Linking Service then directly or indirectly aggregates the attributes from multiple IdPs
- User only needs to login to a single IdP

# Feedback from Community

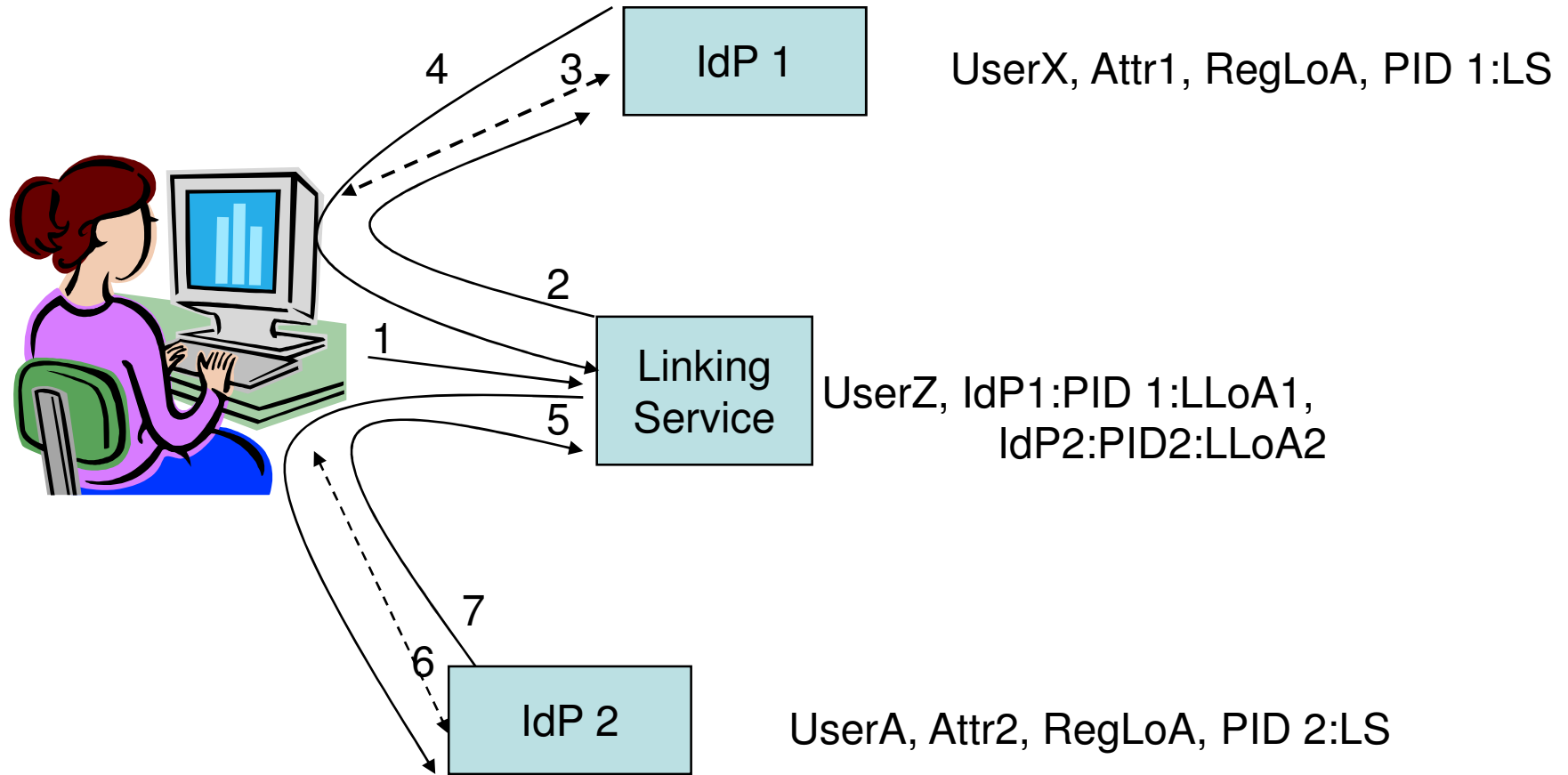
- We should support Linking Service aggregation as well as SP aggregation
- We should add support for Level of Authentication (LoA)

# Level of Assurance (LoA)



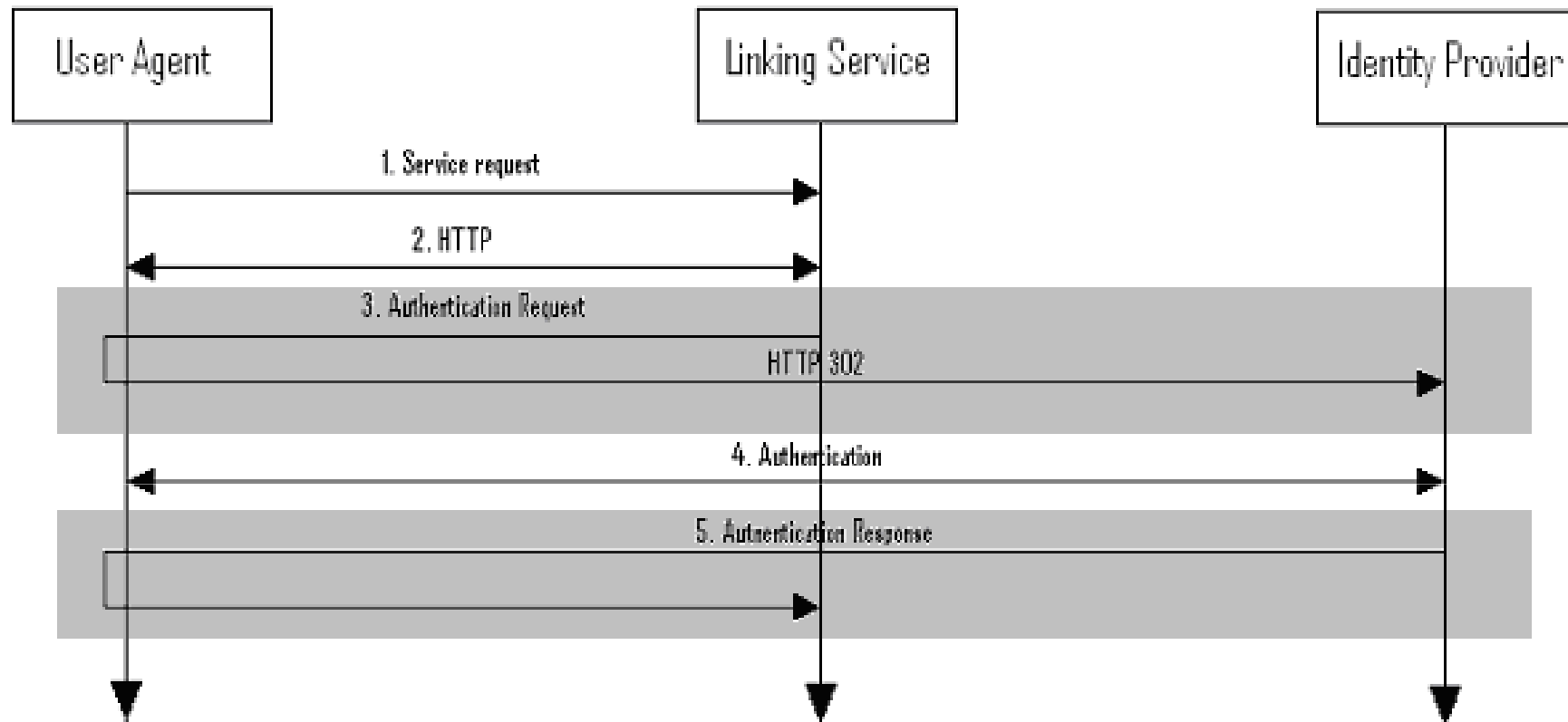
# Linking Service

## *Storage Requirements*





# Linking Protocol



- 2. User selects IdP from displayed page
- 3. `<samlp:AuthnRequest>`
- 4. IdP proprietary authentication
- 5. `<samlp:Response>` with persistent ID and authn context (to derive linked LoA)

## Linking Table

UserID	PId	IdP	LinkLoA
Fred	A=123	Airmiles.com	1
Fred	EduX=u23@kent.ac.uk	Kent.ac.uk	2
Mary	ABC=456	XYX Co	1
Fred	uid=123345	Cardbank.com	3

UserID	SP	IDP
Fred	Books.co.uk	Kent.ac.uk
Fred	Books.co.uk	Cardbank.com
Mary	Books.co.uk	XYX Co
Fred	Cardbank.com	*
Fred	Compstore.com	Cardbank.com
Fred	Compstore.com	Airmiles.com
Fred	*	Kent.ac.uk

**Link Release Policy Table**

# Use of LoA

- User first registers at an IdP with a Registration LoA
- Thereafter each act of Authn can only be at the same or lower Session LoA
- Linking Service optionally records the Session LoA at linking time as the Linking LoA
- During service requests, Linking Service will only contact linked IdPs with Linking LoAs .GE. than the current Session LoA
- This prevents a user claiming an attribute with a higher Session LoA than Registration LoA

# User Association Models

- User contacts SP and from there chooses his preferred IdP for authn (IdP Direct model)
- User contacts SP and from there chooses his preferred LS for authn, and from there chooses his preferred IdP (IdPviaLS model)
- User contacts SP, is directed back to his own workstation and from there he chooses his preferred set of IdPs (CardSpace model)
  - Still to complete

# Attribute Aggregation at Service Time

- SP Aggregation – The SP collects the attribute assertions from the IdPs
- LS Aggregation – The LS collects the attribute assertions from the IdPs
- User Aggregation – The user's workstation collects the attribute assertions from the IdPs
- There can be no IdP Aggregation because no IdP knows at which other IdPs the user is associated

# Service Provider Protocol Mappings

- Standard SAML2 Authentication Request from SP to IdP (or LS) in which AttributeConsumingServiceIndex attribute is used to specify that both user attributes and Referrals (endpoint reference attributes) should be returned in the response.
- Response contains 3 assertions:
  - a (new) SSO authentication statement, which may contain a LOA attribute as the principal's authentication context signed by IdP
  - the user's attributes that match the attributes requested in the <samlp:AuthnRequest> signed by IdP
  - Referrals encoded as ID-WSF Endpoint References (EPR)

# Conceptual Contents of a Referral

- A user ID that is the PId of the user, originally generated by the recipient IdP, and encrypted to the public key of the recipient IdP.
- The name of the recipient IdP (or LS) that is the destination of the Referral.
- A link to the authentication assertion that was created for this user session.
- The name of the SP that requires the user's attributes
- The name of the initiator of the Referral (i.e. the authenticating IdP or LS)
- The whole construct is digitally signed by the creator of the Referral (i.e. the authenticating IdP or LS)

# Encoding a Referral as an ID-WSF endpoint reference

- Use Liberty ID-WSF endpoint reference attribute in which the <sec:Token> contains a SAML assertion
- The ServiceType set to SAML V2.0 Authn provider
- the SecurityContext element contains:
  - A <SecurityMechID> element of value “urn:liberty:security:2005-02:TLS:SAML” indicating the <sec:Token> element below is a SAML V2.0 assertion.
  - A single <sec:Token> element, containing a referral assertion
- The Referral assertion (see next Slide)



# Referral Assertion

- Issuer attribute set to the issuer of the Referral
- ds:Signature element contains the issuer's signature
- Assertion's Subject is an <EncryptedID> element.  
The decrypted value of this contains the persistent ID valid between the issuer and the target of the referral.
- Conditions element contains an Audience restriction of the IdP/LS recipient
- Advice element contains an AssertionIDRef element that points to the authentication assertion used in the initial act of authentication. This provides the link between the referral and the principal's act of authentication.

# SP Handling of Reply from IdP

- Discussed protocol mappings with Internet2 and Liberty experts.
- Three different protocol mappings have been proposed for handling the incoming Referrals and encoding them in outgoing messages
- Straight SAML protocol creates an outgoing Authn Request (LS knows to treat this as an ID mapping request)
- Thin and Fat Liberty Alliance ID-WSF Identity Mapping Requests create different variants of the mapping request
  - Thin sends a SAML Attribute Query and Identity Mapping Request
  - Fat embeds the attribute query inside the Identity Mapping Request

# Straight SAML

1. SP sends `samlp:AuthnRequest` via browser to IdP A
2. IdP A returns `samlp:Response` to SP containing SSO assertion with `NameID` in Subject shared by SP and IdP A and some attributes. One of the attributes is an ID-WSF EPR pointing to the linking service.
3. SP wants to get attributes from IdP B, which is linked to LS. SP sends a `samlp:AuthnRequest` to LS using the ID-WSF SSOS spec, asking for a token for the SP to use in querying IdP B. The token inside the EPR in step 2 is attached to the message and the message is signed by the SP.
4. LS checks policy and then returns a token with an `EncryptedID` containing the `NameID` shared by LS and IdP B. It is targeted at IdP B and is bound to the SP's key.

# Straight SAML (cont)

5. SP sends a `saml:AttributeQuery` to IdP B and attaches the token it got in step 4. It signs the message with its key. The query's Subject is the EncryptedID from the token in step 4.
6. IdP B checks policy and returns attributes permissible to return to the SP. The subject of that assertion is the EncryptedID that was in the query. The SP adds that token to its existing set of material for that session and does what it likes with the attributes.
7. Steps 3-6 can be repeated for any number of IdPs and can be optimized by requesting multiple tokens up front in step 3 from LS.

Advantages. Scott Cantor thinks only SP code needs to change

Disadvantages. SP has to know which other IdPs it wishes to contact . Not symmetric (cannot interchange LS and IdPs).  
User is identified by different encryptedIDs (no consistency).  
Does not support LoA

# Liberty Alliance ID-WSF Identity Mapping Request

- This message requests the recipient to map the identity of the user from the PId (which it knows) into the random id in the authentication assertion (which it does not know)
- An ID-WSF Identity Mapping Request comprises a <sec:Token> which identifies the entity for whom new identity tokens are required, and a <sec:TokenPolicy> which specifies the characteristics of the identity token that is to be returned

# Thin Liberty Alliance ID-WSF Identity Mapping Request

- One <MappingInput> element in which the ReqID is missing
- <sec:Token> contains the <sec:Token> of the EPR from which this request was created
- <sec:TokenPolicy> consists of the following:
  - Type equals *urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion*
  - A <sec:Token> is added which contains the initial authn assertion, used to specify the identifier to be used in the new tokens
  - A new Aggregate attribute (boolean) is added to the token policy.
    - Its presence is mandatory to signal that this profile is being used.
    - Its value is advisory and can be ignored by the recipient. True means the recipient should perform aggregation. False indicates the recipient should not perform aggregation, in which case Referrals (a set of EPR's) should be returned in the response.

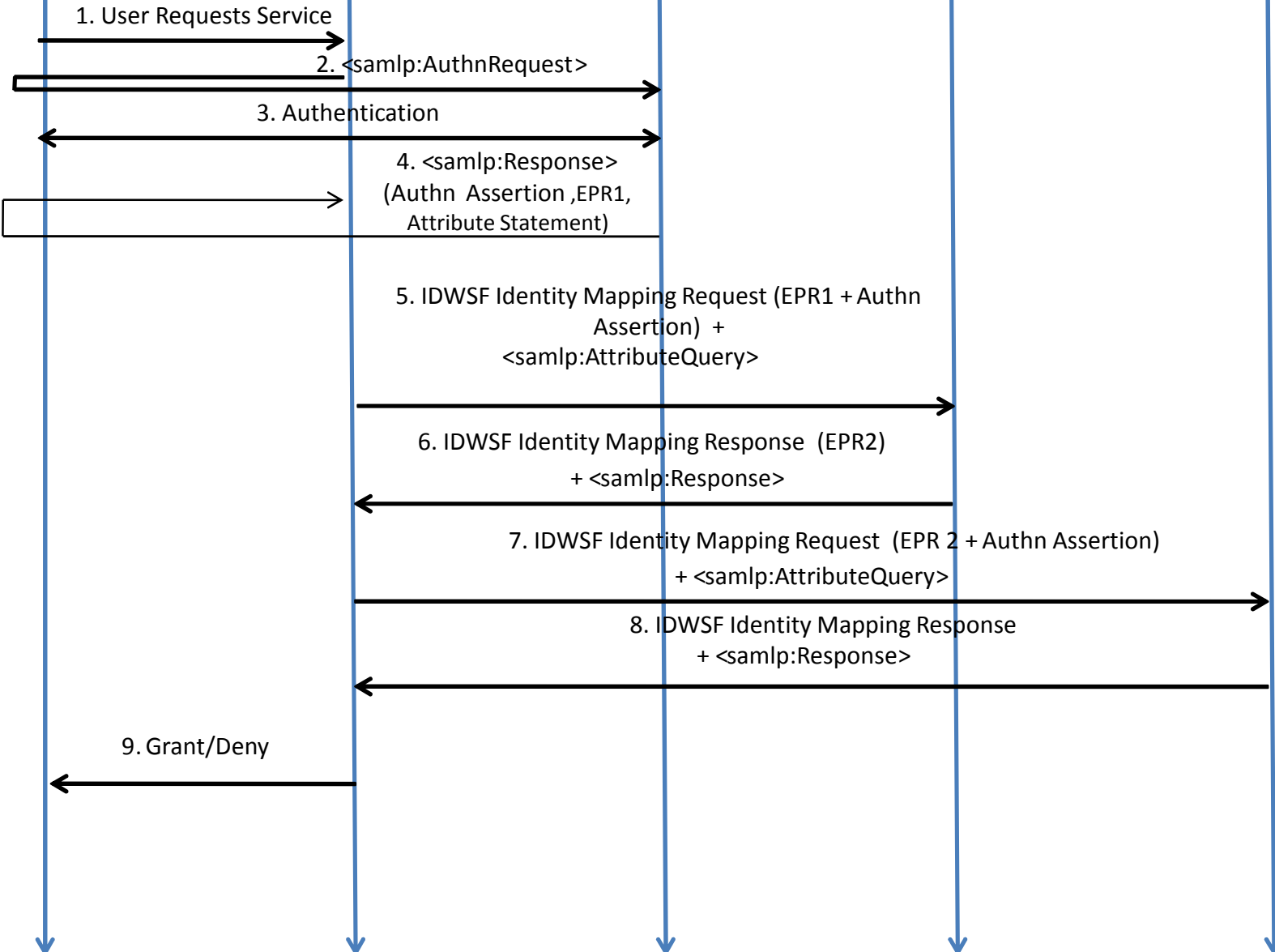
# SAML <samlp:AttributeQuery>

- <saml:Subject> contains the random name identifier used in the initial authentication query issued by the authenticating IdP.
- The <saml:Attribute> element MAY contain a request for each of the attributes required to authorise the principal but may be omitted indicating that all attributes should be returned.
- Due to the potential for distributed aggregation that our conceptual model proposes, there is a need for a new attribute (AssertionConsumerServiceURL ) which is equivalent to the attribute of the same name in the SAML V2.0 <AuthnRequest>. Its purpose is to identify the ultimate consumer of the aggregated attributes i.e. the SP, so as to allow the attributes to be encrypted using the SPs public key

# FAT Liberty Alliance ID-WSF Identity Mapping Request

- Combines the thin ID Mapping Request and Attribute Query
- One <MappingInput> element in which the ReqID is missing
- <sec:Token> element contains the <sec:Token> element of the ID-WSF EPR from which this request was created
- <sec:TokenPolicy> consists of the following:
  - Type set to *urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion*
  - A new <sec:Token> is added which contains the initial authn assertion, used to specify the identifier to be used in the new tokens
  - The new Aggregate attribute (boolean)
  - Zero or more <saml2:attribute> elements to specify the attributes whose value(s) are to be returned . If no attributes are specified, it indicates that all the attributes allowed by policy are requested.
  - The new AssertionConsumerServiceURL attribute to specify the entity to which any resulting attribute assertions should be encrypted.





IdP Direct SP aggregation thin IDWSF Id Mapping

# Pros and Cons of LA ID Mapping

- Advantages: SP is given assertions in which the nameID is always the same as that in the original Authn assertion. Supports LoA. Fully supports conceptual model, with aggregation by different entities, and transparent replacement of IdPs and LS
- Disadvantages. Requires more mods to IdP code but less mods to SP. However we should be able to provide plugins.

# Responses

- To SAML Authn Query
  - an Authn Assertion and Attribute Statement containing IDWSF endpoint reference attributes
- To Thin ID-WSF Attribute Mapping Request
  - An Identity Mapping Response containing additional EPR attributes as <sec:Token> elements
  - a <samlp:Response> containing any aggregated attribute assertions or none
- To Fat ID-WSF Attribute Mapping Request
  - An Identity Mapping Response containing an attribute assertion with EPR attributes and/or user attributes

# Implementation

- An open source Linking Service in Java (BSD licence) with a web interface for user linking and a web services interface for SP linking
- IdP Java toolkit
  - An IdP interceptor that takes an incoming WS-Identity Mapping Request and [Attribute Query], validates its security assertions, then forwards a standard Attribute Query to the IdP using the Permanent ID
  - An IdP plugin for outgoing messages that replaces PId with the random ID from authn assertion
- SP toolkit?? No Java SP at the moment
  - An SP plugin that takes an incoming EPR and turns it into an outgoing Identity Mapping Request

# Any Questions?

- Latest information can be obtained from Shintau website
- <http://sec.cs.kent.ac.uk/shintau>