FPKIPA

Federal Public Key Infastructure Policy Authority

Rich Attribute Exchange with PKI Certificates

Version 1.0.0
June 6, 2007

Document History

Status	Release	Date	Comment	Audience
Final	1.0.0	6/6/07	Public release incorporating feedback from Peter Alterman	Public

Editors

Chris Brown	Treb Farrales	Matt King
Chris Louden	Terry McBride	Tim Pinegar
Dave Silver	Peter Alterman	

Executive Summary

Public key certificates provide a very strong authentication mechanism. However, they contain very little information about the end user. This limits the relying party, particularly regarding authorization decisions.

Since higher assurance authentication is becoming increasingly necessary, there is increasing adoption of public key certificates. Therefore, relying parties need “rich attribute exchange” in context of public key certificate based authentication, to have the information necessary for such things as authorization. This paper highlights viable options for rich attribute exchange, focusing on getting the attributes to the relying party. This includes four (4) conceptually different approaches (and alternative implementations of each):

- Prompt the End User;
- Web Based Identity Standard;
- Provisioning; and
- Attributes in Certificates

For each approach, this paper discusses essential considerations such as attribute extensibility, confidentiality, information assurance, and complexity. A listing of pros and cons highlights critical points. In addition, transaction flow diagrams highlight the workings of each approach.

The paper concludes by summarizing and comparing the approaches in a set of tables. Comparisons are at both a high level and a detailed level. Color-coding indicates how each approach fairs per essential consideration.

The appendices provide brief overviews of technologies potentially useful to rich attribute exchange. The glossary explains essential terms and concepts used throughout the document.

Table of Contents

1 Introduction....6
2 Approaches to Rich Attribute Exchange with PKI....7
2.1 Prompt the End User....9
2.1.1 No Attribute Verification....9
2.1.2 Attribute Verification....11
2.1.2.1 Out of band....11
2.1.2.2 Attribute Authority....12
2.2 Web Based Identity Standard....14
2.2.1 Attribute Authority Discovery....14
2.2.1.1 Thick Client....14
2.2.1.2 Attribute Authority Certificate Extension....16
2.2.2 Pre-Arranged Attribute Authority....18
2.2.2.1 Exchange Metadata....18
2.2.2.2 Assertion-based Authentication....21
2.3 Provisioning....23
2.3.1 Service Provisioning Markup Language (SPML)....23
2.3.2 Shared Directory....26
2.4 Attributes in Certificates....27
2.4.1 Public Key Certificate Extensions....28
2.4.2 Attribute Certificates....30
3 Summary and Conclusions....32
Appendix A: Overview of Relevant Technologies....38
A1. CardSpace....39
A2. Web Services Trust Language (WS-Trust)....42
A3. Liberty Identity Web Service Framework (ID-WSF)....45
A4. Security Assertion Markup Language (SAML)....47
A5. Service Provisioning Markup Language (SPML)....48
A6. Public Key Infrastructure (PKI)....52
A6. Public Key Infrastructure (PKI)....53
A7. Domain Name Service (DNS)....54
Appendix B: Document References....56
Appendix C: Glossary....58
Appendix D: Acronyms....63

Figures

Figure 2-0 Summary of Approaches....7
Figure 2-1 Prompt End User with No Attribute Verification....10
Figure 2-2 Prompt End User with Out of Band Verification....11
Figure 2-3 Prompt End User with Attribute Authority Verification....12
Figure 2-4 Thick Client....15
Figure 2-5 Attribute Authority Certificate Extension....16
Figure 2-6 Exchange Metadata....19
Figure 2-7 Assertion-based Authentication....21
Figure 2-8 SPML....24
Figure 2-9 Shared Directory....26
Figure 2-10 Public Key Certificate Extensions....28
Figure 2-11 Attribute Certificates....30

Figure A1-1 CardSpace Interactions....40
Figure A2-1 WS-Trust Applicability in the Enterprise Solution....44
Figure A3-1 ID-WSF Sequence Diagram....45
Figure A5-1 How SPML Works....50
Figure A5-2 SPML Applicability in the Enterprise Solution....51
Figure A5-3 SPML Features Roadmap....52
Figure A7-1 DNS Tree....55

Tables

Table 3-1Level 1 Comparison 33
Table 3-2 Level 2 Comparison 33
Table 3-3 Level 3 Comparison, Attribute Extensibility....................34
Table 3-4 Level 3 Comparison, Confidentiality 35
Table 3-5 Level 3 Comparison, Information Assurance....................36
Table 3-6 Level 3 Comparison, Complexity 37

1 INTRODUCTION

Public key Infrastructure (PKI) based authentication of end users has significant security strengths deriving from the ability to keep the private key secret. In addition, the use of mutually authenticated Secure Sockets Layer (SSL) / Transport Later Security (TLS), ubiquitously available in web browsers and web servers, creates substantial session management strengths that are often overlooked. A mutually authenticated SSL/TLS channel is substantially more resistant to session hijacking than cookie-based session identifiers, even when strong cryptography is used to protect cookies.

However, the only information readily available to relying parties in mutually authenticated TLS connections is the information embedded in the X.509v3 certificates (public key certificates). Privacy considerations generally keep the information in a public key certificate to a minimal set of information, typically “simple identity” information such as name. Volatile information such as roles and privileges are generally absent from public key certificates, partially due to privacy considerations and partially due to a desire to minimize the frequency of public key certificate reissuance. In most cases, PKI based authentication of end users results in very high confidence in very little information. This can create a challenge for a relying party (RP), where the “square peg” of simple authentication is forced into the “round hole” of authorization (e.g., role based access control). This challenge is summarized as follows:

- Trusted public key certificate policies provide very strong authentication mechanism;
- Cryptography provides a very strong binding of the authentication act and end user session;
- There are mechanisms for public key certificate path discovery and validation (PDVal), which can allow previously unknown end users can be authenticated at the RP with high assurance;
- There is very little information in the public key certificate. Therefore, the RP needs more information about the end user for many different purposes, but primarily for authorization decisions;
- RPs are adopting public key certificates because higher assurance is becoming increasingly necessary. Therefore, RPs need “rich attribute exchange” in context of public key certificate based authentication, to have the information necessary to do proper authorization

This paper highlights viable options for rich attribute exchange, focusing on getting the rich attributes to the RP. For each option discussed, authentication has already occurred using a public key certificate and associated private key. Rich attribute exchange is in addition to the authentication act.

2 APPROACHES TO RICH ATTRIBUTE EXCHANGE WITH PKI

Figure 2-1 highlights alternative approaches to rich attribute exchange. The levels in the left margin indicate a progression from the high-level objective to specific alternative implementations:

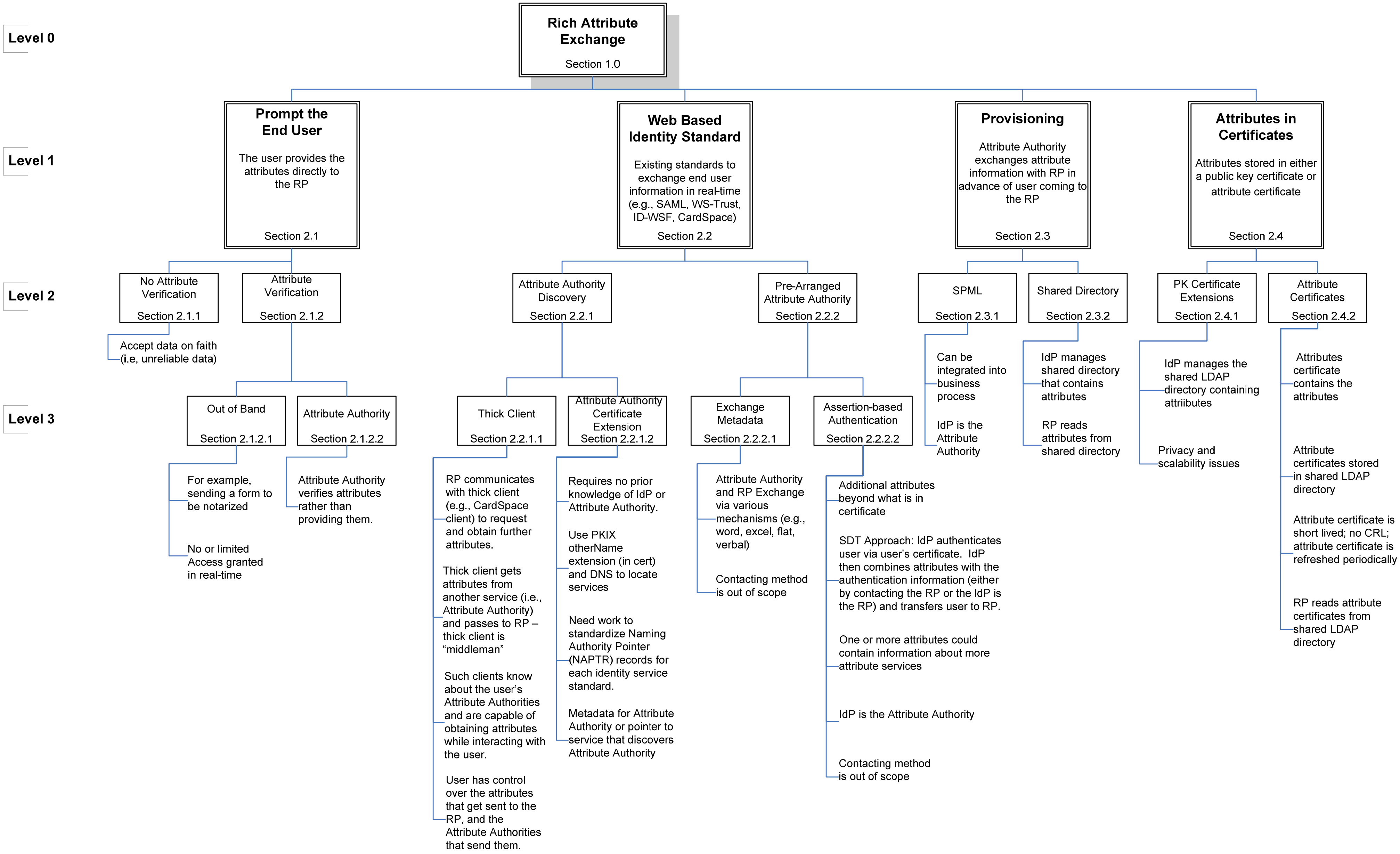
- Level 0 – the primary objective, which is rich attribute exchange
- Level 1 – four very different conceptual approaches to achieving the objective
- Level 2 – alternative implementations of each conceptual approach
- Level 3 – alternatives within a specific implementation

The diagram provides an initial footing for the reader. A more detailed assessment of the alternatives follows the diagram.

To enhance understanding of alternative approaches, a diagram for each is included that steps through the high-level transaction flow. Black arrows indicate real-time processing (i.e., at the time the end user seeks access to RP applications and/or resources). Pink arrows indicate off-line, non real-time processing that occurs before or after the authentication act.

The authoritative source of attributes is the Attribute Authority (AA) – person or system – most directly relevant to the RP's use of the rich attributes.

Figure 2-0 Summary of Approaches


Level 0
Level 1
Level 2
Level 3
Rich Attribute Exchange
Section 1.0
Prompt the End User
The user provides the attributes directly to the RP
Section 2.1
Web Based Identity Standard
Existing standards to exchange end user information in real-time (e.g., SAML, WS-Trust, ID-WSF, CardSpace)
Section 2.2
Provisioning
Attribute Authority exchanges attribute information with RP in advance of user coming to the RP
Section 2.3
Attributes in Certificates
Attributes stored in either a public key certificate or attribute certificate
Section 2.4
No Attribute Verification
Section 2.1.1
Accept data on faith (i.e, unreliable data)
Attribute Verification
Section 2.1.2
Out of Band
Section 2.1.2.1
For example, sending a form to be notarized
No or limited Access granted in real-time
Attribute Authority
Section 2.1.2.2
Attribute Authority verifies attributes rather than providing them.
Attribute Authority Discovery
Section 2.2.1
Thick Client
Section 2.2.1.1
RP communicates with thick client (e.g., CardSpace client) to request and obtain further attributes.
Thick client gets attributes from another service (i.e., Attribute Authority) and passes to RP – thick client is “middleman”
Such clients know about the user’s Attribute Authorities and are capable of obtaining attributes while interacting with the user.
User has control over the attributes that get sent to the RP, and the Attribute Authorities that send them.
Attribute Authority Certificate Extension
Section 2.2.1.2
Requires no prior knowledge of IdP or Attribute Authority.
Use PKIX otherName extension (in cert) and DNS to locate services
Need work to standardize Naming Authority Pointer (NAPTR) records for each identity service standard.
Metadata for Attribute Authority or pointer to service that discovers Attribute Authority
Pre-Arranged Attribute Authority
Section 2.2.2
Exchange Metadata
Section 2.2.2.1
Attribute Authority and RP Exchange via various mechanisms (e.g., word, excel, flat, verbal)
Contacting method is out of scope
Assertion-based Authentication
Section 2.2.2.2
Additional attributes beyond what is in certificate
SDT Approach: IdP authenticates user via user’s certificate. IdP then combines attributes with the authentication information (either by contacting the RP or the IdP is the RP) and transfers user to RP.
One or more attributes could contain information about more attribute services
IdP is the Attribute Authority
Contacting method is out of scope
SPML
Section 2.3.1
Can be integrated into business process
IdP is the Attribute Authority
Shared Directory
Section 2.3.2
IdP manages shared directory that contains attributes
RP reads attributes from shared directory
PK Certificate Extensions
Section 2.4.1
IdP manages the shared LDAP directory containing attriibutes
Privacy and scalability issues
Attribute Certificates
Section 2.4.2
Attributes certificate contains the attributes
Attribute certificates stored in shared LDAP directory
Attribute certificate is short lived; no CRL; attribute certificate is refreshed periodically
RP reads attribute certificates from shared LDAP directory

The following sections discuss in more detail the approaches highlighted in figure 2-1.

2.1 Prompt the End User

This approach requires the RP to interact directly with the end user to obtain attributes. The RP asks the end user for each attribute, whereupon the end user responds with the information. The interaction is in real-time (i.e., when the end user accesses the RP, immediately after the being successfully authenticated) through the end user client (e.g., the end user's web browser). The RP may or may not verify the information provided, with different consequences. In all cases, the end user fully controls the availability of information. The RP receives no information unless the end user voluntarily enters the information and selects a "Submit" button. Thus, the user is in complete control of the attribute exchange.

Since the AA for this approach is the end user, no prior exchange of metadata is required between the RP and AA. Similarly, there is no dynamic discovery of AAs, or support for multiple AAs when prompting the end user for attributes. However, direct interaction with the end user allows the RP to prompt for additional attributes on demand.

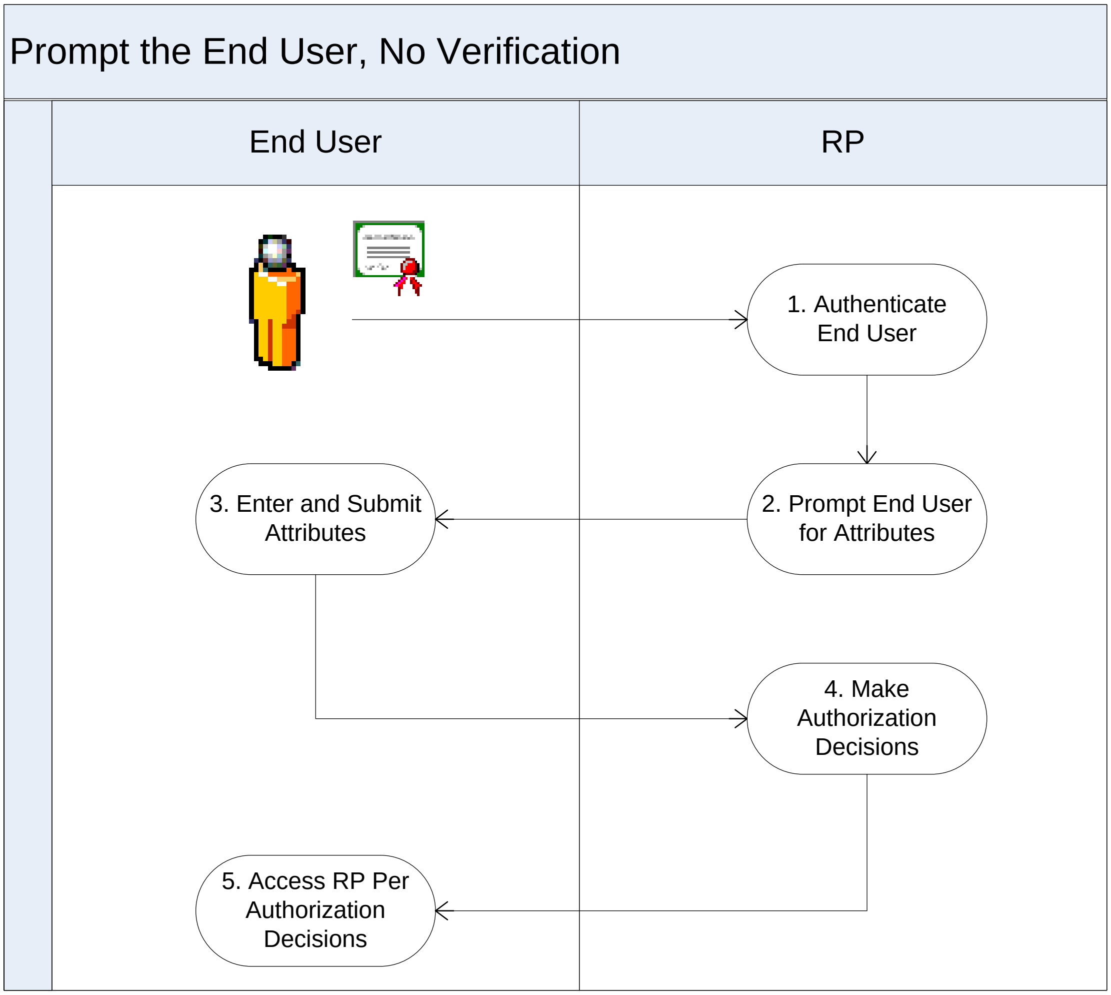
For all prompting approaches:

- Attribute data is typically communicated via HTML Forms
- The RP must rely on TLS for a secure channel (i.e., confidentiality via encryption)
- There is no cryptographic-based assurance (i.e., digital signature) that the data has not been altered during transit
- No changes to Certificate Authority profiles or policies are required
- The approach scales with the number of end users
- There is no standard for exchanging identity information (i.e., attributes) through HTML Forms

2.1.1 No Attribute Verification

The RP accepts end user information without any verification. In doing so, the RP accepts the risk of using incorrect information (maliciously or accidentally provided) to make authorization decisions. Therefore, the RP should assess potential implications and its risk tolerance for those implications. Figure 2-1 describes the high-level transaction flow.

Figure 2-1 Prompt End User with No Attribute Verification


Prompt the End User, No Verification
End User
RP
1. Authenticate End User
2. Prompt End User for Attributes
3. Enter and Submit Attributes
4. Make Authorization Decisions
5. Access RP Per Authorization Decisions

<u>Pros:</u>

- Relatively easy to understand and implement
- Supports real-time access to RP resources
- No parties other than the end user (as the AA) and the RP are involved in the attribute exchange (i.e., attributes are not shared with another party)

<u>Cons:</u>

- Little or no assurance that attribute data is correct
- No attribute data integrity
- No standards-based commercial off the shelf (COTS) software

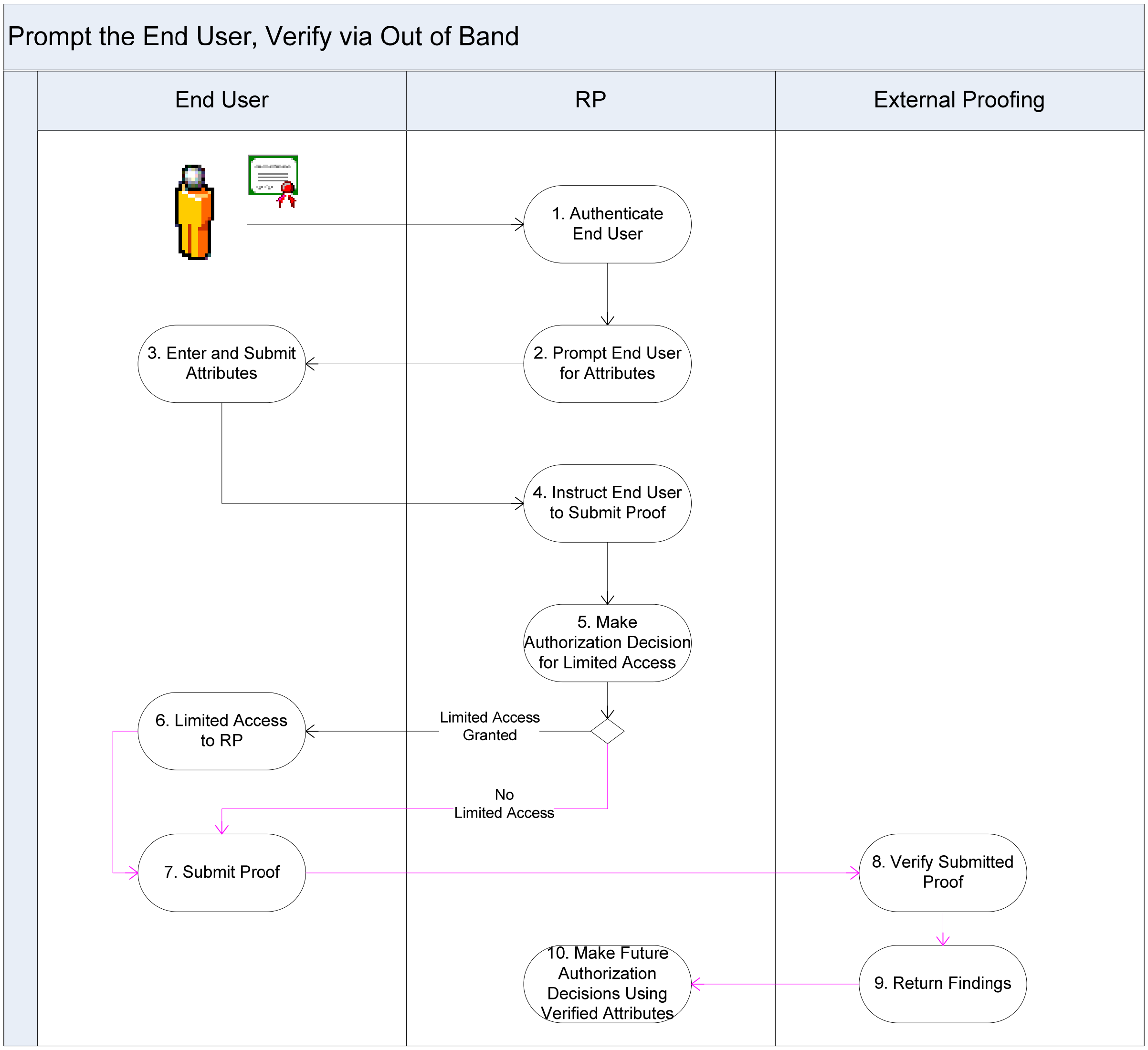
2.1.2 Attribute Verification

The RP verifies the attribute information to the extent possible before using it to make authorization decision. In many cases, attribute data integrity is not necessarily high (e.g., if a Knowledge-based Authority has the information, perhaps others have the information). Attribute verification approaches include, but are not limited to out of band verification and AA verification.

2.1.2.1 Out of band

Verification of attribute information occurs off-line, after the real-time prompting session between the end user and the RP concludes. The RP delays end user access until verification is completed. However, the RP may allow immediate, but limited access. The end user sends proof of attribute correctness (e.g., notarized document) to an RP-specified external entity. The external proofing entity informs the RP of attribute information validity based on its review of submitted proof. Figure 2-2 describes the high-level transaction flow.

Figure 2-2 Prompt End User with Out of Band Verification


Prompt the End User, Verify via Out of Band
End User
RP
External Proofing
1. Authenticate End User
2. Prompt End User for Attributes
3. Enter and Submit Attributes
4. Instruct End User to Submit Proof
5. Make Authorization Decision for Limited Access
Limited Access Granted
6. Limited Access to RP
No Limited Access
7. Submit Proof
8. Verify Submitted Proof
9. Return Findings
10. Make Future Authorization Decisions Using Verified Attributes

Pros:

- Increased attribute information assurance

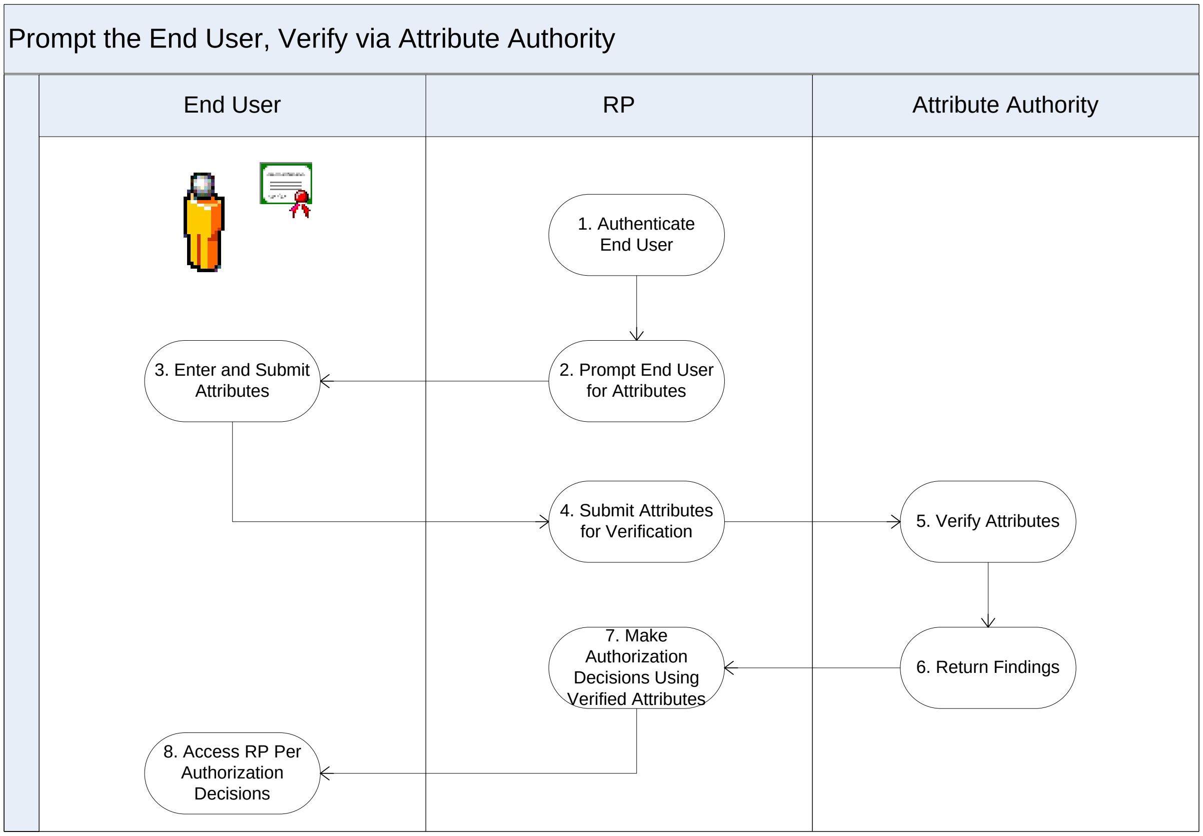
Cons:

- Inconvenience to the end user (e.g., delayed or limited access, submitting proof)
- Out of band submission could be lost, stolen, or tampered with
- Must establish a relationship and processes with an external proofing entity (e.g., enrollment officer or notary)
- No encryption or digital signature of out of band material
- No standards-based COTS software

2.1.2.2 Attribute Authority

The RP uses an AA for real-time verification of attribute information. In this scenario, the AA verifies attributes, rather than providing them. Figure 2-3 describes the high-level transaction flow.

Figure 2-3 Prompt End User with Attribute Authority Verification


Prompt the End User, Verify via Attribute Authority
End User
RP
Attribute Authority
1. Authenticate End User
2. Prompt End User for Attributes
3. Enter and Submit Attributes
4. Submit Attributes for Verification
5. Verify Attributes
6. Return Findings
7. Make Authorization Decisions Using Verified Attributes
8. Access RP Per Authorization Decisions

Pros:

- Increased attribute information assurance
- No end user access delays

Cons:

- More than one AA may need to be contacted
- The RP may ask the AA for more attributes than needed for authorization decisions, thus breaching confidentiality
- The AA is an additional entity with which the RP must technically interoperate, and with whom they must establish and maintain a trust relationship
- The end user does not control the exchange of his or her personal information between the RP and AA

2.2 Web Based Identity Standard

This approach leverages existing standards (e.g., SAML, WS-Trust, Liberty ID-WSF, CardSpace) that have, or are rapidly gaining traction. To varying degrees, one can also leverage valuable insights and lesson learned from real-world use of these standards by others. All the standards are assertion-based (sometimes referred to as claims-based) occurring in real-time (i.e., when the end user accesses the RP).

2.2.1 Attribute Authority Discovery

The RP does not require prior knowledge of the AA because metadata needed for technical interoperation with the AA is provided on demand, at run time. This allows a more dynamic approach because AAs can be added or changed in real-time without having to reconfigure the RP. In addition, attribute authority discovery allows inter-federation trust because of its standards-based approach and the ability to discover AAs across federations. This approach requires a pre-existing trust infrastructure. PKI can be used to ensure that trust.

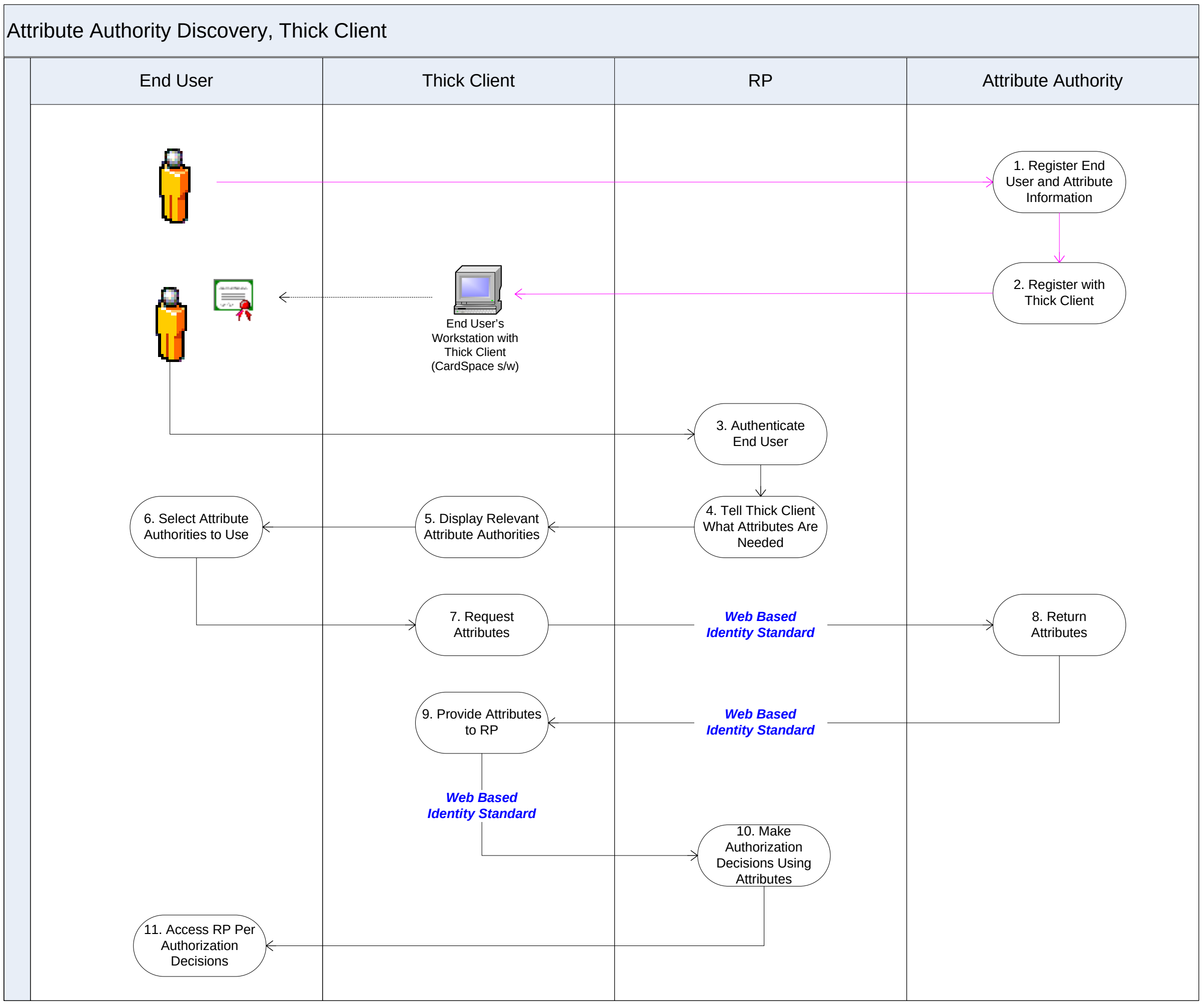
2.2.1.1 Thick Client

The RP communicates with software residing on the end user’s PC (thick client) to request and obtain attributes. In this approach, the end user’s PC comprises a web browser and thick client. Currently, only Microsoft CardSpace ([MS CardSpace]) is applicable for this approach.

The thick client retrieves attributes from one or more AAs, and passes the attributes to the RP. In this context, the thick client is a “middleman”. When the end user registers with an AA, the AA (and the attributes it maintains) is also registered with the end user’s thick client. When the end user accesses an RP, the RP tells the thick client what attributes it needs. The thick client displays the applicable AAs to the end user, and prompts the end user to select which AAs to use (particularly important if more than one AA maintains the same attributes). The thick client retrieves the attributes, and passes them to the RP.

This approach provides the end user with a high degree of control and approval, as the end user must select which AAs to use and may opt not to send any at all. Figure 2-4 describes the high-level transaction flow.

Figure 2-4 Thick Client


Attribute Authority Discovery, Thick Client
End User
Thick Client
RP
Attribute Authority
1. Register End User and Attribute Information
2. Register with Thick Client
End User's Workstation with Thick Client (CardSpace s/w)
3. Authenticate End User
4. Tell Thick Client What Attributes Are Needed
5. Display Relevant Attribute Authorities
6. Select Attribute Authorities to Use
7. Request Attributes
Web Based Identity Standard
8. Return Attributes
9. Provide Attributes to RP
Web Based Identity Standard
Web Based Identity Standard
10. Make Authorization Decisions Using Attributes
11. Access RP Per Authorization Decisions

Pros:

- Uses WS-Security for message-level integrity (signing) and confidentiality (encryption)
- TLS can be used for extra layer of security to encrypt transmission
- Very end user friendly because the end user is in control (i.e., end user has control over the attributes that get sent to the RP, and the AAs that send them)
- Likely to get correct attributes for identity context since end user picks the AAs
- Trend is towards thick client solutions (i.e., Microsoft Vista supports thick clients)

Cons:

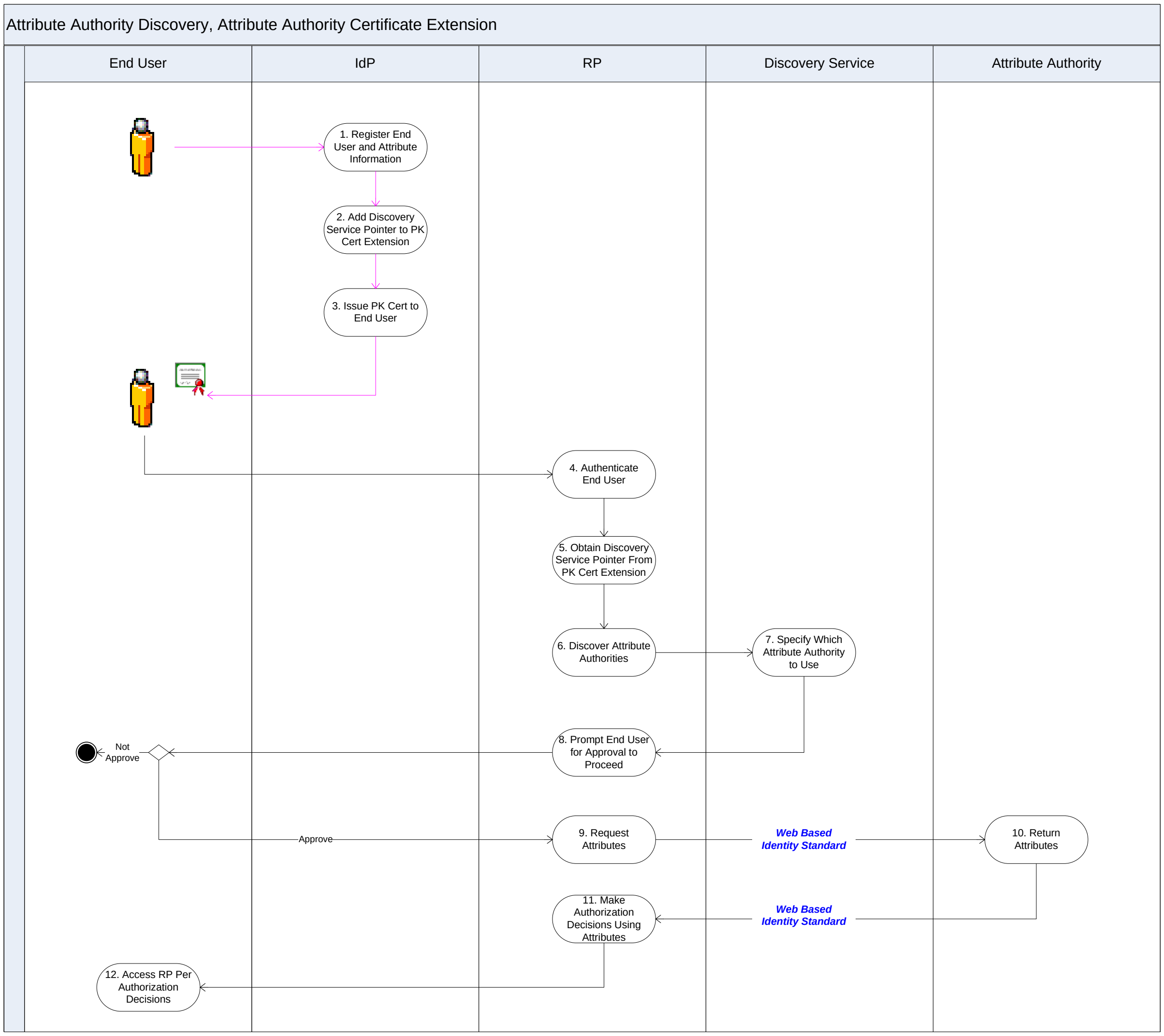
- Every PC the end user uses must have an installed, up-to-date (e.g., latest AA registrations) thick client
- An end user who roams off his or her computer may not have all of his or her AA information in other thick clients
- Requires a robust, highly organized, pre-existing trust infrastructure

2.2.1.2 Attribute Authority Certificate Extension

In this solution, a pointer to an AA discovery service is contained in the end user’s public key certificate. [PKIX] define one standards-based approach, which uses the otherName type in subjectAltName to indicate a domain name where services can be located. The RP can use the Domain Name Service (DNS) to locate the service on demand (i.e., when the RP needs the attribute information). In addition to hostnames and IP addresses, a lesser-known function that returns Naming Authority Pointer (NAPTR) records is used to return URLs and/or AA metadata. Use of a DNS pointer allows AAs to be easily added or changed, allowing maximum flexibility and extensibility. Once the RP discovers the AA, the RP communicates directly with it.

The RP obtains end user approval by prompting the end user for permission to proceed with discovery and attribute retrieval. Thus, the user relies on the RP and AA to follow policy. In addition, a path between the PKI certificates used by the AA and RP must exist to ensure mutual trust; use path discovery and validation. Figure 2-5 describes the high-level transaction flow.

Figure 2-5 Attribute Authority Certificate Extension


Attribute Authority Discovery, Attribute Authority Certificate Extension
End User
IdP
RP
Discovery Service
Attribute Authority
1. Register End User and Attribute Information
2. Add Discovery Service Pointer to PK Cert Extension
3. Issue PK Cert to End User
4. Authenticate End User
5. Obtain Discovery Service Pointer From PK Cert Extension
6. Discover Attribute Authorities
7. Specify Which Attribute Authority to Use
8. Prompt End User for Approval to Proceed
Not Approve
Approve
9. Request Attributes
Web Based Identity Standard
10. Return Attributes
11. Make Authorization Decisions Using Attributes
Web Based Identity Standard
12. Access RP Per Authorization Decisions

Pros:

- Less user involvement, and therefore more user friendly
- Uses WS-Security for message-level integrity (signing) and confidentiality (encryption)
- Uses TLS for encryption during transport

Cons:

- Weaker end user approval mechanism than in other approaches because it is technologically possible that the RP can ignore the end user and still proceed (i.e., no reliance on a step from the end user such as providing attribute information or specifying an AA)
- Work is needed to standardize NAPTR records for each identity service standard
- There are currently no COTS software that look up AA records in DNS

2.2.2 Pre-Arranged Attribute Authority

This approach requires the RP to have prior knowledge of the AA (i.e., before an end user accesses the RP). Knowledge of the AA is on a per end user basis and/or issuer basis (i.e., groups of end users). This is not a dynamic approach. On an ongoing basis, the AA must notify the RP of changes, and the RP must reconfigure accordingly.

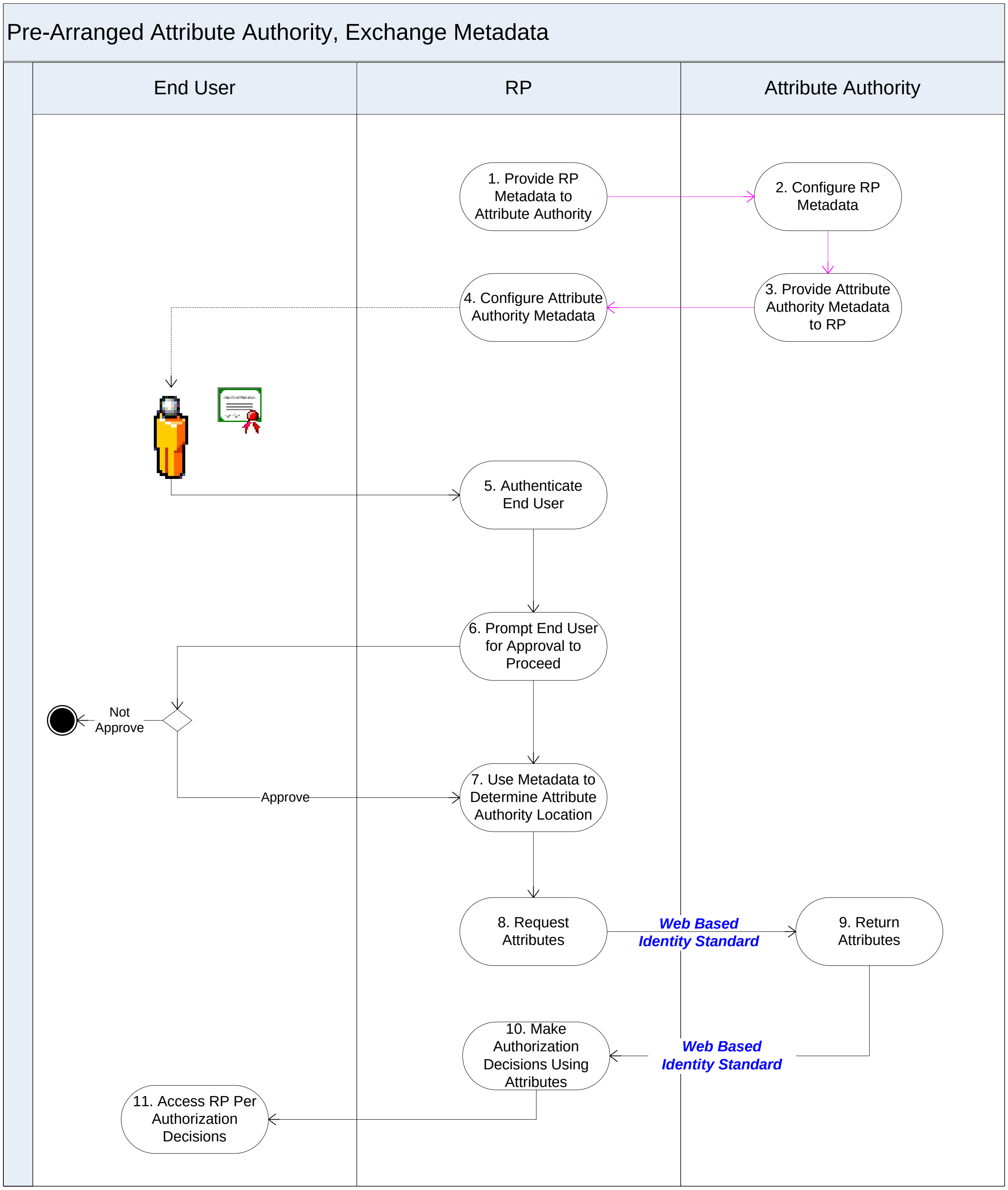
2.2.2.1 Exchange Metadata

The RP and AA exchange metadata that allows each to know about the other (e.g., where each service is located for communication purposes). The RP and AA establish mutual trust via successful metadata exchange. The exchange of metadata is by any means mutually agreed upon by the parties involved.[1] Accordingly, the AA location is already known to the RP when needed at run time (i.e., when an end user accesses the RP). The RP requests and obtains attributes on demand by communicating directly with the AA.

The RP obtains end user approval by prompting the end user for permission to proceed with attribute retrieval. Figure 2-6 describes the high-level transaction flow.

[1] Details of metadata exchange mechanisms are out of scope for this document. However, note that the mechanism selected (or not selecting a mechanism at all) could affect deployment of the approach (e.g., operational process, level of trust).

Figure 2-6 Exchange Metadata


Pre-Arranged Attribute Authority, Exchange Metadata
End User
RP
Attribute Authority
1. Provide RP Metadata to Attribute Authority
2. Configure RP Metadata
3. Provide Attribute Authority Metadata to RP
4. Configure Attribute Authority Metadata
5. Authenticate End User
6. Prompt End User for Approval to Proceed
Not Approve
Approve
7. Use Metadata to Determine Attribute Authority Location
8. Request Attributes
Web Based Identity Standard
9. Return Attributes
10. Make Authorization Decisions Using Attributes
Web Based Identity Standard
11. Access RP Per Authorization Decisions

Pros:

- Uses WS-Security for message-level signing and encryption
- TLS can be used for extra layer of security
- Metadata can (should) be signed to enhance trust
- Uncomplicated trust mechanism because the RP and AA simply exchange and configure metadata
- Generally simpler than real-time discovery approaches because there is no need for run time discovery infrastructure and processing

Cons:

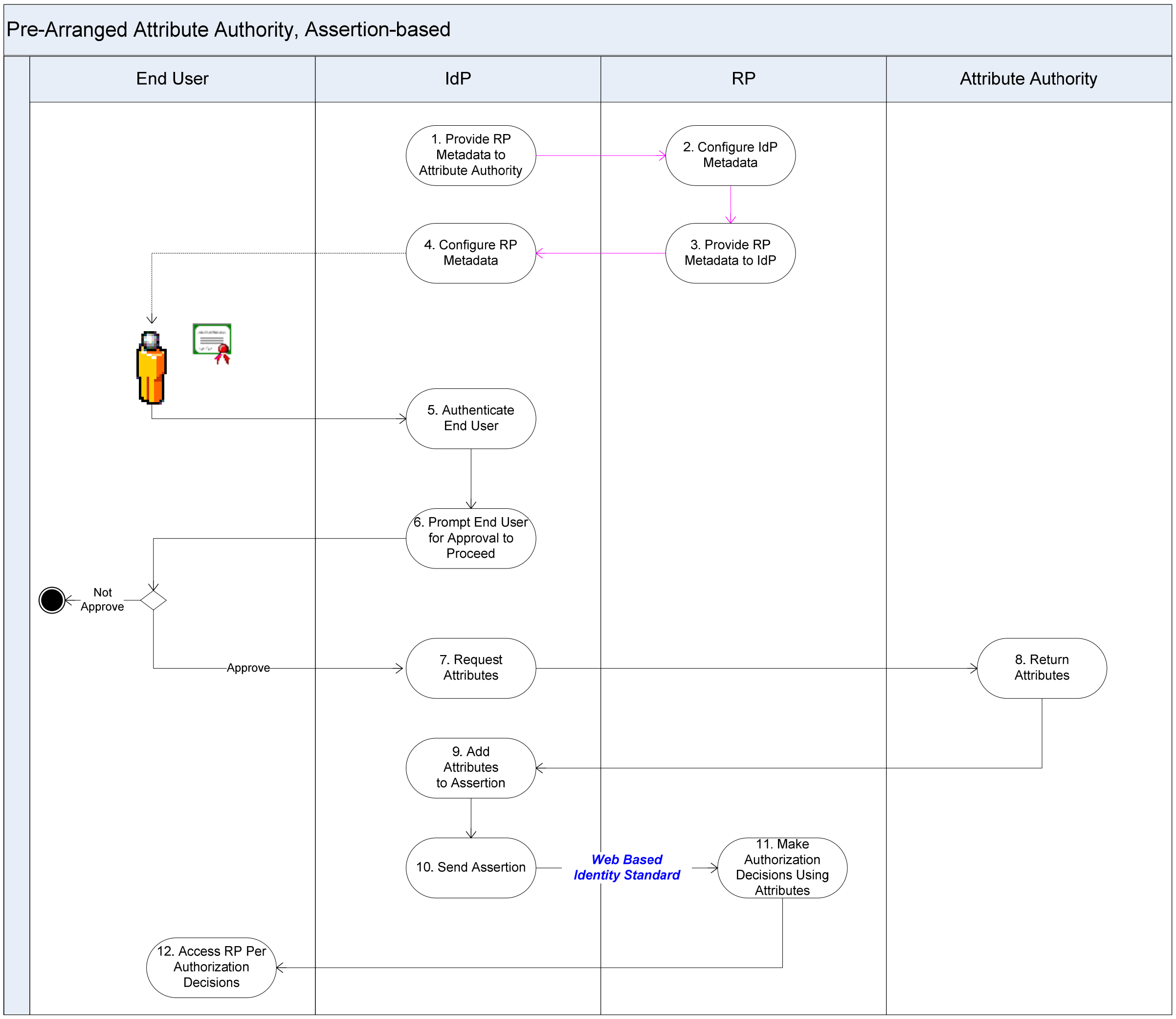
- Weaker end user approval mechanism than in other approaches because it is technologically possible that the RP can ignore the end user and still proceed (i.e., no reliance on a step from the end user such as providing attribute information or specifying an AA)
- The RP needs to pre-configure with the AA’s metadata
- The RP needs to integrate code to request attributes on demand

2.2.2.2 Assertion-based Authentication

In this approach, the end user uses his or her public key certificate to authenticate to the IdP – not the RP. Then after end user authentication, the IdP creates an assertion, adds additional attributes to the assertion as necessary, and sends the assertion to the RP. In this context, the assertion (a) asserts the end user's identity, and (b) contains the rich attributes needed by the RP. From the RP perspective, the IdP is always the AA. Interoperation between the IdP and the AA is out of scope for this document, though in general, it is typically accomplished by mechanisms discussed herein.

The RP and IdP exchange metadata that allows each to know about the other (e.g., where each service is located for communication purposes). The RP and the IdP establish mutual trust via successful metadata exchange. The exchange of metadata is by any means mutually agreed upon by the parties involved. The IdP obtains end user approval by prompting the end user for permission to proceed with attribute retrieval. Figure 2-7 describes the high-level transaction flow.

Figure 2-7 Assertion-based Authentication


Pre-Arranged Attribute Authority, Assertion-based
End User
IdP
RP
Attribute Authority
1. Provide RP Metadata to Attribute Authority
2. Configure IdP Metadata
3. Provide RP Metadata to IdP
4. Configure RP Metadata
5. Authenticate End User
6. Prompt End User for Approval to Proceed
Not Approve
Approve
7. Request Attributes
8. Return Attributes
9. Add Attributes to Assertion
10. Send Assertion
Web Based Identity Standard
11. Make Authorization Decisions Using Attributes
12. Access RP Per Authorization Decisions

Pros:

- Uses WS-Security for message-level signing and encryption
- TLS can be used for extra layer of security
- Metadata can (should) be signed to enhance trust
- Standards-based COTS products easily perform this function

Cons:

- Binding of the authentication to the user session at the RP is weaker because there is no cryptographic binding of the user public key certificate to the RP
- Weaker end user approval mechanism than in other approaches because it is technologically possible that the RP can ignore the end user and still proceed (i.e., no reliance on a step from the end user such as providing attribute information or specifying an AA)
- The RP needs to pre-configure with the AA's metadata
- If the IdP authenticates to the RP without an X.509 credential, then the higher level of trust is lost. So the IdP itself would have to have a certificate and authenticate to the RP prior to passing attribute information

2.3 Provisioning

This approach requires the IdP that registers the end user to populate an attribute repository with the end user's attributes – in advance of the end user accessing an RP. The attribute repository can be either a shared directory managed by the IdP, or an RP itself. In this regard, the IdP is a "broker," collecting attributes from the end user and providing the attributes elsewhere. Registration of an end user encompasses many scenarios including, but not limited to: (a) a new user, (b) employee promotion resulting in a new role, and/or (c) change in personal or professional information such as change of address or primary responder badge number.

Provisioning of an end user's attributes can occur to more than one RP at a time (i.e., to each RP with which the IdP has a relationship). For expediency, IdPs can do batch provisioning, which is multiple sets of end user attributes provisioned at the same time. Provisioning can occur periodically to provide up-to-date attributes to RPs. To ensure federation-wide uniqueness of the end user, (i.e., preclude misidentification of an end user by an RP), the IdP's unique identifier is included in provisioning. Provisioning can be integrated into existing IdP and/or RP business process to facilitate seamless processing.

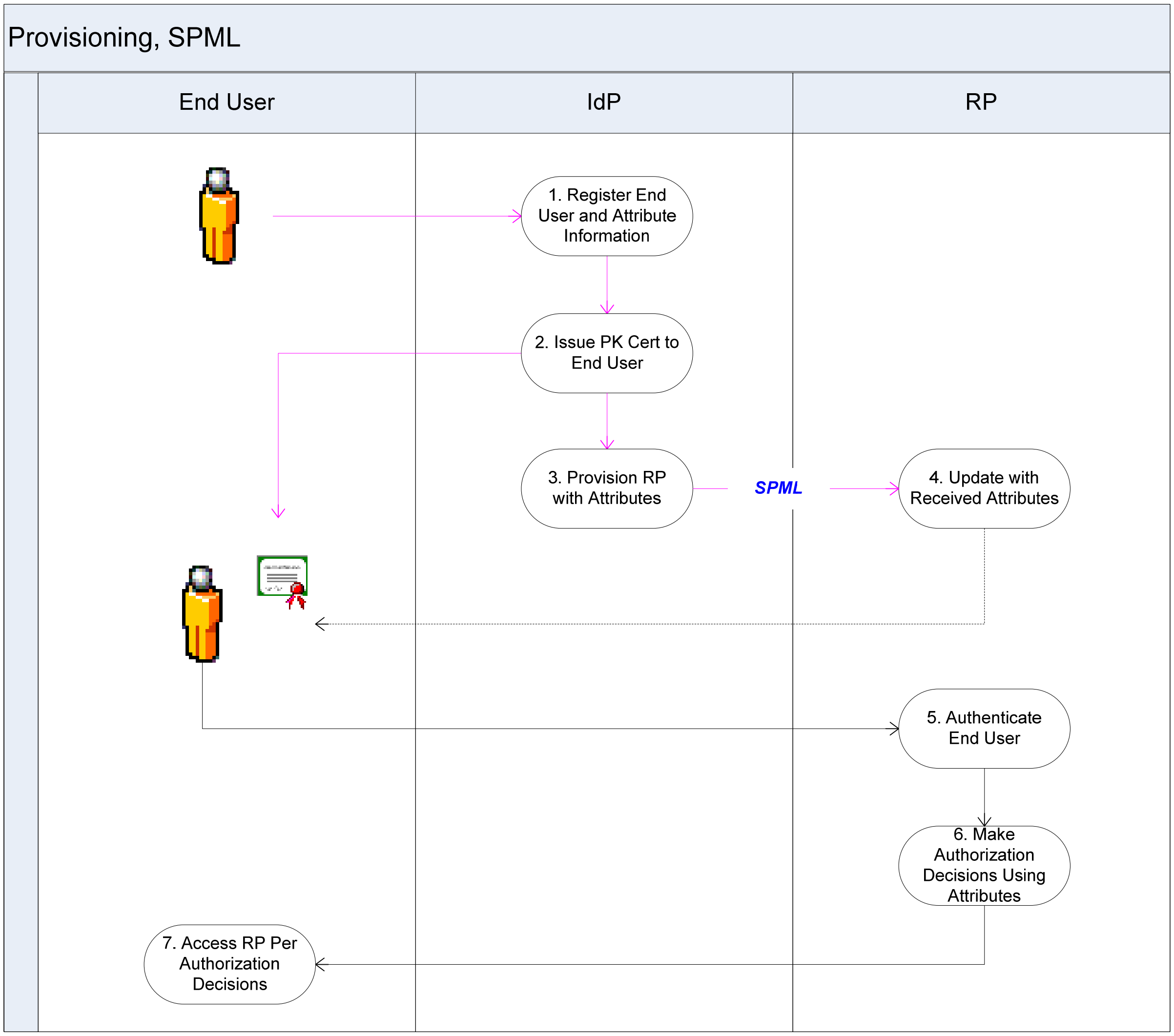
The end user approves provisioning of his/her attributes by signing the IdP end user agreement when registering. This requires IdP end user agreements to be revised to include appropriate provisioning language. This approval mechanism is weaker than some other approaches because there is no end user technological control that requires end user action to initiate provisioning (e.g., end user voluntarily enters attribute information and selects a "Submit" button).

In provisioning, the IdP is generally the AA. Provisioning does not define mechanisms for additional AAs, or discovery of AAs on demand.

2.3.1 Service Provisioning Markup Language (SPML)

The IdP provisions an RP directly. Therefore, the IdP is the AA. The IdP has pre-existing relationships with specific RPs. Ongoing coordination between the idP and partner RPs facilitate provisioning (initial and updates) as necessary. Trust must be established between the IdP and RP in advance of provisioning (via exchange of metadata), and verified at run time by TLS. Implementation is easier for the IdP and RP because of the standards-based specification for attribute exchange. It is likely that IdPs and RPs can integrate COTS products, rather than design and implement themselves. This approach is more appropriate for high user volumes in order to be cost and effort effective. That is, the IdP and RP would want a high degree of confidence that the end user will access the RP in order to provision the end user at the RP. Figure 2-8 describes the high-level transaction flow.

Figure 2-8 SPML


Provisioning, SPML
End User
IdP
RP
1. Register End User and Attribute Information
2. Issue PK Cert to End User
3. Provision RP with Attributes
SPML
4. Update with Received Attributes
5. Authenticate End User
6. Make Authorization Decisions Using Attributes
7. Access RP Per Authorization Decisions

Pros:

- SPML allows for message level signing and encryption
- TLS can be used for an extra layer of security
- More efficient at run-time because there are no calls to third parties for real-time transfer of attributes (i.e., attribute data is stored at the RP and immediately available)
- Attributes are exchanged directly between the AA and RP, and are not shared with other parties
- Standards-based COTS software available for provisioning with SPML

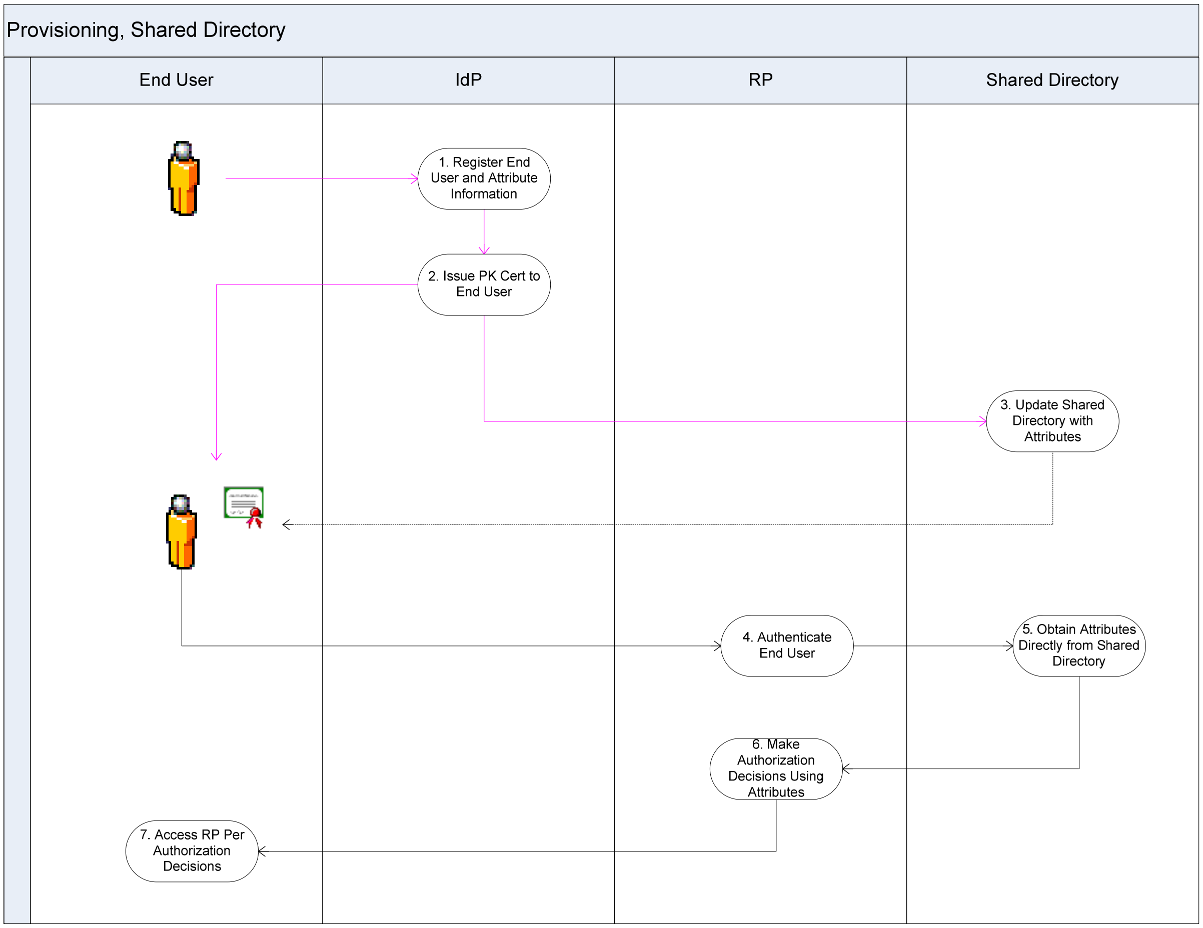
Cons:

- Provisioning updates are needed for new or changed data
- Does not define mechanisms for additional AAs, or discovery of AAs on demand
- No on demand access to authoritative source, which could have newer attributes
- Ongoing coordination between IdP and partner RPs is needed to ensure provisioning updates as necessary
- Weaker end user approval mechanism than some other approaches because there is no end user technological control that requires end user action to initiate provisioning (e.g., end user voluntarily enters attribute information and selects a “Submit” button).

2.3.2 *Shared Directory*

The IdP provisions attributes into a shared directory (e.g., LDAP Directory, X.500 Directory) that the IdP owns and manages. When an end user accesses an RP, the RP reads attributes from the shared directory in real-time. For security and access control purposes, the IdP can (should) configure the shared directory to require RP authentication. Trust must be established between the IdP and RP in advance (via exchange of metadata), and verified at run time by TLS. Figure 2-9 describes the high-level transaction flow.

Figure 2-9 Shared Directory


Provisioning, Shared Directory
End User
IdP
RP
Shared Directory
1. Register End User and Attribute Information
2. Issue PK Cert to End User
3. Update Shared Directory with Attributes
4. Authenticate End User
5. Obtain Attributes Directly from Shared Directory
6. Make Authorization Decisions Using Attributes
7. Access RP Per Authorization Decisions

Pros:

- TLS is used for transport layer security
- The IdP can configure the shared directory to force an RP to authenticate to the shared directory

Cons:

- The RP must have prior knowledge of the shared directory
- Directory synchronization becomes an issue; shared directory updates are needed for new or changed data
- No message layer security (i.e., no signing, no encryption)
- Potential privacy breach as an RP can access attributes for all end users, including those not serviced by it (i.e., no data-level access control)

- Security requirements for shared directory are substantial

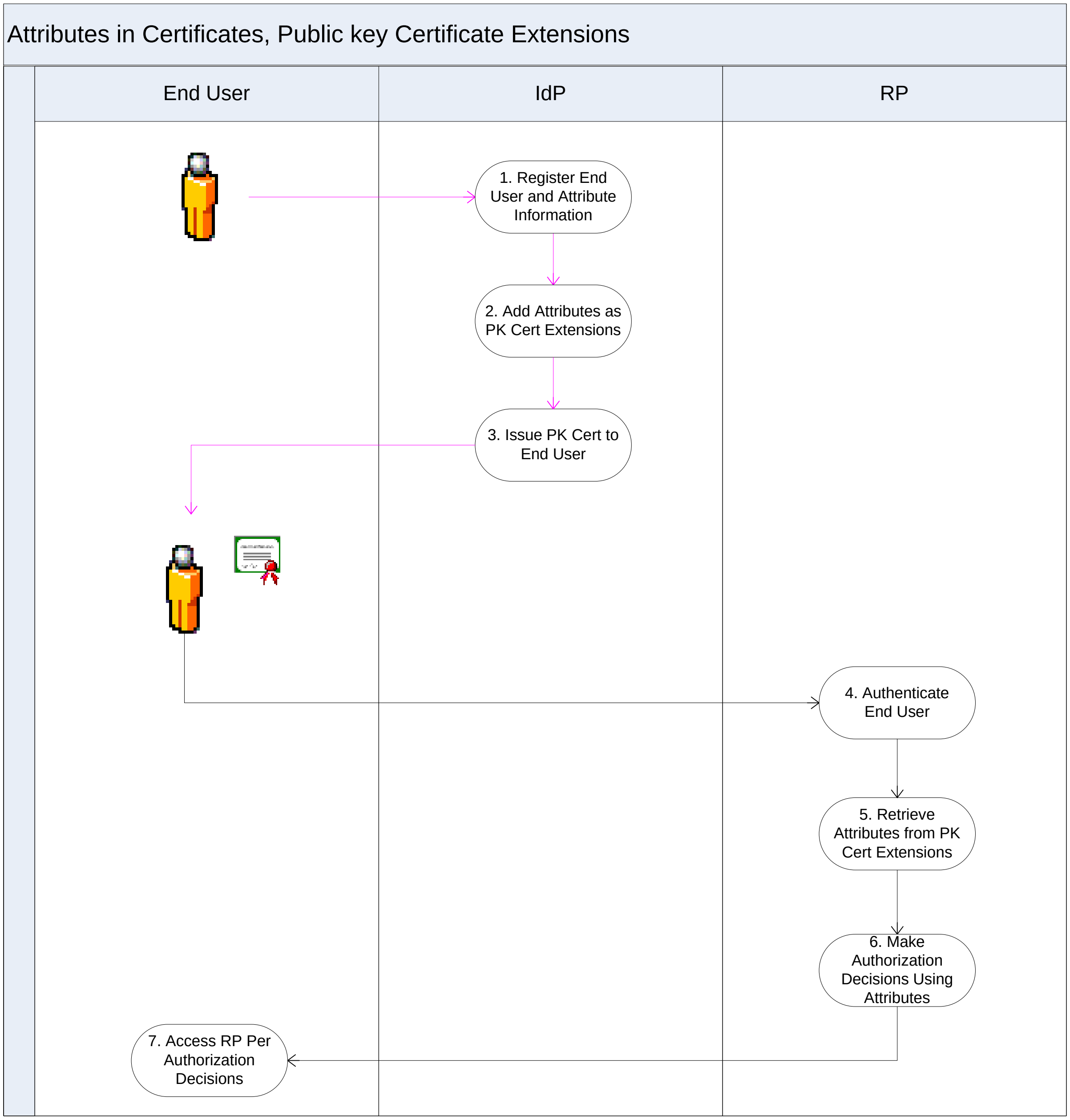
2.4 Attributes in Certificates

This approach leverages X.509 certificates to make end user attributes available to an RP – either an X.509 public key certificate or an X.509 attribute certificate. The Certification Authority (CA) populates the applicable certificate with attribute information obtained during end user registration – in advance of the end user accessing an RP. RP processing of either certificate is in real-time. The end user approves passing of his or her attributes by signing the IdP end user agreement when registering. This requires CAs to revise their end user agreements to include appropriate provisioning language. This approval mechanism is weaker than some other approaches because there is no end user technological control that requires end user action to initiate provisioning (e.g., end user voluntarily enters attribute information and selects a “Submit” button). In addition, there is no provision for adding or changing AAs, or exchanging specific attributes on demand.

2.4.1 *Public Key Certificate Extensions*

End user attributes are stored in the end user's X.509 public key certificate as extensions. The RP retrieves the end user's attributes from the extensions when presented with the public key certificate by the end user. Figure 2-10 describes the high-level transaction flow.

Figure 2-10 Public Key Certificate Extensions


Attributes in Certificates, Public key Certificate Extensions
End User
IdP
RP
1. Register End User and Attribute Information
2. Add Attributes as PK Cert Extensions
3. Issue PK Cert to End User
4. Authenticate End User
5. Retrieve Attributes from PK Cert Extensions
6. Make Authorization Decisions Using Attributes
7. Access RP Per Authorization Decisions

Pros:

- Less user involvement, and therefore more user friendly
- The RP has immediate access to attributes, therefore no need for real-time communication with third parties (i.e., discovery and transfer)
- Trust of attribute is very strong because of the inherent trust of verified public key certificates

Cons:

- No standards exist for attribute extension within the public key certificate
- Attributes are asserted by the CA, which is most likely far removed from the AA
- If an attribute changes, the public key certificate needs to be revoked and a new public key certificate issued; this is contrary to the intent that public key certificates be longer term (i.e., attribute lifetimes may shorten public key certificate lifetime)
- No end user control because the only alternative is not to use public key certificate
- Scalability issue (e.g., adding additional attributes requires additional extensions, which requires additional effort and/or processing); attributes are sent with every exchange of the public key certificate
- The CA must change their end user agreement to include permission to capture and include attributes in public key certificate extensions
- The CA must be enhanced to communicate with AAs and to add the public key certificate extensions
- The RP must be enhanced to parse for new public key certificate extensions, which is not necessarily easy
- Lack of privacy because it’s a public certificate by design

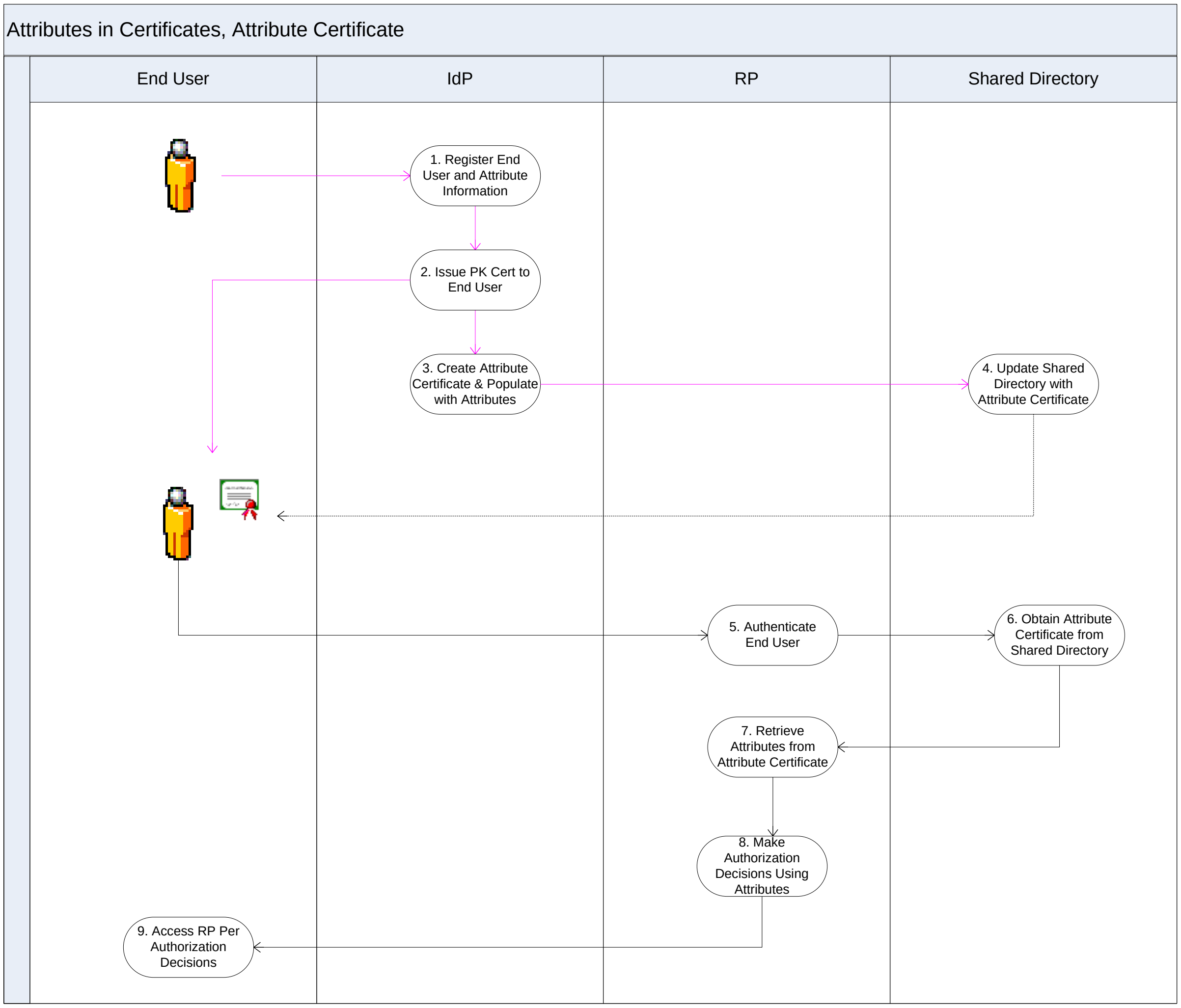
2.4.2 Attribute Certificates

End user attributes are stored in an X.509 attribute certificate. The attribute certificate is in addition to the end user's public key certificate. The IdP updates a shared directory that it owns and manages with attribute certificates. When an end user accesses an RP, the RP retrieves the appropriate attribute certificate from the shared directory. Figure 2-11 describes the high-level transaction flow.

Generally, attribute certificates have a short lifespan. There is no attribute certificate CRL. Attribute certificates should be refreshed periodically to ensure up-to-date attributes. Except for several attributes specified in [RFC 3281], attributes are not standardized.

Specific attributes can be encrypted within an attribute certificate. Encrypted attributes are carried using the EnvelopedData structure defined in [RFC 2630]. This requires the issuer of the attribute certificate to encrypt the attributes for each RP (i.e., a distinct symmetric key for each RP).

Figure 2-11 Attribute Certificates


Attributes in Certificates, Attribute Certificate
End User
IdP
RP
Shared Directory
1. Register End User and Attribute Information
2. Issue PK Cert to End User
3. Create Attribute Certificate & Populate with Attributes
4. Update Shared Directory with Attribute Certificate
5. Authenticate End User
6. Obtain Attribute Certificate from Shared Directory
7. Retrieve Attributes from Attribute Certificate
8. Make Authorization Decisions Using Attributes
9. Access RP Per Authorization Decisions

Pros:

- Less user involvement, and therefore more user friendly
- Privacy is improved over attributes within the public key certificate because attribute certificates allow for encrypted attribute data
- Enhanced trust because the attribute certificate is signed by AA (CA allows AA to sign on its behalf)
- The IdP can configure the shared directory containing attribute certificates to force an RP to authenticate to the shared directory

Cons:

- The CA must change their end user agreement to include permission to capture and include attributes in an attribute certificate
- The RP must have prior knowledge of the shared directory where the attribute certificates reside because there is no explicit pointer in the public key certificate
- [RFC 3281] allows an attribute "push" model (i.e., attribute certificate sent with public key certificate), but TLS prevents use of that model
- [RFC 3281] allows multiple attribute certificates per end user, but [RFC 3281] does not recommended it "since the administration and processing associated with such attribute certificate chains is complex and the use of attribute certificates in the Internet today is quite limited."

3 SUMMARY AND CONCLUSIONS

This section summarizes the findings discussed in Section 2. All findings address the problem noted in Section 1:

1. There is very little information in the public key certificate; and
2. Therefore, the RP needs more information about the end user, primarily for authorization decisions

Tables 3-1 through 3-6 summarize the findings. Each table focuses on a specific level of approach. Tables 3-1 and 3-2 are high-level summaries derived from details in tables 3-3 through 3-6. Tables 3-3 through 3-6 provide an extensive summary of each approach discussed in Section 2.

The high-level summaries address four core considerations and include an indication of the extent to which each alternative achieves the consideration:

- **Attribute Extensibility** – ability to add new AAs and/or acquire new attributes on demand (e.g., use of run time discovery mechanisms);
- **Confidentiality** – ability to preserve authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information (e.g., use of encryption, extent if user control and approval of private information that can be shared);
- **Information Assurance** – degree of confidence in the attribute information obtained (e.g, security mechanisms to ensure attributes are from a trusted parted and un-tampered, up-to-date attribute information); and
- **Complexity** – ease of implementation and maintenance

The color indicators derive from an objective, quantitative analysis of tables 3-3 through 3-6. If the number of checkmarks achieved for a consideration as a percentage of the total number of checkmarks possible is:

- <=33%, then the color indicator is red (low degree of meeting the consideration)
- >33% and <66%, then the color indicator is yellow (medium degree of meeting the consideration)
- >= 66%, then the color indicator is green (high degree of meeting the consideration)

The reader can add weighting to the tables to assist with analysis and decision-making.

In general, the Web Based Identity Standard, Attribute Authority Discovery approach offers the most complete solution. However, other solutions may be appropriate to meet the needs of the end user, RP, and/or IdP. Perhaps the most appropriate case for Web Based Identity Standards is that of identity federations and inter-federations where the coupling tends to be looser between AA and RP.

Table 3-1Level 1 Comparison[2]

Considerations	Attribute Extensibility	Confidentiality	Information Assurance	Complexity
Approaches	(per table 3-3)	(per table 3-4)	(per table 3-5)	(per table 3-6)
Prompt End User (Section 2.1)	🟡	🟢	🔴	🟡
Web Based Identity Standard (Section 2.2)	🟢	🟢	🟢	🟢
Provisioning (Section 2.3)	🔴	🟡	🟢	🟢
Attributes In Certificates (Section 2.4)	🔴	🔴	🟢	🔴

Table 3-2 Level 2 Comparison

Considerations	Attribute Extensibility	Confidentiality	Information Assurance	Complexity
Approaches	(per table 3-3)	(per table 3-4)	(per table 3-5)	(per table 3-6)
Prompt End User No Attribute Verification (Section 2.1.1)	🟡	🟢	🟡	🟢
Prompt End User Attribute Verification (Section 2.1.2)	🟡	🟡	🟡	🟡
Web Based Identity Standard Attribute Authority Discovery (Section 2.2.1)	🟢	🟢	🟢	🟢
Web Based Identity Standard Pre-Arranged Attribute Authority (Section 2.2.2)	🔴	🟡	🟢	🟢
Provisioning SPML (Section 2.3.1)	🔴	🟡	🟢	🟢
Provisioning Shared Directory (Section 2.3.2)	🔴	🔴	🟢	🟢
Attributes In Certificates Public Key Certificate Extensions (Section 2.4.1)	🔴	🔴	🟡	🔴
Attributes In Certificates Attribute Certificates (Section 2.4.2)	🔴	🔴	🟢	🟡

[2] 🔴=Low, 🟡=Medium, 🟢=High

Table 3-3 Level 3 Comparison, Attribute Extensibility

Considerations Approaches	Supports dynamic AA discovery	Does not require exchange of prior metadata between RP and AA	Supports user access to resources in real-time (attributes exchanged in advance or real-time)	Supports multiple AAs
Prompt End User No Attribute Verification (Section 2.1.1)		✓	✓	
Prompt End User Attribute Verification Out of Band (Section 2.1.2.1)		✓		
Prompt End User Attribute Verification Knowledge-based Authority (Section 2.1.2.2)		✓	✓	
Web Based Identity Standard Attribute Authority Discovery Thick Client (Section 2.2.1.1)	✓	✓	✓	✓
Web Based Identity Standard Attribute Authority Discovery Attribute Authority Certificate Extension (Section 2.2.1.2)	✓	✓	✓	✓
Web Based Identity Standard Pre-Arranged Attribute Authority Exchange Metadata (Section 2.2.2.1)			✓	
Web Based Identity Standard Pre-Arranged Attribute Authority Assertion-based Authentication (Section 2.2.2.2)			✓	
Provisioning SPML (Section 2.3.1)			✓	
Provisioning Shared Directory (Section 2.3.2)			✓	
Attributes In Certificates Public Key Certificate Extensions (Section 2.4.1)			✓	
Attributes In Certificates Attribute Certificates (Section 2.4.2)			✓	

Table 3-4 Level 3 Comparison, Confidentiality

Considerations Approaches	Attributes are retrieved directly from the AA (no third party other than the end user).	Supports exchange of specific attributes only when needed	Attribute information not shared with third party	Supports strong end user control/approval mechanism.
Prompt End User No Attribute Verification (Section 2.1.1)	✓	✓	✓	✓
Prompt End User Attribute Verification Out of Band (Section 2.1.2.1)		✓	✓	✓
Prompt End User Attribute Verification Knowledge-based Authority (Section 2.1.2.2)	✓	✓		
Web Based Identity Standard Attribute Authority Discovery Thick Client (Section 2.2.1.1)	✓	✓	✓	✓
Web Based Identity Standard Attribute Authority Discovery Attribute Authority Certificate Extension (Section 2.2.1.2)	✓	✓	✓	
Web Based Identity Standard Pre-Arranged Attribute Authority Exchange Metadata (Section 2.2.2.1)	✓	✓	✓	
Web Based Identity Standard Pre-Arranged Attribute Authority Assertion-based Authentication (Section 2.2.2.2)	✓		✓	
Provisioning SPML (Section 2.3.1)	✓		✓	
Provisioning Shared Directory (Section 2.3.2)				
Attributes In Certificates Public Key Certificate Extensions (Section 2.4.1)				
Attributes In Certificates Attribute Certificates (Section 2.4.2)	✓			

Table 3-5 Level 3 Comparison, Information Assurance

Considerations Approaches	Strong binding of user authentication to RP session	High level of assurance regarding attribute information obtained	Supports message layer security (signature, encryption)	Supports TLS
Prompt End User No Attribute Verification (Section 2.1.1)	✓			✓
Prompt End User Attribute Verification Out of Band (Section 2.1.2.1)	✓			
Prompt End User Attribute Verification Knowledge-based Authority (Section 2.1.2.2)	✓	✓		✓
Web Based Identity Standard Attribute Authority Discovery Thick Client (Section 2.2.1.1)	✓	✓	✓	✓
Web Based Identity Standard Attribute Authority Discovery Attribute Authority Certificate Extension (Section 2.2.1.2)	✓	✓	✓	✓
Web Based Identity Standard Pre-Arranged Attribute Authority Exchange Metadata (Section 2.2.2.1)	✓	✓	✓	✓
Web Based Identity Standard Pre-Arranged Attribute Authority Assertion-based Authentication (Section 2.2.2.2)		✓	✓	✓
Provisioning SPML (Section 2.3.1)	✓	✓	✓	✓
Provisioning Shared Directory (Section 2.3.2)	✓	✓		✓
Attributes In Certificates Public Key Certificate Extensions (Section 2.4.1)	✓	✓		
Attributes In Certificates Attribute Certificates (Section 2.4.2)	✓	✓	✓	✓

Table 3-6 Level 3 Comparison, Complexity

Considerations Approaches	Low Impact to CA Certificate Policy and/or Profiles	Low level of effort for RP/AA to implement and maintain	Standards Based	Can use COTS products (note: may still require customization)	Scalable	Convenient for end user (good usability)	Low impact to end user devices
Prompt End User No Attribute Verification (Section 2.1.1)	✓	✓			✓	✓	✓
Prompt End User Attribute Verification Out of Band (Section 2.1.2.1)	✓				✓		✓
Prompt End User Attribute Verification Knowledge-based Authority (Section 2.1.2.2)	✓			✓	✓	✓	✓
Web Based Identity Standard Attribute Authority Discovery Thick Client (Section 2.2.1.1)	✓	✓	✓	✓	✓	✓	
Web Based Identity Standard Attribute Authority Discovery Attribute Authority Certificate Extension (Section 2.2.1.2)	✓		✓	✓	✓	✓	✓
Web Based Identity Standard Pre-Arranged Attribute Authority Exchange Metadata (Section 2.2.2.1)	✓		✓	✓	✓	✓	✓
Web Based Identity Standard Pre-Arranged Attribute Authority Assertion-based Authentication (Section 2.2.2.2)	✓	✓	✓	✓	✓	✓	✓
Provisioning SPML (Section 2.3.1)	✓	✓	✓	✓	✓	✓	✓
Provisioning Shared Directory (Section 2.3.2)	✓	✓	✓		✓	✓	✓
Attributes In Certificates Public Key Certificate Extensions (Section 2.4.1)						✓	✓
Attributes In Certificates Attribute Certificates (Section 2.4.2)			✓			✓	✓

APPENDIX A: OVERVIEW OF RELEVANT TECHNOLOGIES

There are a number of technologies and standards that have been developed recently for exchanging identity and security information. The following sub-sections describe them.

A1. CardSpace

The Following explanation is per [CardSpace].

Microsoft Windows CardSpace (CardSpace) software enables people to maintain a set of personal digital identities that are shown to them as visual “Information Cards”. These cards are easier to use than passwords. Furthermore, they employ strong cryptography, making them significantly more secure than passwords and other information typed into web forms. There are three participants in any digital identity interaction using CardSpace:

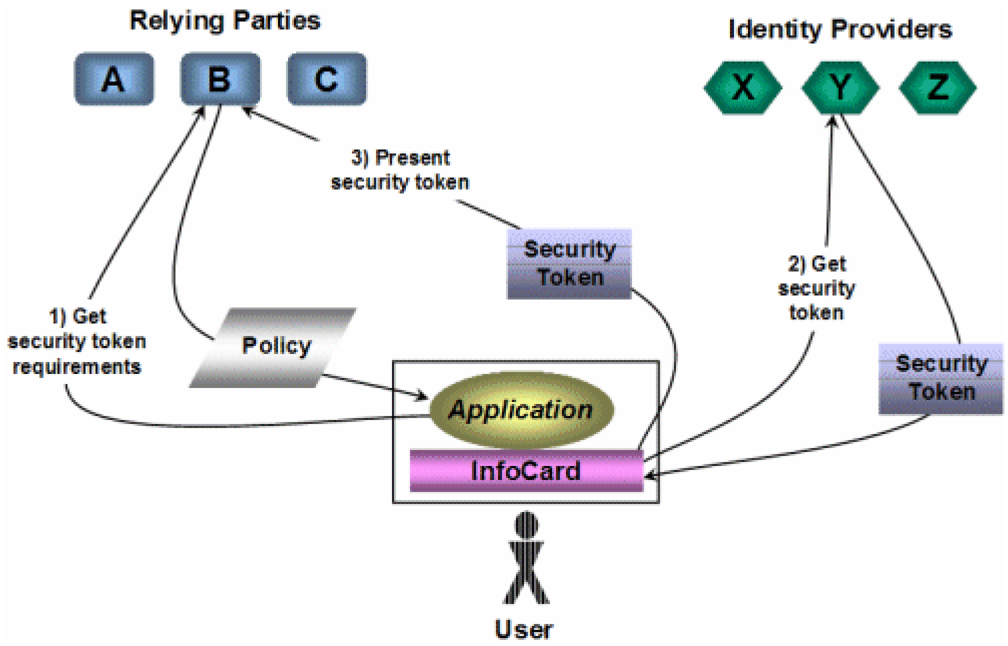
- IdPs issue digital identities for you. For example, businesses might issue identities to their customers, governments might vouch for the identities of their citizens, credit card issuers might provide identities enabling payment, online services could provide verified data such as age, and individuals might use self-issued identities to log onto web sites.
- RPs accept identities for you. Online services that you use can accept digital identities that you choose and use the information provided by them on your behalf, with your consent.
- You are in control of all these interactions. You choose which of your digital identities to have and which to use (or not to use), at places where they are accepted.

CardSpace and interoperable Information Card implementations by others enable the Internet to move beyond sole reliance upon username/password for sign-in. By using Information Cards, people will no longer need a username/password pair for every web site they log into. Because no passwords are used, they no longer need to be remembered, won’t be reused between sites, and can’t be stolen and used by others for fraudulent purposes.

Beyond being used to log into sites, Information Cards can also facilitate other kinds of interactions. The Information Card model provides great flexibility because cards can be used to convey any information from an IdP to an RP that makes sense to both of them and that the person is willing to release. One possibility is online age verification, with IdPs providing proof-of-age cards, and relying parties accepting them for purposes such as online wine sales; other attributes could be verified as well. Another is online payment, where merchants could accept online payment cards from payment issuers, containing only the minimal information needed to facilitate payment.

CardSpace software is based on open communication standards and interoperates with numerous components built by others, both for Windows and for other platforms. Together, these implement an interoperable "Identity Metasystem". CardSpace can be used to provide identities both for web sites and web services applications. CardSpace is Microsoft’s contribution to a widely accepted, broadly applicable, inclusive, comprehensible, privacy enhancing, security-increasing identity solution for the Internet.

Figure A1-1 CardSpace Interactions[3]



As A1-1 suggests, a user might rely on an application that supports CardSpace, such as a web browser, to access any of several RPs. She might also be able to choose from a group of IdPs (or AAs) as the source of the digital identity she presents to those relying parties. Whatever choice she makes, the basic exchange among these parties has three steps:

1. First, the application gets the security token requirements of the RP that the user wishes to access. This information is contained in the RP's *policy*, and it includes things such as what security token formats the RP will accept, and exactly what claims those tokens must contain.
2. Once it has the details of the security token this RP requires, the application passes this information to CardSpace, asking it to request a token from an appropriate IdP [or Attribute Authority].
3. Once this security token has been received, CardSpace gives it to the application, which passes it on to the RP. The RP can then use this token to authenticate the user or for some other purpose.

This high-level view illustrates the most important aspects of the process. They include the following:

- CardSpace and the identity metasystem are entirely agnostic about the format of the security token that's requested from an IdP and passed on to an RP. In fact, CardSpace typically isn't even aware of what format this token is in. Because of this, CardSpace can work with any digital identity system, using any type of security token, including simple usernames, X.509 certificates, Kerberos tickets, SAML tokens, or anything else. This allows CardSpace and the identity metasystem to be used together with whatever digital identity technologies are in place. It also allows plugging in yet-to-be-created digital identity systems that might appear in the future.

[3] This diagram is taken from http://msdn2.microsoft.com/en-us/library/aa480189.aspx

- All of the exchanges defined by the identity metasystem and implemented by CardSpace are done using open, published protocols.

In the most general scenario, an RP's policy is described using WS-SecurityPolicy, that policy is retrieved using WS-MetadataExchange, a security token is acquired using WS-Trust, and that token is conveyed to the RP using WS-Security (all of the WS-* protocols necessary to enable the secure exchange of identity tokens in the identity metasystem are (or soon will be) submitted to standards bodies).

In the simpler (and probably more common) scenario of a Web browser interacting with a website, the RP's policy can be expressed using HTML, and both this policy information and the security token can be exchanged with the site using HTTPS. While interactions with an IdP still depend on WS-Trust, a website isn't required to implement any of the WS-* specifications in order to act as an RP.

In either scenario, working with CardSpace does not require RPs or IdPs to implement any proprietary protocols.

Pros:

1. Highest mix of user control and security since the user can choose attributes and AAs and security provided by TLS and message layer security through WS-Security and WS-Security Policy.
2. CardSpace prevents Phishing because authentication requires the presence of the card. Even if an attacker retrieved the password to a card they would need to also obtain the card, which is located on the user’s machine.

Cons:

1. CardSpace requires the user’s devices to have a thick client installed.
2. The end user cannot easily roam to other desktops because his CardSpace information is loaded onto the computer with the thick client. Microsoft is working on ways to alleviate the roaming problem through devices such as fobs or mobile phones.

A2. Web Services Trust Language (WS-Trust)

WS-Trust provides a standard approach for exchanging security tokens through web services. Security tokens are a collection of claims that can include attributes about the user. Example claims include binary tokens such as proof of possession of a PKI certificate private key, security assertion markup language (SAML) messages, or simple attributes. Per [WS-TRUST], the web service security model defined in WS-Trust is based on a process in which a web service can require that an incoming message prove a set of claims (e.g., name, key, permission, capability). If a message arrives without having the required proof of claims, the service SHOULD ignore or reject the message.

Per [NIST SP 800-95], WS-Federation and WS-Trust were developed by IBM, Microsoft, RSA, Verisign, BEA, and several other web services vendors to develop an identity federation system based on extensions to WS-Security that uses the core web services protocols: SOAP, WSDL, and UDDI. WS-SecurityPolicy is a protocol that allows a web service to define a set of requirements detailing how messages should be secured and what tokens are required by the web service. It is used by WS-Trust to determine what tokens are needed to interact with a particular web service—these are referred to as a set of claims.

WS-Trust is used to exchange trust tokens between web services. It is an extension to WS-Security that provides methods for issuing, renewing, and validating security tokens as well as methods for establishing and brokering trust relationships between web services. If the requester web service does not supply appropriate “claims,” it can use the security policy declared by WS-SecurityPolicy to determine the URI of the provider’s Security Token Service (STS), who can provide the requester with the appropriate “claims.” Additionally, WS-Trust supports multi-messaging exchanges, allowing providers to use a challenge-response mechanism for authorization. Because WS-Trust builds upon WS-Security, claims can be anything from a digital signature to a X.509 certificate or an XML-based token, such as a SAML assertion.

WS-Federation expands on WS-Trust by providing various protocols by which STSs (interchangeably called IdP in WS-Federation), requesters, and providers can interact with one another to allow web services across organizational boundaries trust each other. Each organization is a “Trust Realm,” in which WS-Trust can be used between web services without the aid of WS-Federation. Additionally, WS-Federation provides two profiles for how requesters interact with providers and STSs: the active requester profile and the passive requester profile. The passive requester profile details how messages should be passed between a requester web browser, the provider, and the IP/STSs of both organizations so that WS-Federation can be used within the context of Web Applications, providing users with a single sign-on experience. The active requester profile details how web service requesters should interact with the provider and the IP/STSs to access a provider in another trust realm.

Sometimes, a service provider may not be able to perform the actions that a user or requester web service wishes it to perform, but it knows of a remote web service that can. The service provider may invoke another remote service to satisfy the requester’s request, which is known as *service chaining*. The service provider may use a SAML assertion, a WS-Security message, or both to make certain that both web services trust each other.

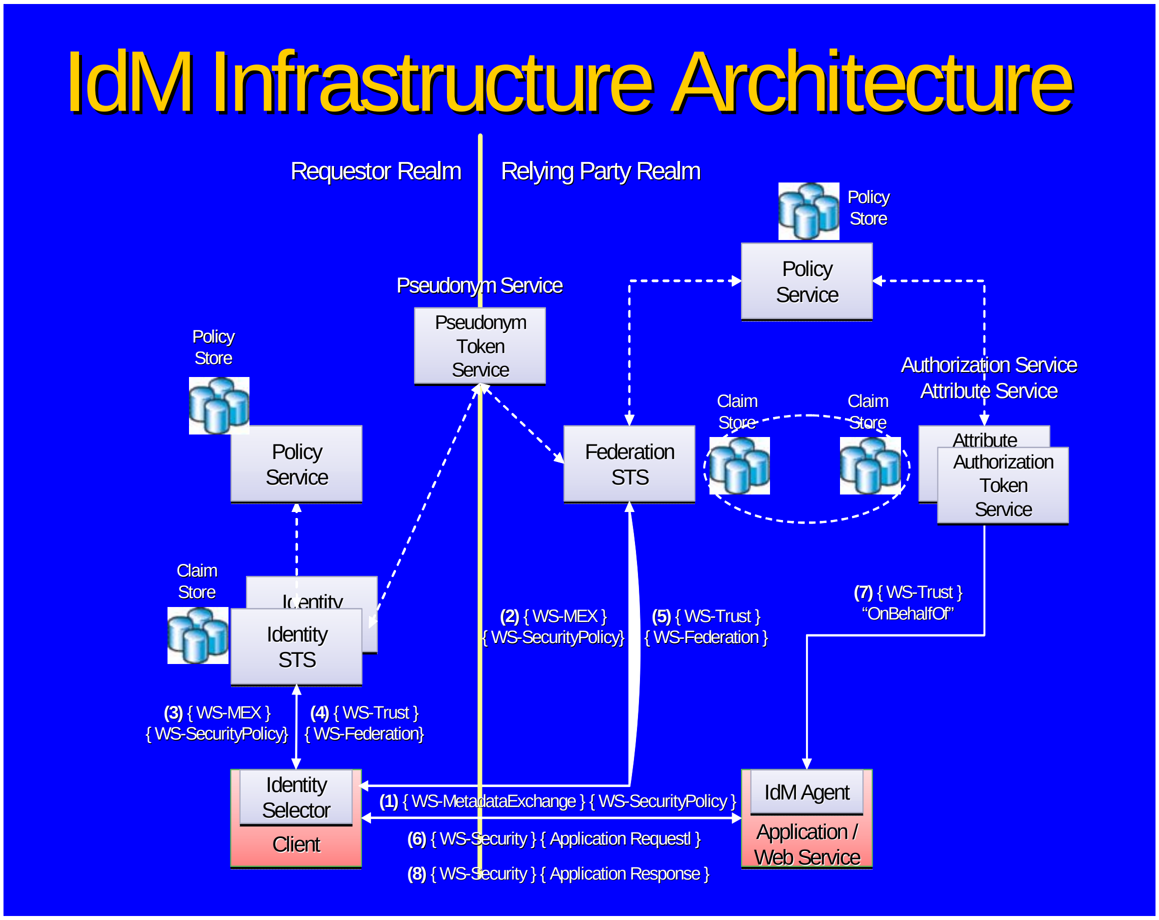
There are two different approaches to service chaining. The web service can access the remote web service either as itself or by taking on the identity of the originator of the request. In the first case, web service needs to be provided the identity of the originator in a trusted fashion. This can be satisfied using WS-Security and SAML.

There are three ways to pass the identity of the originator on to a remote service. First, if the web service received a SAML assertion with the originator's request, that SAML assertion can be passed on to the remote web service. It may be necessary for the requester service to sign either the SAML assertion or the SOAP message so that its own identity is passed to the remote service as well. Another option is for the requester service to generate and sign a SAML assertion for the originator itself and pass this SAML assertion on to the remote web service. In this type of configuration it is not possible for the remote web service to determine who originally requested the information. This limitation may be a hindrance in chain of trust deployments, as the chain is limited to the last web service requester.

By contrast, if the originator's SAML assertion is used or is signed, then it is possible to trace the request back to the requesting entity. By forwarding a SAML assertion to the remote wthe SOAP message using WS-Security or the SAML assertion, the remote web service is provided with the identity of the requester web service and can then determine whether or not it will trust the SAML assertion provided. If the authentication information requires confidentiality, SSL/TLS or WS-Security's encryption functionality should be used.

Figure A2-1 illustrates where WS-Trust fits in an enterprise solution.

Figure A2-1 WS-Trust Applicability in the Enterprise Solution[4]



IdM Infrastructure Architecture

Requestor Realm

Relying Party Realm

Policy Store

Policy Service

Pseudonym Service

Pseudonym Token Service

Policy Store

Policy Service

Authorization Service

Attribute Service

Claim Store

Claim Store

Federation STS

Attribute

Authorization Token Service

Claim Store

Identity

Identity STS

(2) { WS-MEX }
{ WS-SecurityPolicy}

(5) { WS-Trust }
{ WS-Federation }

(7) { WS-Trust }
"OnBehalfOf"

(3) { WS-MEX }
{ WS-SecurityPolicy}

(4) { WS-Trust }
{ WS-Federation}

Identity Selector

Client

(1) { WS-MetadataExchange } { WS-SecurityPolicy }

(6) { WS-Security } { Application Requestl }

(8) { WS-Security } { Application Response }

IdM Agent

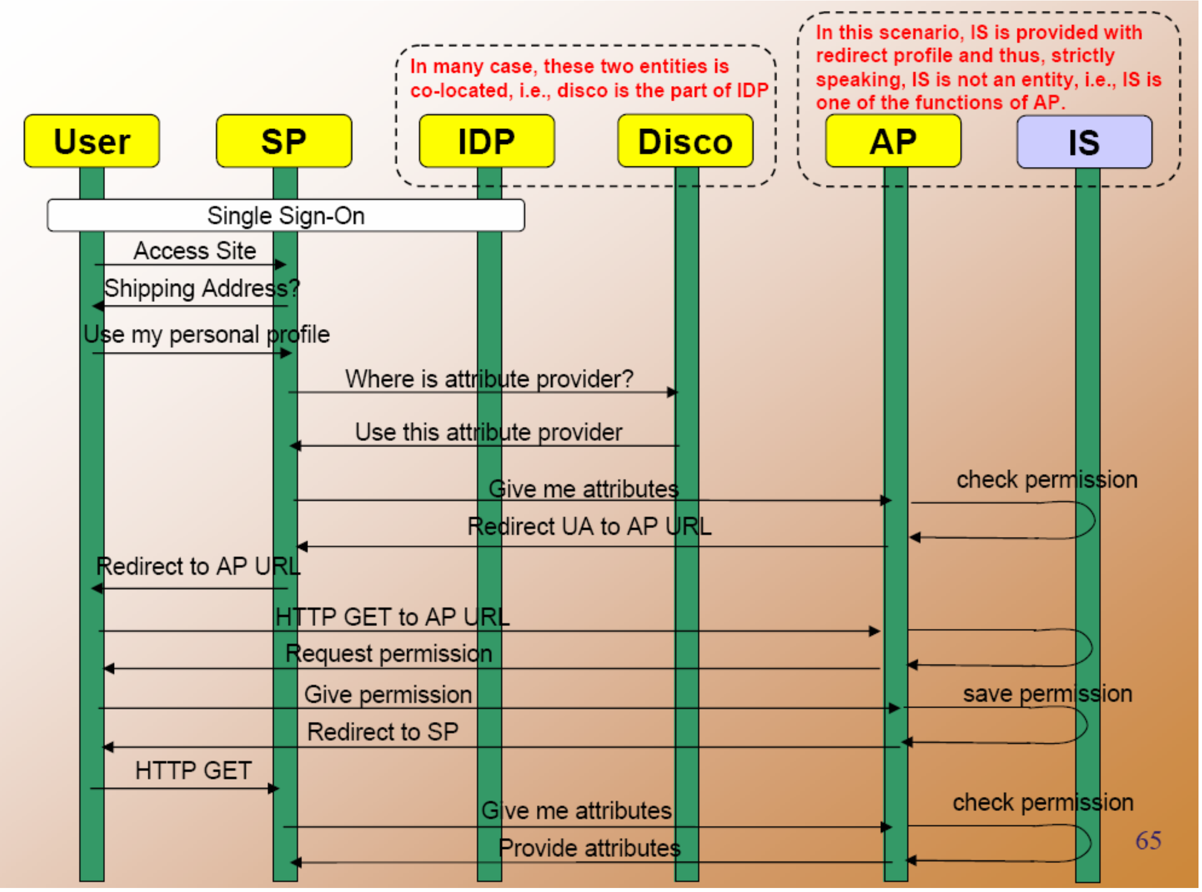
Application / Web Service

[4] Diagram taken from "Identity Federation Lessons Learned and the Way Forward", Don Schmidt, Principal Program Manager / Architect; Microsoft Corporation

A3. Liberty Identity Web Service Framework (ID-WSF)

The Liberty ID-WSF Discovery Service Specification defines the framework that enables a client to locate the appropriate web service for retrieving, updating, or modifying a specific piece of identity data. [Sun Reference] In terms of PKI rich attribute exchange, the web service (application) would use the discovery service to find the appropriate attribute service for the end user. AAs have the ability to register, update, and delete their metadata from the discovery service. There are two pre-defined services that could be useful in the context of rich attribute exchange. The first is the 'employee profile service' that describes attributes about an employee such as contact information and employee status. The 'personal profile service' contains attributes such as name, address, and face photo. ID-WSF makes it easy to create other attribute services to provide other information such as role-based information.

Figure A3-1 ID-WSF Sequence Diagram[5]



In the figure A3-1, the SP (Service Provider corresponds to RP) is trying to obtain attributes from an AP (Attribute Provider corresponds to Attribute Authority). The AP first interacts with the user to obtain permission through the IS (Interaction Service). Afterward, the SP contacts the AP directly to obtain the attributes.

[5] This diagram is taken from http://projectliberty.org/liberty/resource_center/tutorials

Pros:

1. User is not tied to one attribute service. As the attribute services for the end user changes, the Liberty Discovery Service is able to dynamically serve the appropriate results. When used with an authentication assertion, the assertion can be used as a token to provide assurance that the end user approves of the release of attributes from the attribute authority.

Cons:

1. The EPR (Endpoint Reference) URL must somehow be transmitted to the public key enabled application. Usually, the EPR is transmitted via an attribute in a SAML assertion. Since the end user is authenticating directly using an X.509 certificate, there is no associated assertion. The solution outlined in section 2.2.1.2 provides a method that could be used to obtain the ID-WSF Discovery Service via DNS.

A4. Security Assertion Markup Language (SAML)

The Security Assertion Markup Language (SAML) defines the syntax and processing semantics of assertions made about a subject by a system entity. [OASIS SAML-Core]. SAML is an XML-based framework developed by the Organization for the Advancement of Structured Information Standards (OASIS) that standardizes authentication and authorization data exchange between security domains. [WIKI SAML].

One component of a SAML assertion is an <AttributeStatement>. An <AttributeStatement> conveys information about a principal including name, address, role and other domain specific information. For example, the Internet2 effort has defined a specific profile called 'eduPerson' that includes widely-used person attributes in higher education.

The SAML attribute exchange itself is a request/response protocol. In the context of this paper, the RP would send an <AttributeQuery> to the end user’s Attribute Authority. The location of the Attribute Authority could be obtained by discovery, pre-configuration, or some other means. The Attribute Authority would send an <AttributeResponse> that correlates to the <AssertionQuery>.

Pros:

1. An RP has the ability to know what attributes a specific Attribute Authority offers by consuming SAML metadata.
2. Common attributes can be distributed by communities of interest.

Cons:

1. The Attribute Authority does not know that the end user approves of the distribution of his/her attributes. A possible solution is a digital signature on the <AttributeQuery> by the end user. This would require a thick client on the end user's device.
2. Metadata must be consumed by the RP. The location of the attribute authority metadata is in question when the end user authenticates using a certificate. A pre-configuration by the RP would be necessary in some cases.

A5. Service Provisioning Markup Language (SPML)

The following explanation is per [WIKI SPML].

Service Provisioning Markup Language (SPML) is an XML-based framework for exchanging user, resource and service provisioning information between cooperating organizations. It is the open standard protocol for the integration and interoperation of service provisioning requests. SPML is an OASIS standard. SPML version 1.0 was approved in October 2003. SPML version 2.0 was approved in April 2006.

Service provisioning refers to the "preparation beforehand" of IT systems' materials or supplies required to carry out a specific activity. It goes beyond the initial "contingency" of providing resources, to encompass the entire lifecycle management of these resources. This includes the provisioning of digital services such as user accounts and access privileges on systems, networks and applications, as well as the provisioning of non-digital or "physical" resources such as cell phones and credit cards.

Provisioning is the automation of all the steps required to manage (setup, amend and revoke) user or system access entitlements or data relative to electronically published services.[6]

The goal of SPML is to allow organizations to securely and quickly set up user interfaces for web services and applications, by letting enterprise platforms such as Web portals, application servers, and service centers generate provisioning requests within and across organizations. This can lead to automation of user or system access and entitlement rights to electronic services across diverse IT infrastructures, so that customers are not locked into proprietary solutions. For example, a supply partner (Company A) goes to its partner's (Company B) supply chain portal and requests access to its inventory data, which is stored in a back-office system. In response, Company B initiates a request using SPML to communicate with SPML-enabled identity management software. After automatically acquiring the appropriate permissions, Company B grants the appropriate access levels to Company A to gain access to the data it needs.[7]

This process takes place without the need for the portal environment to have an intimate understanding of the back-office environment. In other words, it is all automatic. The prototype encompasses all of the provisions of the proposed SPML standard while also leveraging the benefits of the Security Assertion Markup Language (SAML).

[6] This information was copied from http://www.openspml.org/spml_faq.html

[7] This information was copied from http://www.nwfusion.com/details/5623.html

Why have a provisioning standard:

1. Interoperability between provisioning systems
2. Organizations can share resources for maximum efficiency
3. Automate provisioning process as much as possible

SPML's Value[8]:

- SPML offers a scalable and practical provisioning standard
    - Breaks through current barriers to many Business scenarios using secure web services
    - Allows for the interoperability between existing disperse systems within your Business infrastructure, saving costs within the enterprise for these applications to work together (e.g., SPML-compliant provisioning systems, web access control, directories, metadirectory-based systems)
    - Supports the capability of having a "universal plugin/proxy" to provision to systems with minimum system functionality

Pros:

1. Provides full life-cycle user and account management
2. Useful when a new employee is added, corporate mergers occur or new applications are rolled out
3. Provides means to ensure assets are secured when employee leaves or business relationships are terminated
4. Requesting authority may batch sets of provisioning actions
5. Execution semantics for the batch can be controlled (e,g., "on fail do..")
6. Synchronous (wait for response) and asynchronous (don't wait – status query) processing model
7. Clients (e.g., Requesting Authority) can interrogate the SPML server for the details of the operations that it supports
8. Based on W3C XML Schema, and adds to it the object class definition and attribute-sharing model
9. Requesting Authority can request specific schemas

Cons:

1. SOAP with large binary file attachments not possible.

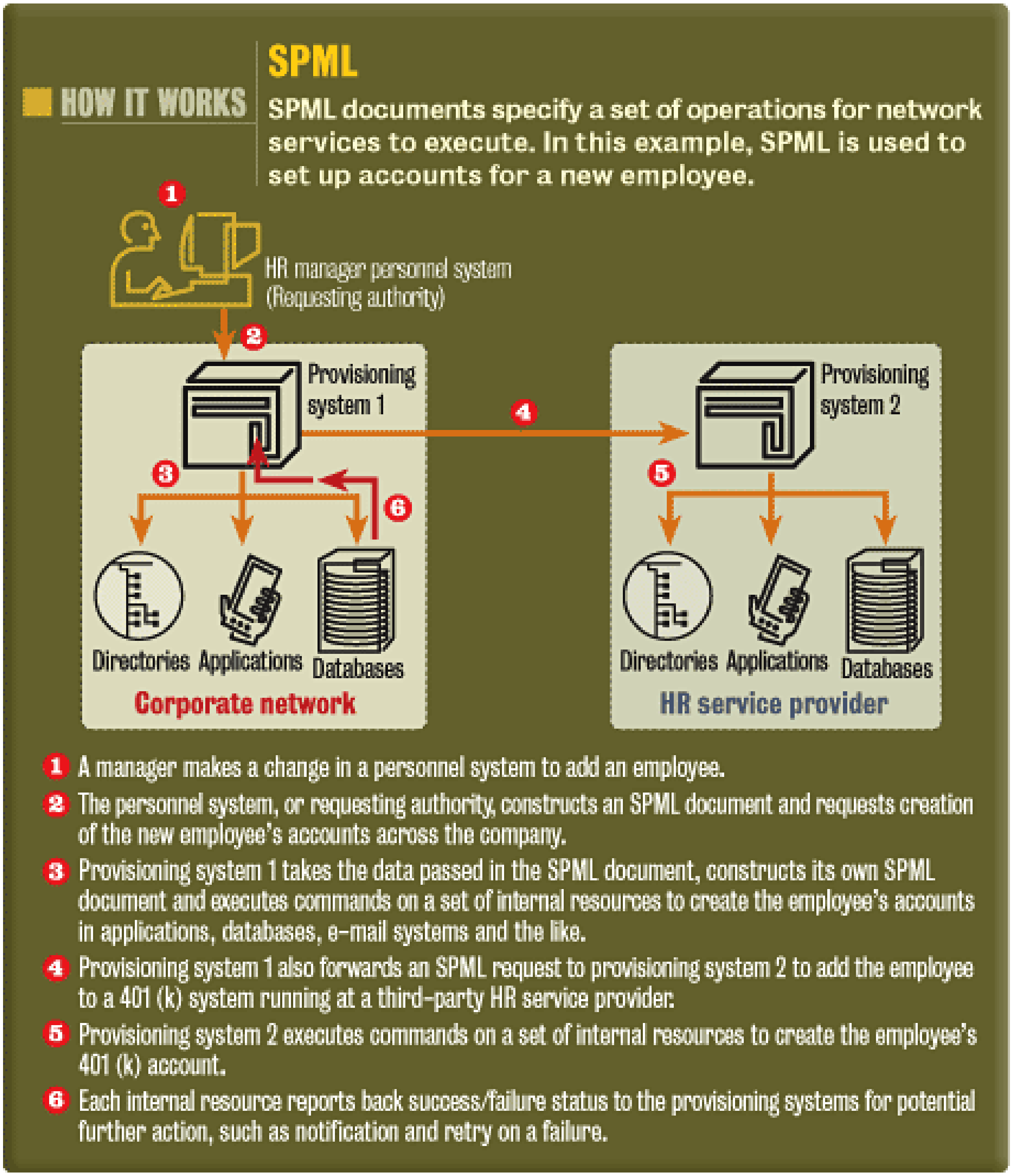
Figure A5-1 highlights how SPML works. Figure A5-2 illustrates where SPML fits in an enterprise solution.[9] Figure A5-3 describes the progression of the SPML standard to its current version 2.0.[10]

[8] "OASIS Updates" presentation; Gavenraj Sodhi; eTrust Security Management Solutions, Computer Associates

[9] Ibid

[10] Ibid

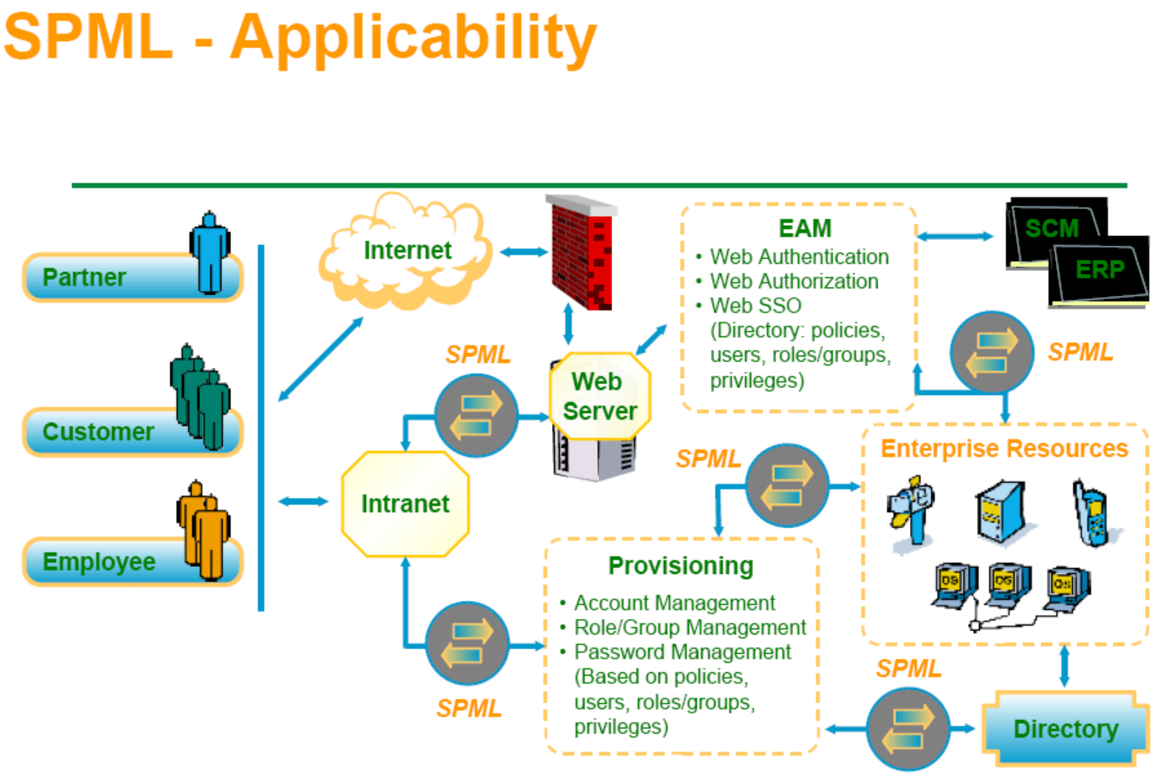
Figure A5-1 How SPML Works



SPML documents specify a set of operations for network services to execute. In this example, SPML is used to set up accounts for a new employee.

1. A manager makes a change in a personnel system to add an employee.
2. The personnel system, or requesting authority, constructs an SPML document and requests creation of the new employee's accounts across the company.
3. Provisioning system 1 takes the data passed in the SPML document, constructs its own SPML document and executes commands on a set of internal resources to create the employee's accounts in applications, databases, e-mail systems and the like.
4. Provisioning system 1 also forwards an SPML request to provisioning system 2 to add the employee to a 401 (k) system running at a third-party HR service provider.
5. Provisioning system 2 executes commands on a set of internal resources to create the employee's 401 (k) account.
6. Each internal resource reports back success/failure status to the provisioning systems for potential further action, such as notification and retry on a failure.

Figure A5-2 SPML Applicability in the Enterprise Solution



SPML - Applicability

Partner

Customer

Employee

Internet

Intranet

Web Server

SPML

EAM

- Web Authentication
- Web Authorization
- Web SSO (Directory: policies, users, roles/groups, privileges)

SCM

ERP

SPML

SPML

Enterprise Resources

Provisioning

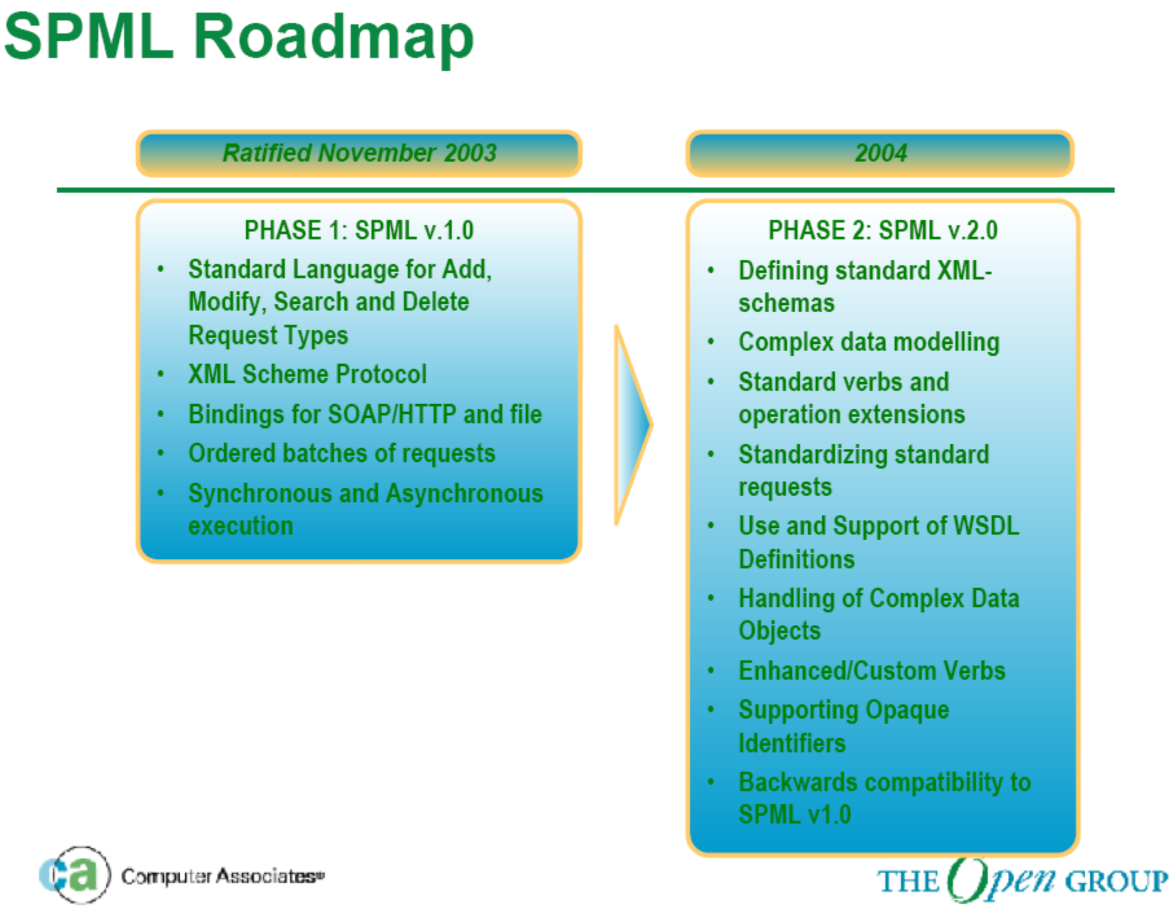
- Account Management
- Role/Group Management
- Password Management (Based on policies, users, roles/groups, privileges)

SPML

SPML

Directory

Figure A5-3 SPML Features Roadmap



SPML Roadmap

Ratified November 2003

PHASE 1: SPML v.1.0

- Standard Language for Add, Modify, Search and Delete Request Types
- XML Scheme Protocol
- Bindings for SOAP/HTTP and file
- Ordered batches of requests
- Synchronous and Asynchronous execution

2004

PHASE 2: SPML v.2.0

- Defining standard XML-schemas
- Complex data modelling
- Standard verbs and operation extensions
- Standardizing standard requests
- Use and Support of WSDL Definitions
- Handling of Complex Data Objects
- Enhanced/Custom Verbs
- Supporting Opaque Identifiers
- Backwards compatibility to SPML v1.0

Computer Associates®

THE Open GROUP

A6. Public Key Infrastructure (PKI)

In cryptography, a public key certificate (or identity certificate) is a certificate that uses a digital signature to bind together a public key with an identity (e.g., name of a person, an organization, address). The public key certificate can be used to verify that a public key belongs to an individual. [WIKI PKI].

Extensions to public key certificates are profiled in [RFC 3280]. Extensions are optional structures included in the certificate that "provide methods for associating additional attributes with users or public keys" [RFC 3280]. Standardized extensions include AuthorityKeyIdentifier, KeyUsage, and certificatePolicies. However, there are no standard extensions that express traditional authorization information (role, group, or other user profile information). In fact, inserting authorization information in a public key certificate would be undesirable:

> *"… authorization information often does not have the same lifetime as the binding of the identity and the public key. When authorization information is placed in a PKC extension, the general result is the shortening of the PKC useful lifetime. Second, the PKC issuer is not usually authoritative for the authorization information. This results in additional steps for the PKC issuer to obtain authorization information from the authoritative source."*
> [RFC 3281]

As a result, attribute certificates are used to address the need for rich user attributes. Attribute certificates have a similar structure to public key certificates except that there is no public key. Authorization information such as group, role, and security clearance are embedded in the attribute certificate. Attribute certificates also allow for domain specific attributes. Finally, an attribute certificate RP may accept an attribute certificate signed by an entity other than the certificate authority. This is because the certificate authority is often not the authoritative source of user attributes.

The use of attribute certificates has not been widely deployed. One reason attribute certificates have not been widely adopted is the issue of distribution. [RFC 3281] defines two distribution methods, "push" and "pull". The "push" method does not work with traditional web based application that use mutually authenticated TLS because the protocol does not allow an attribute certificate to be sent along with the authentication certificate. The "pull" method is also difficult because the RPy would need pre-knowledge of the location of the attribute certificate repository.

A7. Domain Name Service (DNS)

The following explanation is per [WIKI DNS].

On the Internet, the Domain Name Service stores and associates many types of information with domain names; most importantly, it translates domain names (computer hostnames) to IP addresses. It also lists mail exchange servers accepting e-mail for each domain. In providing a worldwide keyword-based redirection service, DNS is an essential component of contemporary Internet use.

The most basic use of DNS is to translate domain names (computer hostnames) to IP addresses. It is in very simple terms like a phone book. For example, if you want to know the internet address of en.wikipedia.org, DNS can be used to tell you it is 66.230.200.100. DNS also has other important uses.

Pre-eminently, the DNS makes it possible to assign Internet destinations to the human organization or concern they represent, independently of the physical routing hierarchy represented by the numerical IP address. Because of this, hyperlinks and Internet contact information can remain the same, whatever the current IP routing arrangements may be, and can take a human-readable form (such as "wikipedia.org") which is rather easier to remember than an IP address (such as 66.230.200.100). People take advantage of this when they recite meaningful URLs and e-mail addresses without caring how the machine will actually locate them.

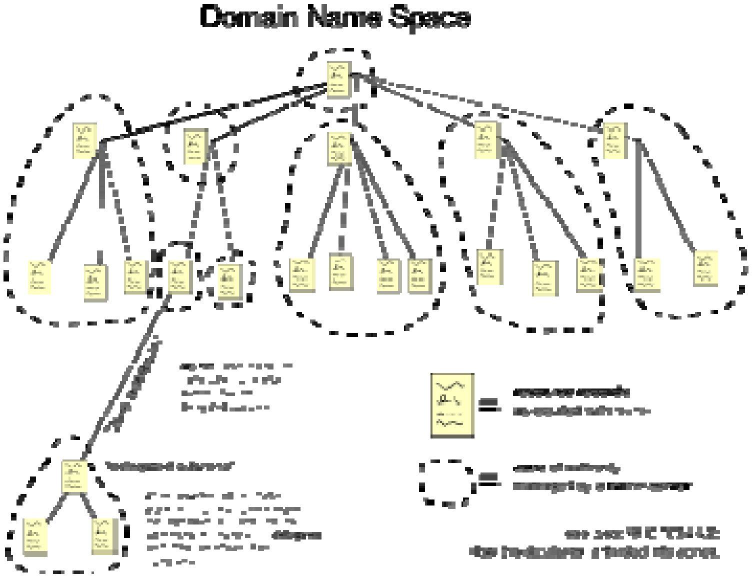
The DNS also distributes the responsibility for assigning domain names and mapping them to IP networks by allowing an authoritative server for each domain to keep track of its own changes, avoiding the need for a central registrar to be continually consulted and updated.

The domain name space consists of a tree of domain names. Each node or leaf in the tree has one or more resource records, which hold information associated with the domain name. The tree sub-divides into zones. A zone consists of a collection of connected nodes authoritatively served by an authoritative DNS nameserver. Note that a single nameserver can host several zones.

When a system administrator wants to let another administrator control a part of the domain name space within his or her zone of authority, he or she can delegate control to the other administrator. This splits a part of the old zone off into a new zone, which comes under the authority of the second administrator's nameservers. The old zone becomes no longer authoritative for what comes under the authority of the new zone.

A resolver looks up the information associated with nodes. A resolver knows how to communicate with name servers by sending DNS requests, and heeding DNS responses. Resolving usually entails iterating through several name servers to find the needed information. Figure A7-1 illustrates a DNS tree. In the figure, domain names are arranged in a tree, cut into zones, and each served by a nameserver.

Figure A7-1 DNS Tree



Users generally do not communicate directly with a DNS resolver. Instead DNS resolution takes place transparently in client applications such as web browsers, mail clients, and other Internet applications. When a request is made which necessitates a DNS lookup, such programs send a resolution request to the local DNS resolver in the operating system which in turn handles the communications required.

The DNS resolver will almost invariably have a cache (see above) containing recent lookups. If the cache can provide the answer to the request, the resolver will return the value in the cache to the program that made the request. If the cache does not contain the answer, the resolver will send the request to a designated DNS server or servers.

DNS was not originally designed with security in mind, and thus has a number of security issues. DNS responses are traditionally not cryptographically signed, leading to many attack possibilities; DNSSEC modifies DNS to add support for cryptographically signed responses. There are various extensions to support securing zone transfer information as well.

Some domain names can spoof other, similar-looking domain names. For example, "paypal.com" and "paypa1.com" are different names, yet users may be unable to tell the difference. This problem is much more serious in systems that support internationalized domain names, since many characters that are different (from the point of view of ISO 10646) appear identical on typical computer screens.

APPENDIX B: DOCUMENT REFERENCES

[CardSpace] A One-Page Introduction to Windows CardSpace
Michael B. Jones, Microsoft Corporation, January 2007
© 2007 Microsoft Corporation. All rights reserved.
cardspace.netfx3.com/files/folders/8932/download.aspx

[Claims] Building a Claims-Based Security Model in WCF]; Michele Leroux Bustamente
http://www.theserverside.net/tt/articles/showarticle.tss?id=ClaimsBasedSecurityModel

[Liberty ID-WSF] Liberty Alliance Identity Web Services Framework (ID-WSF) 1.1 Specifications
http://www.projectliberty.org/resourcecenter/specifications/libertyallianceid_wsf11specifications

[MS CardSpace] Introducing Windows CardSpace
http://msdn2.microsoft.com/en-us/library/aa480189.aspx

[MS WS-Trust] Organization for the Advancement of Structured Information Standards (OASIS) WS-Trust v1.3.
http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf

[MS WS-*] Web Services Specification
http://msdn2.microsoft.com/en-us/webservices/aa740689.aspx

[NIST SP 800-95] Guide to Secure Web Services; National Institute of Standards and Technology
http://csrc.nist.gov/publications/drafts/Draft-SP800-95.pdf

[OASIS SAML 2.0] Organization for the Advancement of Structured Information Standards (OASIS) Security Assertion Markup Language (SAML) v2.0.
http://docs.oasis-open.org/security/saml/v2.0

[OASIS SAML-Core] Organization for the Advancement of Structured Information Standards (OASIS) “Assertions and Protocol for the OASIS Security Markup Language (SAML) V2.0”, OASIS Standard, 15 March 2005.
http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[PKIX] Internet X.509 Public Key Infrastructure Subject Alternative Name for expression of service name; Internet Engineering Task Force (IETF)
http://www.ietf.org/internet-drafts/draft-ietf-pkix-srvsan-04.txt

[Public Key Cert] Internet X.509 Public Key Infrastructure Certificate and CRL Profile
http://www.ietf.org/rfc/rfc2459.txt

[RFC 822] Standard For The Format of ARPA Internet Text Messages
http://www.ietf.org/rfc/rfc0822.txt

[RFC 2630] Cryptographic Message Syntax
http://www.ietf.org/rfc/rfc2630.txt

[RFC 3280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
http://www.ietf.org/rfc/rfc3280.txt

[RFC 3281] An Internet Attribute Certificate Profile for Authorization
http://www.ietf.org/rfc/rfc3281.txt

[SPML] Organization for the Advancement of Structured Information Standards (OASIS) Service Provisioning Markup Language (SPML) v1.0
http://www.oasis-open.org/committees/download.php/3032/cs-pstc-spml-core-1.0.pdf

[Sun Reference] Sun Microsystems
http://docs.sun.com/app/docs/doc/819-2142/6n4evuvv2?a=view

[WIKI DNS] Wikipedia Encyclopedia
http://en.wikipedia.org/wiki/Domain_name_system

[WIKI PKI] Wikipedia Encyclopedia
http://en.wikipedia.org/wiki/Public_key_certificate

[WIKI SAML] Wikipedia Encyclopedia
http://en.wikipedia.org/wiki/Saml

[WIKI SPML] Wikipedia Encyclopedia
http://en.wikipedia.org/wiki/SPML

[WS-Trust SAML] Web Services Trust Language (WS-Trust)
http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-trust/ws-trust.pdf

APPENDIX C: GLOSSARY

Term	Definition
Assertion	A piece of data regarding either an act of authentication performed on a end user, attribute information about the end user, or authorization data applying to the end user with respect to a specified resource.
Attribute Authority	An Attribute Authority verifies and/or releases end user attributes in accordance with policy (i.e., what is released to who).
Attribute Certificate	Per [RFC 3281], An attribute certificate (AC) is a structure similar to a public key certificate (PKC); the main difference being that the AC contains no public key. An AC may contain attributes that specify group membership, role, security clearance, or other authorization information associated with the AC holder. Authorization information may be placed in a PKC extension or placed in a separate AC. The placement of authorization information in PKCs is usually undesirable for two reasons. First, authorization information often does not have the same lifetime as the binding of the identity and the public key. When authorization information is placed in a PKC extension, the general result is the shortening of the PKC useful lifetime. Second, the PKC issuer is not usually authoritative for the authorization information. This results in additional steps for the PKC issuer to obtain authorization information from the authoritative source. For these reasons, it is often better to separate authorization information from the PKC. Yet, authorization information also needs to be bound to an identity. An AC provides this binding; it is simply a digitally signed (or certified) identity and set of attributes. An AC may be used with various security services, including access control, data origin authentication, and non-repudiation.
Attributes	Specific information about an end user (e.g., first responder badge number, social security number, job title).
Authentication (AuthN)	Authentication is the process of establishing confidence in user identities. This is accomplished by establishing that someone is in fact who he or she claims to be. Authentication is typically a precursor to authorization.
Authoritative Source of Attributes	The authoritative source of attributes is the Attribute Authority (person or system) most directly relevant to the RP's use of the rich attributes.
Authorization (AuthZ)	Authorization is the process of giving someone, once identified (i.e., authenticated), permission to do or have something.
CardSpace	Windows CardSpace is the identity management component in Microsoft's upcoming .NET 3.0. With CardSpace, end users create a digital file on their own computer that includes user-specific information, like a work phone number, e-mail address or snail mail address. Users are able to save several different profiles and use whichever profile is appropriate for a certain situation.
Certificate Revocation List (CRL)	A list of revoked public key certificates created and digitally signed by a Certification Authority. Any certificate listed on the CRL must not be trusted.

Term	Definition
Certification Authority (CA)	A trusted entity that issues and revokes public key certificates. More specifically, a CA is an authority in a network that issues and manages security credentials and public keys for message encryption. As part of a public key infrastructure (PKI), a CA checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate. If the RA verifies the requestor's information, the CA can then issue a certificate. Depending on the public key infrastructure implementation, the certificate includes the owner's public key, the expiration date of the certificate, the owner's name, and other information about the public key owner.
Claims	An approach to security at service boundaries that improves on role-based and permission-based security models. Claims can represent many different types of information including identity, roles, permissions or rights and even general information about the end user that may be useful to the RP. A set of claims is also vouched for by an issuer such as a security token service, adding credibility to the information described by each claim – something not present in role-based or permission-based models. An additional benefit of using a claims-based security model is that it supports federated and single sign-on scenarios. [Claims]
Client	Name for a PC on a network.
Confidentiality	Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [44 U.S.C., Sec. 3542]
Cryptographic-based Assurance	Protecting information by encrypting it into an unreadable format.
Domain Name Service (DNS)	Service that translates numerical IP addresses into names to identify servers on the network.
End User	Any citizen, government employee, contractor, or business that uses an RP.
End User Devices	For example, an end user's web browser.
Extensible Markup Language (XML)	Specification developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.
Hypertext Transfer Protocol (HTTP)	Underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
Identity Provider (IdP)	Creates, maintains, and manages identity information for end users and provides end user authentication to RPs within a federation.
Identity Web Services Framework (ID-WSF)	The Liberty ID-WSF Discovery Service Specification defines the framework that enables a client to locate the appropriate web service for retrieving, updating, or modifying a specific piece of identity data.
Inter-Federation	Technical interoperation between different federations.

Term	Definition
Lightweight Directory Access Protocol (LDAP)	An application protocol for querying and modifying directory services running over TCP/IP. A directory is a set of information with similar attributes organized in a logical and hierarchical manner. An LDAP directory often reflects various political, geographic, and/or organizational boundaries, depending on the model chosen. LDAP deployments today tend to use Domain Name Service (DNS) names for structuring the topmost levels of the hierarchy. Deeper inside the directory might appear entries representing people, organizational units, printers, documents, groups of people or anything else which represents a given tree entry (or multiple entries).
Metadata	Information necessary for systems to technically interoperate. Systems exchange and configure metadata prior to operations.
Microsoft Vista	The next version of Windows for clients and servers. Vista supports thick clients at the end user device.
Naming Authority Pointer (NAPTR)	Newer type of DNS record that supports regular expression based rewriting.
On Demand	When needed.
Out of Band	Communication (e.g., exchange of information) via a means entirely separate from the current/original means of communication. For example, identity proofing documents may be submitted via mail after initial identity information is obtained online via electronic means.
Path Discovery and Validation (PDVal)	PDVal is used to validate a public key certificate. Path discovery is the process of locating all of the intermediate certificates and certificate revocation lists (CRLs) needed to validate an end entity certificate (e.g., the end user’s public key certificate) or determining that no valid certification path exists. Path validation is the process of verifying the discovered chain of certificates. Verification checks each certificate in the path for a variety of factors relevant to trust, including, but not limited to: • Verifying the digital signature on each certificate in the discovered path • Verifying that each certificate in the discovered path has not been revoked • Verifying that each certificate in the discovered path has not expired • Verifying that each certificate in the discovered path has a compatible assurance level
Privacy	See Confidentiality
Private Key	The secret part of an asymmetric key pair that is typically used to digitally sign or decrypt data.
Public Key Certificate	X.509v3 digital certificates in a Public Key Infrastructure (PKI) for authentication. A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a Subscriber to a public key. The certificate indicates that the Subscriber identified in the certificate has sole control and access to the private key.
Public Key Infrastructure X.509 Group (PKIX)	The Public-Key Infrastructure X.509 group, or PKIX, is a working group of the Internet Engineering Task Force dedicated to creating RFCs and other standards documentation on issues related to public key infrastructure (PKI) based on X.509 certificates.

Term	Definition
Relying Party (RP)	An entity that relies upon the subscriber's credentials (i.e., requires an end user to be authenticated), typically to process a transaction or grant access to information or a system.
Rich Attributes	Rich Attributes are the additional attributes needed by an RP (above and beyond those available from the standard act of authentication) to make authorization (e.g., role based access control) decisions.
Secure Sockets Layer (SSL)	Protocol for transmitting private documents via the Internet by using a private key to encrypt data transferred over the SSL connection.
Security Assertion Markup Language (SAML)	The set of specifications describing security assertions that are encoded in XML, profiles for attaching the assertions to various protocols and frameworks, the request/response protocol used to obtain the assertions, and bindings of this protocol to various transfer protocols (for example, SOAP and HTTP). SAML addresses web single sign-on, web services authentication, attribute exchange, authorization, non-repudiation, and secure communications. SAML defines assertion message formats that are referenced in Liberty Alliance, Shibboleth, WS-Security, and other specifications. SAML has become the standard web SSO identity management solution. Several versions have been released to date, including SAML 1.0, SAML 1.1, and SAML 2.0. The Organization for the Advancement of Structured Information Standards (OASIS) oversees SAML.
Service Provisioning Markup Language (SPML)	Service Provisioning Markup Language (SPML) is an XML-based framework for exchanging user, resource and service provisioning information between cooperating organizations. It is the open standard protocol for the integration and interoperation of service provisioning requests. The goal of SPML is to allow organizations to securely and quickly set up user interfaces for web services and applications, by letting enterprise platforms such as web portals, application servers, and service centers generate provisioning requests within and across organizations.
SOAP	Lightweight XML-based messaging protocol used to encode the information in web service request and response messages before sending them over a network. It consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP messages are independent of any operating system or protocol and may be transported using a variety of Internet protocols, including MIME and HTTP. Per W3C SOAP Version 1.2, "In previous versions of this specification the SOAP name was an acronym. This is no longer the case."
Step Down Translator (SDT)	A translator that facilitates technical interoperability between a certificate-based IdP and an assertion-based RP.
Strong binding between end user, credential and session	Mutually authenticated TLS utilizes cryptographic operations with the end user's private key to establish a symmetric key that can then be used to identify the end user's session at the RP. By involving the end user's private key in the creation of the session identifier, the session is tightly bound to the act of authentication.

Term	Definition
Thick Client	A thick (or fat) client does as much processing as possible and passes only data required for communications and archival storage to the server. This oftent means that the program(s) to do such processing resides locally on the user's device (e.g., computer) rather than the server.
Transport Layer Security (TLS)	TLS is a protocol created to provide authentication, confidentiality and data integrity between two communicating applications. TLS is based on a precursor protocol called "The Secure Sockets Layer Version 3.0" (SSL 3.0) and is considered to be an improvement to SSL 3.0. TLS is defined by [RFC 2246] and [RFC 3546]. TLS is effectively SSL version 3.1.
Trust	Trust is the characteristic where one entity is willing to rely upon a second entity (e.g., technically interoperate with another entity, accept as true all the claims sent by the other entity)
Universal Description Discovery and Integration (UDDI)	XML-based registry for businesses worldwide to list themselves on the Internet. Its ultimate goal is to streamline online transactions by enabling companies to find one another on the Web and make their systems interoperable for e-commerce.
Universal Resource Identifier (URI)	Compact string of characters used to identify or name a resource. The main purpose of this identification is to enable interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols.
Universal Resource Locator (URL)	See URI
Web Browser	Web browsers communicate with web servers primarily using HTTP (hypertext transfer protocol) to fetch web pages. HTTP allows web browsers to submit information to web servers as well as fetch web pages from them. Web pages are located by means of a URL (uniform resource locator), which is treated as an address. Cookies can be sent by a server to a web browser and then sent back unchanged by the browser.
WS-Security	Uses security tokens to facilitate SOAP message integrity and confidentiality. X.509 certificates and SAML assertions can be used as tokens. WS-Trust builds on WS-Security. WS-Security is broadly adopted, and has been approved by OASIS.
Web Service Definition Language (WSDL)	XML-based service description on how to communicate using web services. The WSDL defines services as collections of network endpoints, or ports.
WS-Trust	WS-Trust is used to exchange trust tokens between web services. It is an extension to WS-Security that provides methods for issuing, renewing, and validating security tokens as well as methods for establishing and brokering trust relationships between web services. Security tokens are a collection of claims that can include attributes about the user. Example claims include binary tokens such as proof of possession of a PKI certificate private key, SAML messages, or simple attributes.
X.500	Series of computer networking standards covering electronic directory services.

APPENDIX D: ACRONYMS

Acronym	Definition
AA	Attribute Authority
AP	Attribute Provider
ARPA	Advanced Research Projects Agency
AuthN	Authentication
AuthZ	Authorization
CA	Certification Authority
COTS	Commercial Off The Shelf
CRL	Certificate Revocation List
DNS	Domain Name Service
EPR	Endpoint Reference
FPKIPA	Federal Public Key Infrastructure Policy Authority
HTTP	Hypertext Transfer Protocol
IdP	Identity Provider
ID-WSF	Identity Web Services Framework
IS	Interaction Service
ISS	Identity Service Standard
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
NACI	National Agency Check and Inquiries
NAPTR	Naming Authority Pointer
NIST	National Institute for Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
PC	Personal Computer
PDVal	Path Discovery and Validation
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509
RFC	Request For Comment
RP	Relying Party
SAML	Secure Assertion Markup Language
SDT	Step Down Translator
SPML	Service Provisioning Markup Language
SSL	Secure Sockets Layer
STS	Security Token Service
TLS	Transport Layer Security
UDDI	Universal Description Discovery and Integration
URI	Universal Resource Identifier
URL	Universal Resource Locater
W3C	World Wide Web Consortium
WS	Web Services
WSDL	Web Service Definition Language
XML	Extensible Markup Language