Technical Approach for the Authentication Service Component

Version 1.0.0
June 28, 2004

Executive Summary

As part of the President's Management Agenda, the E-Authentication Initiative has been established to enable trust and confidence in E-Government transactions via the establishment of integrated policy and technical infrastructure for electronic authentication. After careful analysis and proofs-of-concept, the E-Authentication Program Management Office (PMO) decided to implement E-Authentication infrastructure as a federated architecture called the Authentication Service Component (ASC). The ASC leverages credentials from multiple credential providers through certifications, guidelines, standards adoption and policies. The ASC accommodates assertion-based authentication (i.e., authentication of PIN and Password credentials) and certificate-based authentication (i.e., Public Key Infrastructure (PKI) digital certificates) within the same environment. Over time, the ASC will support multiple schemes such as the Security Assertion Markup Language (SAML) and Liberty Alliance, and therefore is not built around a single scheme or commercial product. In this light, the ASC is more precisely defined as an architectural framework. The ASC is targeted for incorporation into the Federal Enterprise Architecture (FEA), as its government-wide authentication component.

The technical approach presented in this document is aligned with Office of Management and Budget (OMB) M-04-04, which provides policy guidance for identity authentication. It is also aligned with National Institute for Standards and Technology (NIST) SP 800-63, which is the technical companion document to OMB M-04-04. While the ASC architecture addresses authenticating end users to applications, authorization privileges at the application are beyond the scope of the ASC architecture and this document.

Core architectural requirements derived from the E-Authentication Strategic Plan are discussed, including high-level requirements (leverage credentials, single sign-on, privacy, and governance) and design goals (standards based, use of commercial off the shelf products, federation, durability, and flexibility). Key components (agency applications (AAs), credential services (CSs), end users, and the E-Authentication Portal) are defined and discussed, as well as session types within the framework (browser session, authentication session, and agency session). The technical approaches to assertion-based authentication and certificate-based authentication are then discussed in separate sessions, recognizing the significant difference between them.

The assertion-based authentication technical approach is discussed in terms of transaction flows of various use cases: (1) end user begins at the E-Authentication Portal, (2) end user starts at an AA, and (3) end user starts at a CS. Transactions flows also highlight single sign-on, which allows end users to move amongst AAs of equal or lesser assurance level without re-authenticating. Support of multiple schemes is shown via a transaction flow that seamlessly includes a Scheme Translator interposed between the different scheme protocols. A methodology for scheme adoption is also detailed.

The certificate-based authentication technical approach is discussed in terms of transaction flows of PKI use cases: (1) an AA uses a certificate validation service, and (2) an AA integrates validation software to perform local certificate validation. Various validation mechanisms are supported including, but not limited to, Online Certificate Status Protocol (OCSP), Simple Certificate Validation Protocol (SCVP), and XML Key Management Specifications (XKMS). The technical approach also supports use of PKI credentials at assertion-based AAs. A transaction flow for this use case is presented, highlighting a special Scheme Translator called a Step Down Translator that facilitates this.

PKI credentials offer considerable advantages for authentication. They can be validated using only public information. Standards for PKI are also more mature and more widely used than the emerging standards for assertion-based authentication of PIN and password credentials. The Federal PKI (FPKI) employs a Bridge Certification Authority (BCA) to harmonize policies and procedures for Certification Authorities (CAs). The E-Authentication Initiative defers assessment and governance of PKI based CSs to the FPKI Policy Authority (PA), the governing body for the Federal Bridge CA (FBCA).

The technical approach addresses exception scenarios. The E-Authentication Portal has a Uniform Resource Locator (URL) that any CS or AA can redirect an end user to in the event of a problem. The URL supports a number of standard error codes that can be passed to it. This approach helps to minimize the usability and exception handling burden on CSs and AAs, and ensures consistent exception processing throughout the architecture.

The technical approach supports secure Email by leveraging the PKI certificate validation techniques available in certificate-based authentication. Any Secure/Multipurpose Internet Mail Extensions (S/MIME) capable email software product can be used to process signed and encrypted email. Four use case transaction flows are discussed: (1) email application requests certificate verification from validation service, (2) email application validates the certificate directly by running certificate validation software on the end user's desktop, (3) email application uses a dedicated validation service for organizations who trust certification authorities that are not trusted government-wide, and (4) a combination of the previous options.

Additionally, the technical approach supports secure submission of electronic forms. Some E-Government business is performed with electronic form applications rather than web forms. These applications do not have the same characteristics as browser-based applications. Two use case transaction flows are discussed: (1) certificate-based authentication of electronic forms, and (2) pop up an E-Authentication browser window in the electronic form to leverage all E-Authentication CSs and to minimize the need to customize electronic forms applications.

Table of Contents

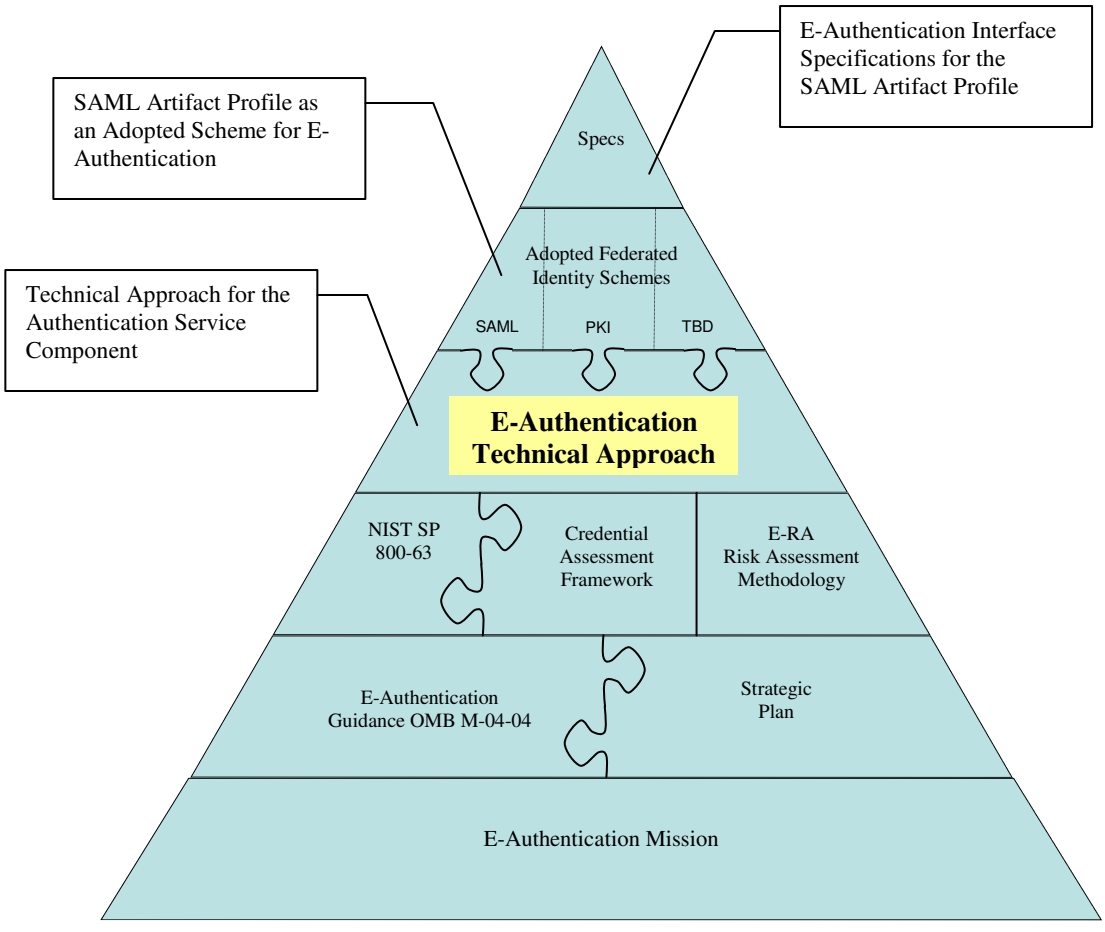
1 Introduction

1.1 Purpose of this Document

The purpose of this document is to set the technical direction and approach for the Authentication Service Component (ASC). This is considered a technical document and is intended for a technical audience familiar with the E-Authentication Initiative. For additional information on E-Authentication please visit http://www.cio.gov/eauthentication/. Generally, this document is intended to describe the architectural framework under which the E-Authentication Program Management Office (PMO) will implement technologies, products and technical standards to meet its program objectives. This document further provides a methodology for the graceful adoption of new schemes as they emerge.

This document is not autonomous, but builds upon of the Office of Management and Budget E-Authentication Guidance for Federal Agencies (OMB M-04-04), National Institute for Standards and Technology Recommendation for Electronic Authentication (NIST SP 800-63). Additional technical specifications build on this document. Figure 1 shows the documentation relationships for E-Authentication.

Figure 1: E-Authentication Document Hierarchy



This document is part of the ASC technical suite, which also includes Security Assertion Markup Language (SAML) Artifact Profile As an Adopted Scheme for E-Authentication and E-Authentication Interface Specifications for the SAML Artifact Profile. For complete comprehension, this document should be read prior to the Adopted Scheme and Interface Specifications. The most recent version of these documents are available at the E-Authentication website, http://www.cio.gov/eauthentication/.

1.2 The E-Authentication Concept

As part of the President's Management Agenda, the E-Authentication Initiative will ultimately enable trust and confidence in E-Government transactions through the establishment of an integrated policy and technical infrastructure for electronic authentication. Through the initiative, citizens and businesses will have simpler access to multiple agency applications (AAs) through the re-use of credentials and established identities.

The principal goal of the E-Authentication Initiative is to provide the ASC, a government-wide authentication component for the Federal Enterprise Architecture (FEA). The FEA includes several reference models, including the Performance Reference Model (PRM), the Business Reference Model (BRM), the Service Component Reference Model (SRM), and the Technical Reference Model (TRM). Specifically, the E-Authentication Initiative resides in the SRM, providing security management services within the Support Services domain. In general, the initiative aligns to the PRM through its mission of increasing the public trust, to the BRM by supporting the delivery of services, and to the TRM by identifying technologies and standards relevant to E-Authentication.

The E-Authentication concept is best described through the trust relationships among AAs, Credential Service Providers (CSP) and end users. CSPs are commercial or government entities authorized by the PMO to provide credentials (e.g., Personal Identification Numbers (PINs), Passwords, Digital Certificates) to potential end users for access to government systems. AAs are government applications, systems or services that rely on (or trust) the authentication/credential services (CS) of CSPs. End users are people or organizations that have credentials issued by a CSP and desire to use that credential to conduct business with an AA. It is the management of trust among these entities (AA, CSPs and end users) that is the essence of the E-Authentication Initiative. E-Authentication provides:

- Policies and guidelines for federal authentication
- Credential service assessments
- Interoperability testing of candidate products, schemes or protocols
- Management and control of accepted federation schemes operating within the environment

To manage the trust relationships, the PMO does not envision building an authentication infrastructure as a central broker for these entities. Instead, the ASC will adopt a federated architecture that leverages credentials from multiple domains through certifications, guidelines, standard adoption and policies. The architectural framework accommodates the use of assertion-based credentials (PIN and Passwords) as well as certificate-based credentials within the same environment. Over time, the architecture will leverage multiple emerging schemes such as the SAML and Liberty Alliance, and will not be built around a single scheme or commercial product.

1.3 Scope

The E-Authentication technical approach is aligned with OMB M-04-04, which provides policy guidance for identity authentication, not authorization or access control. Specifically, the ASC implements the identity authentication recommendations documented in NIST SP 800-63, which is the technical companion document to OMB M-04-04.

NIST SP 800-63 provides recommendations for humans authenticating to applications and does not specifically address whether the end user is using a browser or some other mechanism to access the application. E-Authentication's core architecture is specifically aligned with humans using a web browser for authenticating to applications. Authorization privileges at the application are beyond the scope of this document, NIST SP 800-63, and the ASC. Authorization and related functionality such as access control, entitles, and provisioning are left to the application owners.

Many features and scenarios were considered carefully but are not addressed in these guidance documents because of the need to properly balance utility, complexity, and patience with industry standards. Approaches for secure email and electronic forms applications are provided in appendices. Other features, including Personal Identifiable Information (PII) and attribute sharing, service discovery, single sign-off, non-browser thin clients (e.g., personal digital assistant, cell phone), cell phone proxies, billing and charge-back protocols, group/role identification, trust agent and power of attorney scenarios have been deferred to a later revision to allow industry standards and government requirements to mature.

1.4 Requirements

The vision and direction for the E-Authentication Initiative is contained in the E-Authentication Strategic Plan. The strategic plan presents specific actionable tasks to achieve the E-Authentication Mission. The following architectural requirements are derived from the strategic plan:

High Level Requirements:

1. **Leverage**: A credential from any approved CS should be usable at any application of equal or lower assurance level. AAs must be able to leverage existing credentials rather than establish new credentialing systems.
2. **Single Sign-on**: Once an end user has authenticated, he or she must be able to move among applications with equivalent (or lower) assurance levels without re-authenticating. For privacy considerations, end users must take explicit actions to opt-in both at the E-Authentication Portal to set a browser-session CS preference and at the CS to enable single sign-on.
3. **Privacy**: There must be no central audit log of which end users accessed which applications and no centralized electronic authentication system. Credentialing must be federated among multiple providers.
4. **Governance**: The architectural framework must provide for explicit control over which applications and credential services can join the E-Authentication community.

Design Goals:

1. **Standards**: The architectural framework should rely on existing industry standards while remaining cognizant of emerging standards.

2. **Commercial off the Shelf (COTS)**: The architecture should employ COTS products wherever possible.
3. **Federation**: Authentication should be federated among multiple CSPs
4. **Durability**: The architectural framework should be designed to allow for the evolution of technology, providing for easy migration as the industry evolves.
5. **Flexibility**: The architectural framework should not rely on any single standard, vendor, product, or integrator.

2 Approach

The technical approach for E-Authentication is based on an architectural framework that allows for the co-existence of multiple federated identity schemes within a single architecture. The architectural framework includes a methodology and process for evaluating and adopting these schemes over time. The goal of the architectural framework is to provide a lasting architectural model for E-Authentication that is not irrevocably bound to a single industry standard, vendor, or product. This is in accordance with the previously mentioned E-Authentication Design Goals and NIST SP 800-63 directive that the E-Authentication approach be technology neutral, if possible.

The approach is presented through use cases depicting the high level interaction of E-Authentication components in various scenarios. The major sub-components of E-Authentication are:

1. **Agency Applications:** E-Government applications that perform some business function online. If an E-Government application has multiple interfaces (e.g., administration and service application), each interface with distinct authentication requirements is considered a stand-alone AA. AAs manage all business transactions and all end user authorization decisions. One of the principal goals of the E-Authentication Initiative is to provide broad authentication services to AAs, allowing the complete deferral of identity management.
2. **Credential Services:** Services that provide end users with credentials that can be used at E-Authentication-enabled AAs. CSs are provided by CSPs[1], which are companies or agencies that operate one or more CS.
3. **E-Authentication Portal (Portal)**: A website that helps end users locate the CSs and AAs they need to complete their transactions. The Portal also maintains information about CSs and AAs, referred to as metadata, which includes technical interface data as well as descriptive information.
4. **End Users**: Any citizen, government employee, contractor, or business that authenticates to an AA using a credential issued by a CS. One of the principal goals of E-Authentication is to make the end user experience as simple as possible by improving the availability and ease of use of credentials.

Within the architectural framework, the end user interacts directly with AAs, CSs, and the Portal. Typically, the end user starts at the Portal in order to locate the appropriate AAs and CSs. The end user interacts with the CS to obtain, manage, and validate credentials. The CS interacts directly with the AA in order to pass the end user's identity information, so the AA knows with whom it is dealing. Once the AA knows the identity information, the end user interacts directly with the AA for business transactions. Authorization is handled completely by the AA.

Governance is accomplished by managing the interaction between the AAs and CSs. The government will issue credentials to approved AAs and CSs, which will be validated before the end user's identity information is handed off.

There are three types of sessions discussed in the architectural framework:

1. **Browser Session:** The period of time the end user's browser is open. The browser session begins when the end user opens the browser and ends when it is closed. All session cookies are terminated when the browser session ends. Any browser with Transport Layer Security

[1] CSPs are sometimes referred to as Electronic Credential Providers (ECPs) in other documents.

(TLS) and session cookie support can be used with the ASC, although individual AAs and CSs may have additional requirements.

2. **Authentication Session:** The period of time that an end user remains trusted after the end user authenticates. A CS typically does not require an end user to re-authenticate for every page requested; they continue to be trusted for some period of time after each authentication. The CS's allowed period between re-authentication is referred to as the authentication session.
3. **Agency Session:** The period of time an AA will trust an end user before handing the end user off to the CS for re-authentication. AAs do not have access to authentication session information; they must maintain their own session with an end user and decide how long an end user remains authenticated once starting a transaction.

The ASC technical approach has two different architectural models, assertion-based authentication and certificate-based authentication. PIN and Password (including one time use passwords) authentication uses assertion-based authentication, where end users authenticate to a CS, which then asserts their identity to the AA. Assertion-based authentication can be used as defined in NIST SP 800-63 for assurance levels 1 and 2. Considering certificate-based authentication relies on X.509v3 digital certificates in a Public Key Infrastructure (PKI) for authentication, it can be used at any assurance level. Section 3.1 describes the approach for assertion-based authentication; section 3.2 describes the approach for certificate-based authentication.

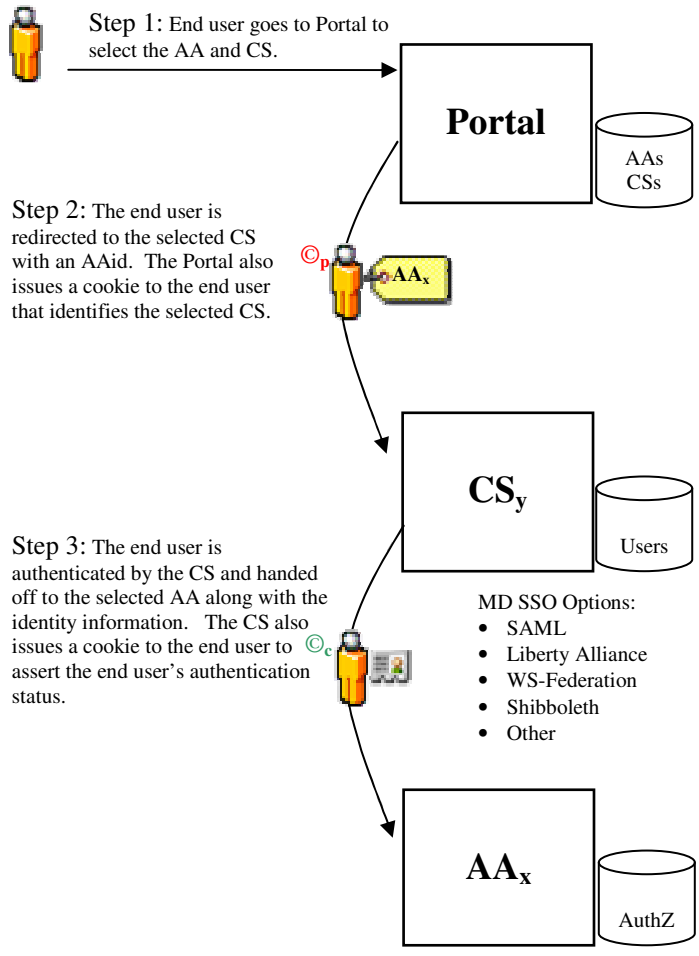
3 Assertion-Based Authentication

Currently, there are a number of standards based schemes for federation that are at various stages of maturity, including Liberty Alliance, WS-Federation, SAML, and Shibboleth. Any of these schemes could be used to meet the assertion-based requirements of E-Authentication. It is unclear which of these schemes will become dominant in the market, and it is quite possible that more than one will be in common use. The following sections describe the architectural framework for assertion-based authentication, show where the federation schemes fit in, and describe how multiple schemes can be leveraged simultaneously.

3.1 Base Case

The Base Case is the foundation of the architectural framework, and all other use cases build and expand upon this case. Figure 2 depicts the sequence of events for the Base Case. In Step 1, the end user goes to the Portal and selects an AA. The Portal then presents the end user with a list of CSs with appropriate assurance levels. Once the CS has been selected, the end user is redirected to the CS with the identifier for the selected AA (AAid), shown in Step 2. As part of this redirect, the Portal gives the end user a session cookie indicating which CS the end user has selected, which will remain operational for the duration of the browser session. This cookie is used to enable single sign-on in later transactions. The end user then authenticates to the CS directly, and the CS assigns a session cookie to manage the authentication session. In Step 3, the CS passes the authenticated end user on to the AA along with the identity information, allowing the AA to manage transactions and authorization. Typically, the AA will assign a cookie to manage the agency session.

Figure 2: Base Case



Since the hand-off to the AA includes the identity of the end user, some PII will be included. The CS may adjust the PII made available to a given AA based on the end user's preferences, their privacy policies, or by prompting the end user before the hand-off. E-Authentication interface specifications specify the minimum set of identity attributes required for all hand-offs.

The hand-off from the CS to the AA shown in Step 3 is a classic case of Multi-Domain Single Sign-On (MD SSO); an end user authenticated in one domain (the CS) needs to become known to another domain

(the AA) without re-authenticating. This hand-off is where the various federation schemes can be used. Currently SAML, Liberty Alliance, Shibboleth, and WS-Federation schemes all provide mechanisms for MD SSO. Other standards based mechanisms are likely to become available and existing schemes are likely to evolve.

The impact from the uncertain future of industry standards is isolated to this final step of the Base Case. Section 3.4 discusses how various schemes can co-exist within the architectural framework, and how graceful migration away from dying schemes is accomplished. The following sections describe additional use cases within the architectural framework.

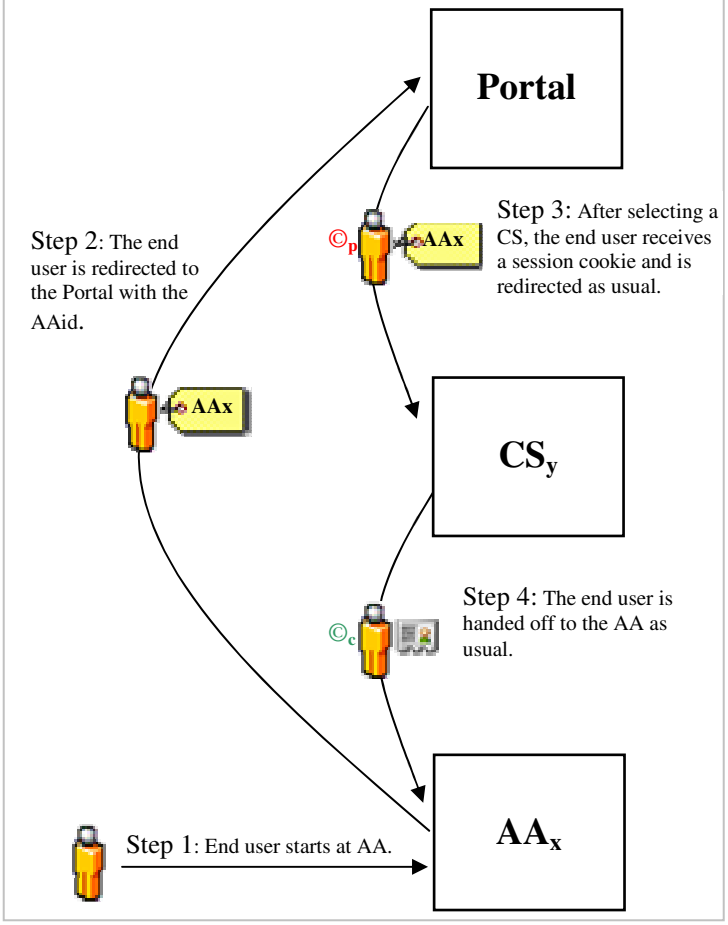
3.2 Starting at the Agency Application

It is unlikely that all end users will start at the Portal. Figure 3 depicts the sequence of events for end users that start at the AA in Step 1. All applications in the architectural framework must be configured to redirect any unauthenticated end user to the Portal when attempting to access a protected resource. When the end user is redirected, the AAid is included and passed to the Portal, as shown in Step 2. The Portal does not have to ask the end user to select an application; it can simply display a list of appropriate CSs for selection by the end user. If the end user has previously authenticated during this session, the Portal cookie will present the Portal with the CS previously selected by the end user, allowing the Portal to immediately redirect the end user without any end user interaction. If the assurance level of the previously selected CS is insufficient for the AA requested, the Portal can notify the end user and allow them to select an alternative CS.

Once the end user has been redirected to the CS in Step 3, the end user is handed off to the AA as described in Step 4. The end user authenticates to the CS and is handed off to the originating AA using one of the MD SSO schemes. If the end user has previously authenticated during this browser session, then the CS cookie can be used to determine the end user's identity and the CS can initiate the hand-off without any end user interaction, completing the single sign-on sequence.

The combination of the Portal cookie and the CS cookie is the mechanism for architecture-wide single sign-on, regardless of the MD SSO scheme being used. Once an end user has authenticated the first time, subsequent visits to other applications during the session will not require re-authentication. After authenticating, an end user moving from one application to another will automatically become known to other applications. The Portal cookie allows the end user to be redirected to the CS without end user interaction, and the CS cookie allows the end user to be passed to the AA without interaction, providing seamless and invisible single sign-on.

Figure 3: End user Starts at AA



Since the CS is directly involved in the single sign-on, it has an opportunity to intervene if the end user has opted out of the single sign-on capability, or if its privacy policy prevents it. Single sign-on is one of many end user preferences. The management of these end user preferences is encapsulated along with other identity management issues at the CSP; there is no need for a government-wide repository of these preferences.

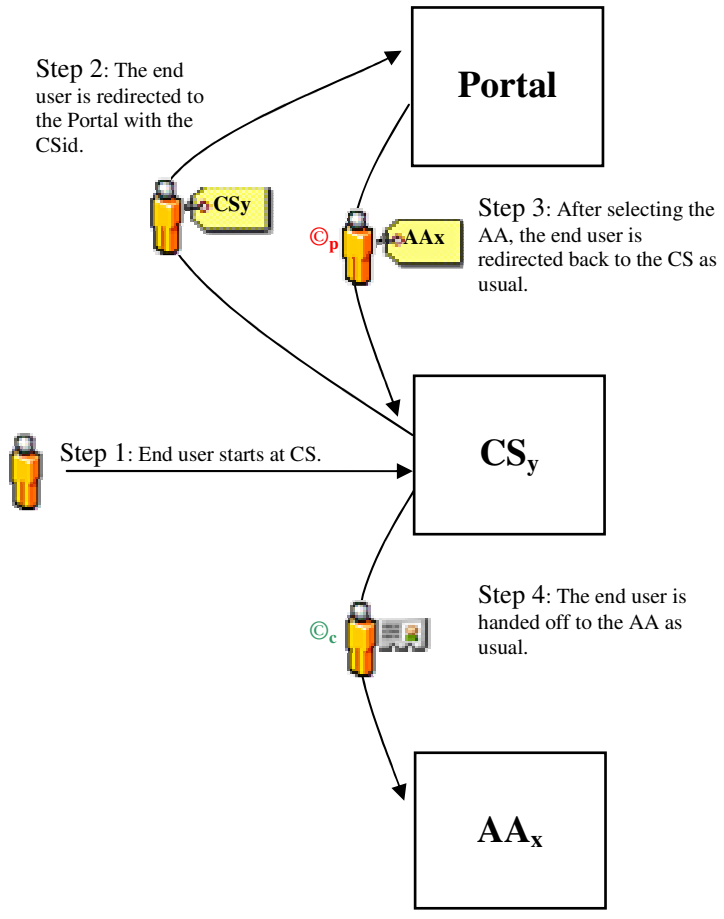
3.3 Starting at the Credential Service

In some cases the end user may begin a session at the CS. For example, a bank that is integrated with E-Authentication may provide a link to the government Portal and inform the end user that the credential may be used to conduct government business. The end user, who is already authenticated to the bank and conducting business, may select the link and begin a session. Figure 4 depicts the sequence of events for this case.

The end user starts at the CS and selects a link to the Portal that includes a CS identifier (CSid), as shown in Step 1. In Step 2, the end user is redirected to the Portal and presented a list of applications that may be accessed using the CS, as well as some indication that other applications may be available for higher level credentials. The end user selects an application and is then redirected back to the originating CS with the AAid, as shown in Step 3. In step 4, the end user is handed off to the AA as usual. If the end user has already authenticated to the CS, then the end user could immediately be handed off to the AA.

This use case demonstrates how CSs can advertise the utility of their credential, increasing the value proposition for CSPs. It also opens up every CS as a channel to advertise the availability of various AAs.

Figure 4: User Starts at CS



The use case further illustrates the flexibility of the Portal. In addition to supporting single sign-on, the principal function of the Portal is to help the end user select the CS and AA. If the end user explicitly makes one of those selections before accessing the Portal, the architectural framework allows the Portal to avoid redundant end user interaction. This capability reduces the required click count and generally simplifies the end user experience.

See Appendix A for more use cases leveraging the flexibility of the Portal.

3.4 Multiple Scheme Support

The description of the Base Case specified a role for schemes like Liberty Alliance, SAML, Shibboleth, and WS-Federation. These schemes specify protocols and standards for federated identity mechanisms that allow different entities to share identities without requiring the end user to manage multiple accounts. In the architectural framework, these schemes are used in the final step of the Base Case to hand-off of the end user from the CS to the AA. This is also referred to as MD SSO, where an end user who has authenticated to one domain (the CS) becomes known to another domain (the AA) without re-authenticating.

It is currently unclear which of these schemes will become dominant in the market. Each is based on a different philosophy and approach, and each is at different levels of maturity. The only thing known for certain is that the future is unpredictable. The architectural framework is designed to be durable and flexible, avoiding reliance on a single standard. Even if a single standard had clear advantages today, the technical approach should allow for seamless migration as new standards emerge or existing standards evolve. These different schemes also provide additional functionality beyond simple authentication. Some of these additional functions may be necessary in certain cases and useless in others.

The encapsulation of an end user's decision at the Portal offers the opportunity for multiple schemes to coexist gracefully. The first step for the end user at the Portal is to select the desired application, allowing the Portal to present a list of appropriate CSs. The list of CSs is based on compatible assurance levels and MD SSO schemes. If the end user selects an AA that supports SAML and Liberty Alliance, then only CSs supporting one of those schemes would be presented to the end user.

However, supporting multiple schemes introduces a problem; it may be impossible for an AA to leverage all the CSs if there are too many protocol disparities. This is a problem that will have to be monitored closely by the E-Authentication Initiative, or a principal vision of the initiative may be lost. There are several solutions available to mitigate this problem:

1. Multiple protocol support by CSPs
2. Multiple protocol support by AAs
3. E-Authentication sponsored Scheme Translators

Encouraging CSs to support multiple protocols may be a viable solution if there are not many CSs or if there is a sufficient business case to warrant the investment. A credential issued by a service that supports more than one scheme is certainly more valuable because it would be usable by more sites, but depending on the CS's model the value may not warrant the investment.

Encouraging AAs to support multiple protocols may also be viable for similar reasons. For example, if the AA uses a COTS authentication front end that natively supports multiple schemes, or if the perceived value of being able to rely on more CSs warranted the additional investment, then it would make sense to do so.

E-Authentication sponsored scheme translators are one viable solution to protocol disparities. At a high level, the role of the translator is simple; it acts as an intermediary for incompatible CSs and AAs by supporting multiple MD SSO schemes. For example, a scheme translator that supported WS-Federation and Liberty Alliance would allow the two communities to have interoperable authentication while enjoying whatever other benefits each scheme provides to each community.

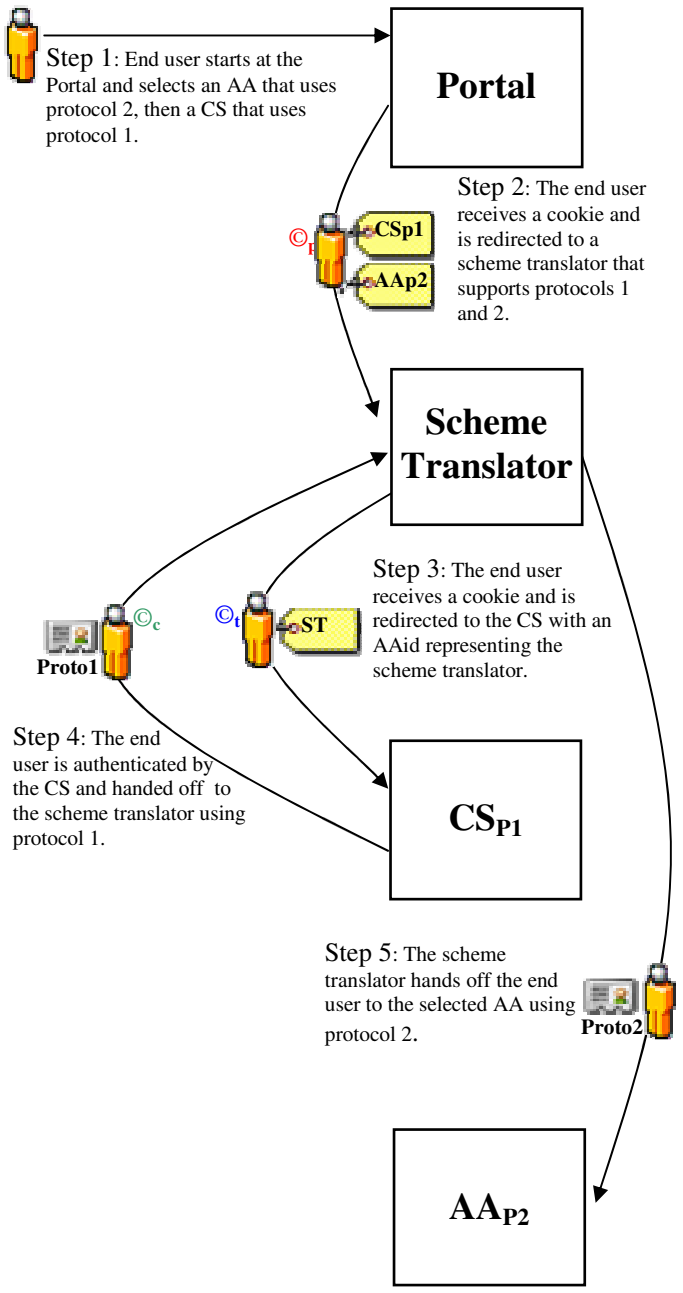
The scheme translators simply pass through identity information based on standards already adopted in the architecture. The architectural framework allows for multiple translators to be deployed allowing for an increase of availability and end user privacy. There is also no need for AAs or CSs to engage in any special integration for translators. The translators appear to be any other CS from the AA perspective, and any other AA from the CS perspective. Organizations that have invested in one of the supported architectures will be able to use their existing systems so long as the translators are available.

The following section depicts the sequence of events for an authentication that involves a scheme translator.

3.4.1 Scheme Translators

Figure 5 shows how a scheme translator fits into the architectural framework. The end user starts at the Portal to select a CS and AA, as shown in Step 1. When the end user selects the AA, the Portal provides a list of CSs that have an appropriate assurance level, have MD SSO schemes compatible with the AA, or have MD SSO schemes compatible with an appropriate scheme translator. If the CS and AA are directly compatible, the session continues as described in the Base Case. If the translator is required, the end user is redirected to the translator with the AA and CSids, as shown in Step 2. The translator then provides the end user with a cookie that contains both AA and CSids and redirects the end user to the CS with an AAid that represents the translator, as shown is Step 3. The CS performs the same functions as any other use case, authenticating and handing off the end user, as shown in Step 4. The translator now has the identity information for the end user and initiates a hand-off to the AA using the second protocol.

Figure 5: Scheme Translators



The translator cookie records the destination AAid, so the translator knows where to hand-off the end user once he is returned from the CS.

Since the translator does not interact with the end user, its role is completely transparent. The CS interacts with the translator as if it were any other AA, so no additional functionality is required by CSs to interface with translators. The AA interacts with the translator as if it were any other CS, so no additional functionality is required by the AA to interface with translators. Only the Portal configuration and the translator are required to bridge the gap among multiple schemes.

3.4.2 Scheme Adoption

Considering the architectural framework allows for multiple scheme translators to co-exist, the E-Authentication Initiative must carefully govern the introduction of new schemes. The translators add complexity to the architecture and establish an additional point of failure in transactions, so their use should be minimized. Ideally, only a small number of schemes would exist in the architectural framework at any given time, and scheme translators would be phased out over time as various components adopt dominant schemes.

Figure 6 depicts the lifecycle for adopting new schemes. As new schemes emerge that meet E-Authentication requirements, they are assessed for the availability of interoperable COTS, then piloted on a small scale. If the pilots are successful, then existing components can either migrate to support the new scheme or the initiative can deploy translators. The translators eliminate the need for every component to migrate at the same pace; slower components can rely on translators until they are ready. E-Authentication components (i.e. AAs and CSs) that have adopted new schemes can begin to use them immediately, enjoying other features they may offer without losing authentication interoperability with the rest of the E-Authentication components.

Figure 6: Scheme Adoption Lifecycle

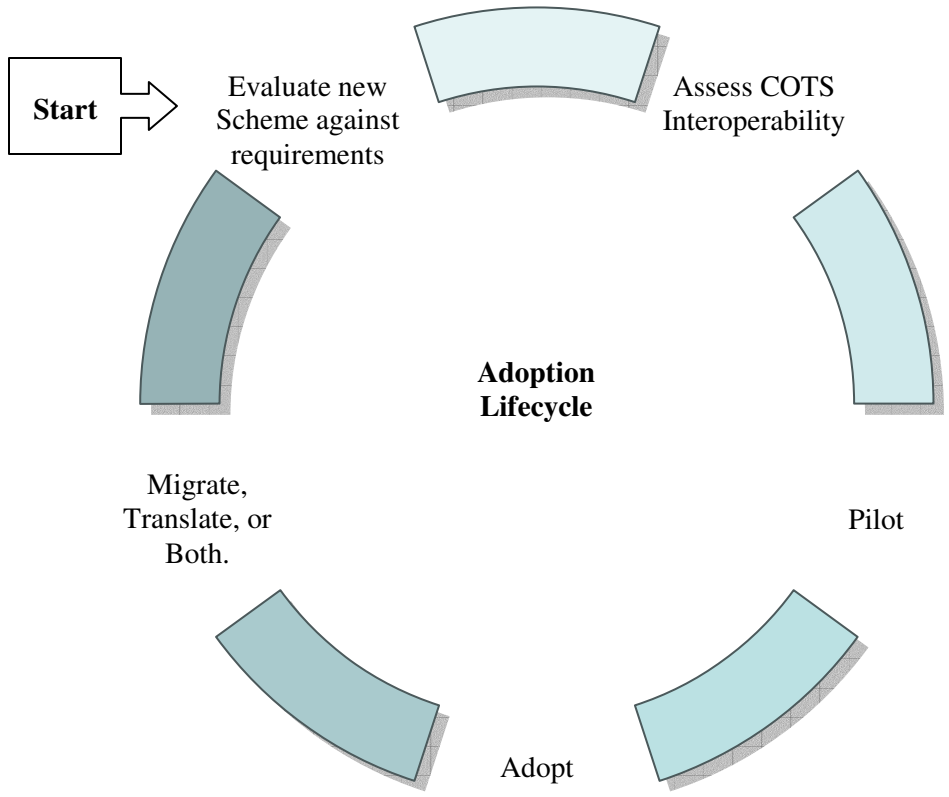
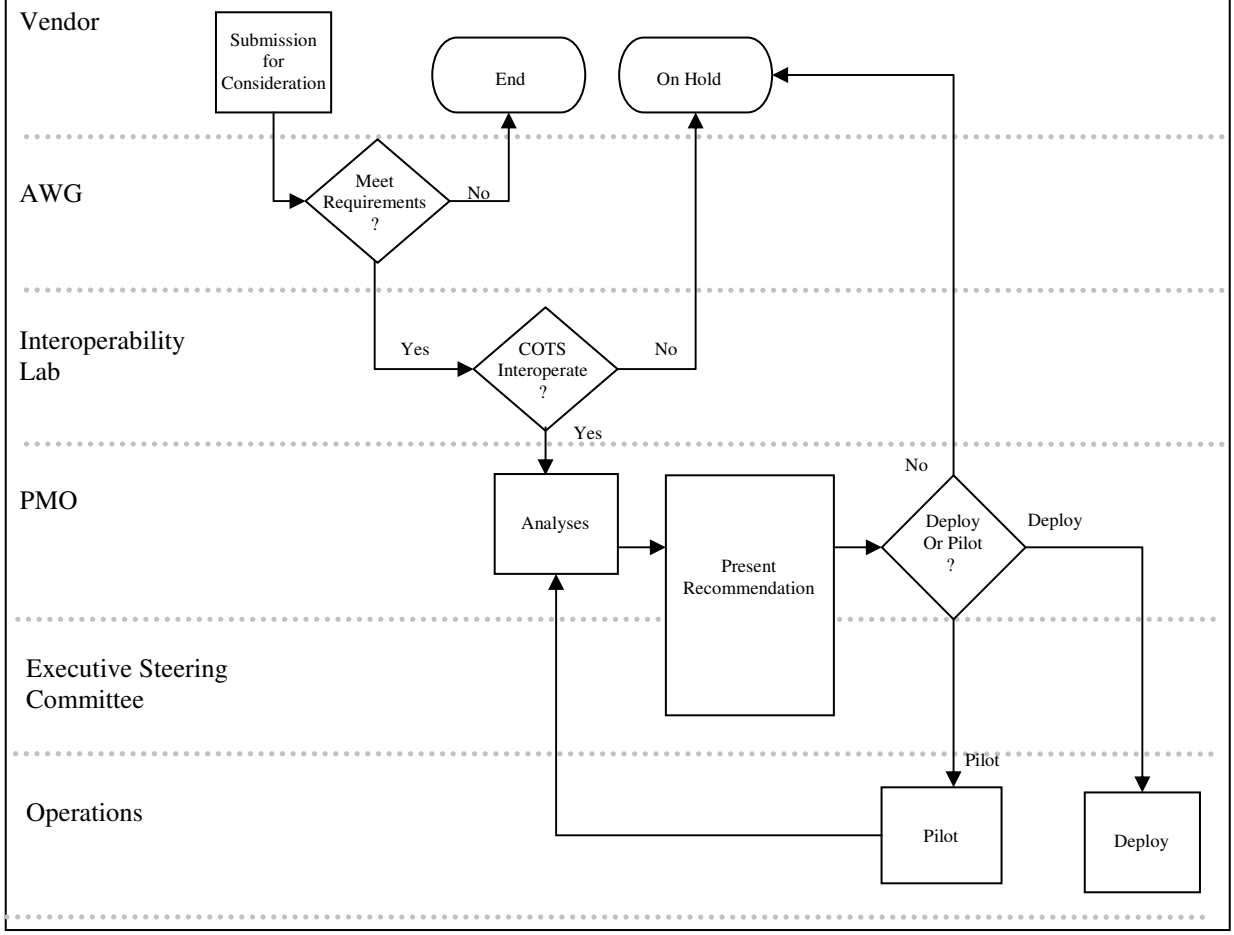


Figure 7 shows a process for adopting new schemes. New schemes are evaluated against E-Authentication requirements by the Architecture Working Group (AWG). The minimum requirements for an adopted scheme in the ASC are:

1. **Standard**: The scheme must be based on an industry standard approved by a recognized standards body. Proprietary mechanisms will not be considered. De Facto standards can be considered if they are deemed sufficiently mature by the AWG.
2. **MD SSO**: The candidate scheme must provide a mechanism for MD SSO. This is the principal function for schemes in the architecture, required for the hand-off from the CS to the AA.
3. **Service Authentication**: The scheme must provide a mechanism for services within the component to authenticate. An AA must have a way to authenticate a CS, as sanctioned by the initiative, before relying on its assertion. Likewise, a CS must have a mechanism for authenticating an AA before an assertion is provided. The specific mechanism will vary from scheme to scheme.

Next, the E-Authentication Interoperability Lab will assess the state of COTS interoperability and provide analyses to the PMO, including recommendations for using scheme translators. If sufficient interoperable COTS exist and the scheme offers sufficient benefit to the government, the Executive Steering Committee will decide whether to deploy or pilot the scheme within the architecture.

Figure 7: Scheme Adoption Process



4 Certificate-Based Authentication

PKI based credentials offer considerable advantages for authentication. They are capable of certificate-based authentication transactions and can be validated using only public information. The standards for PKI are also more mature and more widely used than the emerging standards for federated PIN/Password based electronic authentication.

The Federal PKI (FPKI) employs a Bridge Certification Authority (BCA) to harmonize policies and procedures for Certification Authorities (CAs). The E-Authentication Initiative has deferred assessment and governance of PKI based CSs to the FPKI Policy Authority (PA), the governing body for the Federal Bridge CA (FBCA). Additional information on the FPKI is available at http://www.cio.gov/fpkipa/.

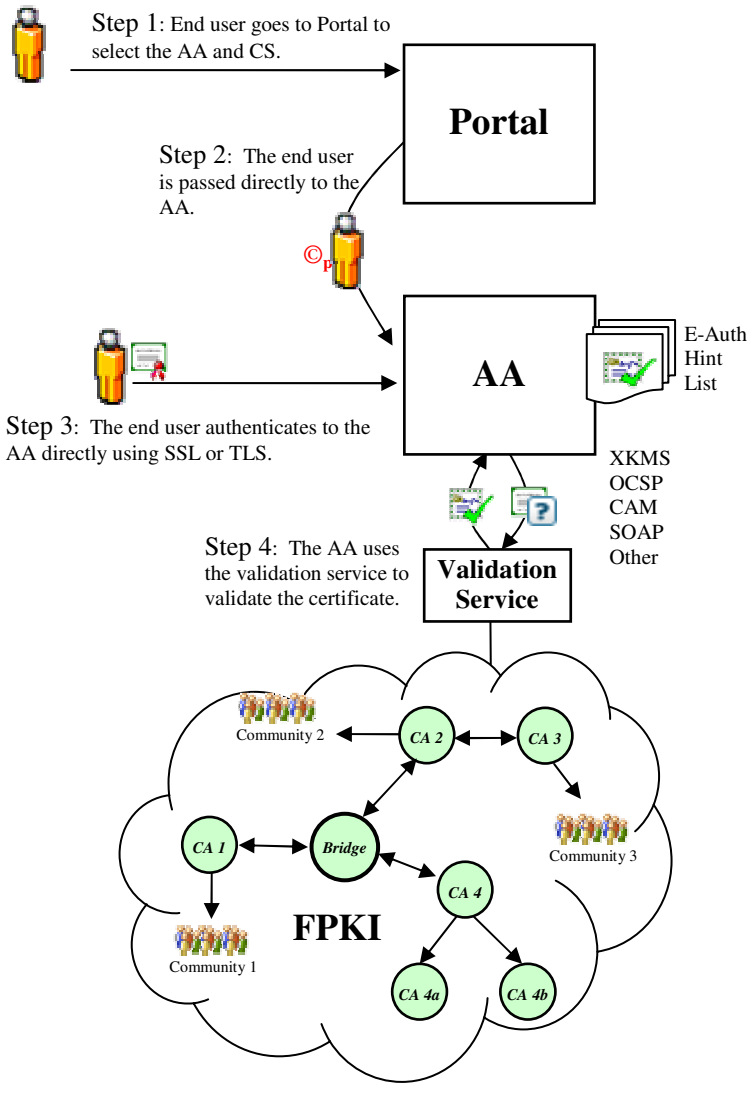
The E-Authentication technical approach for accepting PKI based credentials is based on providing mechanisms for AAs to validate certificates. The following sections describe the various use cases for certificate validation services.

4.1 Certificate Validation Service

E-Authentication will offer a certificate validation service to AAs. Figure 8 depicts the use of the certificate validation service for authentication. In Step 1, the end user starts at the Portal as usual, but is passed directly to the AA for authentication, as shown in Step 2. There is no need for the end user to be sent to the CS because TLS and Secure Socket Layer (SSL) allow the end user to authenticate using a certificate without revealing any secret information. The AA authenticates the end user in Step 3, then delegates validation of the certificate to the validation service in Step 4.

To the greatest extent possible, the validation service will be comprised of COTS products using standard protocols. NIST has established requirements for certificate path validation that can be used by the E-Authentication Interoperability Lab to determine appropriate products and interface specifications.

Figure 8: Certificate Validation Service



Over time, the validation service may support multiple products and standards, but the functionality will remain the same. Again, the approach is a an architectural framework showing where appropriate standards can be adopted as they mature.

The TLS/SSL protocol requires the web server to present a list of acceptable CAs to the browser during the TLS/SSL handshake in Step 3. The E-Authentication Initiative will publish a hint list of CAs available for use by AAs. This E-Authentication hint list is only made available to help end users select an appropriate certificate, and is not used when validating certificates[2].

[2] Additional information on the use of hint lists is available at http://www.cio.gov/eauthentication.

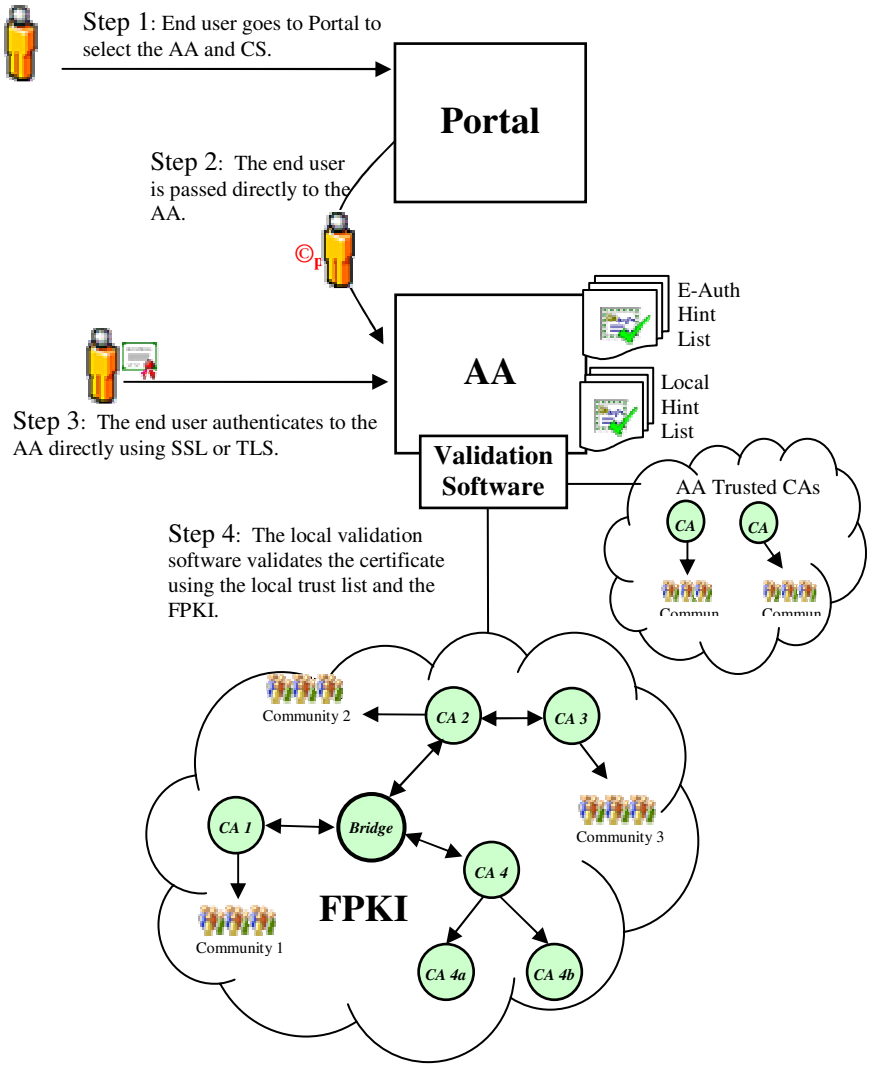
4.2 Local Validation

In some cases, agencies may wish to perform certification validation locally. For example, if an agency has elected to trust CAs that are not cross-certified with the FBCA, the agency would have to add those CAs to its local trust list.

The E-Authentication Initiative will support these agencies by performing software evaluation on products that can be run locally. The initiative will perform software evaluation based on the FPKI requirements established by NIST and provide an approved product list for agencies.

Figure 9 depicts this use case. In Step 1, the end user starts at the Portal and validates to the AA. The end user is then passed directly to the AA and authenticates to the AA using SSL or TLS, as shown in Steps 2 and 3. In Step 4, the AA then uses locally installed validation software that validates credentials using the agency's trust list. Communication with the validation service is not required.

Figure 9: Local Validation



4.3 Certificate-Based Credentials at Assertion-Based Applications

One of the requirements for E-Authentication is that credentials should be usable with any AA that has an equal or lower assurance level. That implies that PKI credentials should be usable at assertion-based authentication applications. In order to avoid the need for assertion-based authentication applications to validate certificates, the initiative will deploy a step down translator, which is a scheme translator that supports certificate validation and the dominant MD SSO schemes in use by assertion-based authentication applications.

Figure 10: Certificates at Assertion-Based AAs

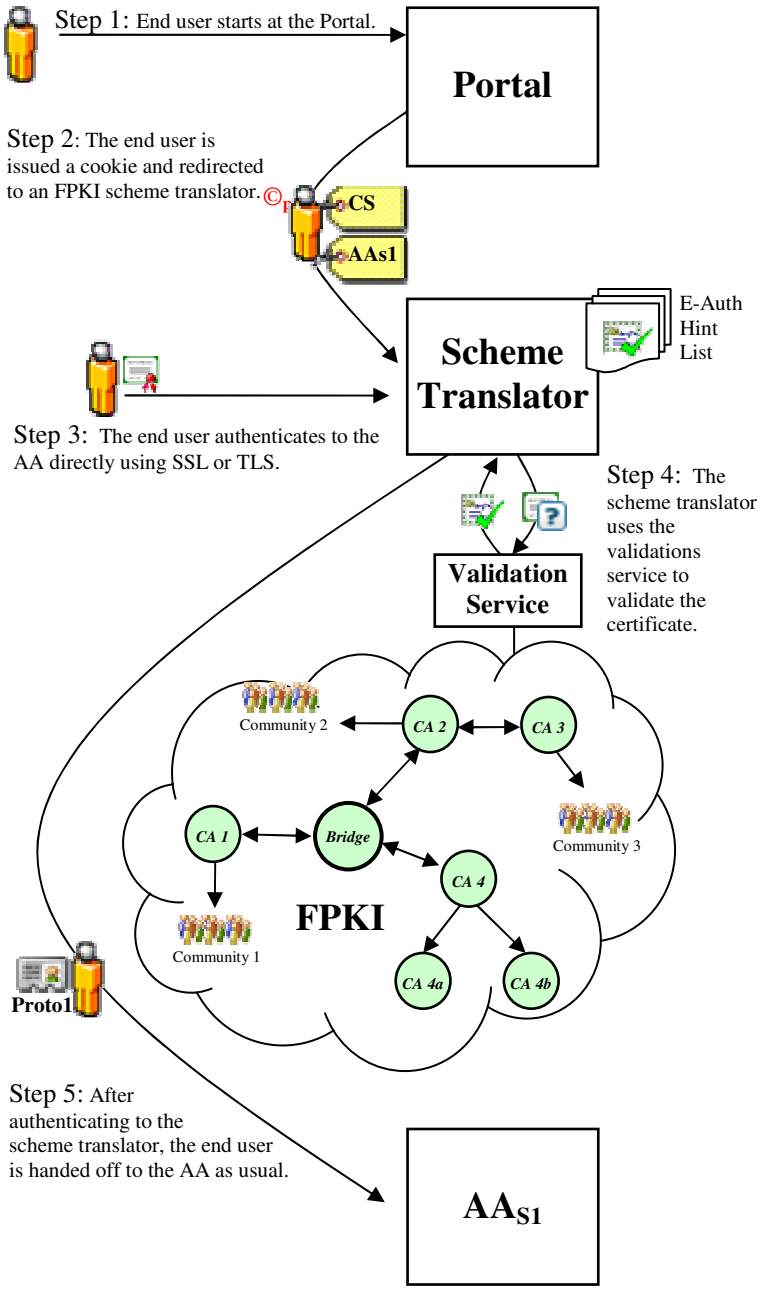


Figure 10 shows the sequence of events for an end user with a PKI credential accessing an assertion-based authentication application. In Step 1, the end user begins at the Portal and selects a CS and AA. The Portal then hands off the end user to the scheme translator with the CS and AAids as shown in Step 2. In Step 3, the end user authenticates to the translator using a certificate. Next, the translator uses the validation service to validate the certificate before handing off the end user to the AA in Step 5.

The AA does not have to deploy any special capabilities to leverage the scheme translator. The translator interacts with the AA like any other CS in the Base Case. Decisions on whether a translator is required are also encapsulated at the Portal, further insulating the AA from any special requirements.

5 Implementation

The architectural framework presented in this document does not prescribe the specific standards currently employed; therefore, this document must be accompanied by further specification of adopted schemes that depict the architecture at another point in time. Each of the adopted schemes must be further accompanied by interface specifications that provide detailed technical specifications for how to use a given scheme within the architectural framework. Current information will be maintained by the E-Authentication Initiative and made available at their website, http://www.cio.gov/eauthentication/.

AAs and CSs joining the E-Authentication community must select one (or more) of the adopted schemes to interoperate with other components in the architecture. The initiative will provide a list of tested COTS products within each scheme that have proven interoperability according to federal standards and specifications. Additional agreements beyond the scope of this document are also required. Interested parties should contact the PMO for more information.

Appendix A: Distributed Portal Functionality

1 Introduction

The AA and CS metadata stored at the Portal could be shared with other entities. There is nothing sensitive about the information and no reason to keep it isolated at the Portal. Other sites equipped with the information could assist end users during the selection of an AA or CS. The Portal's ability to process passed AA and CSids enables other sites to add value without requiring redundant interaction with end users.

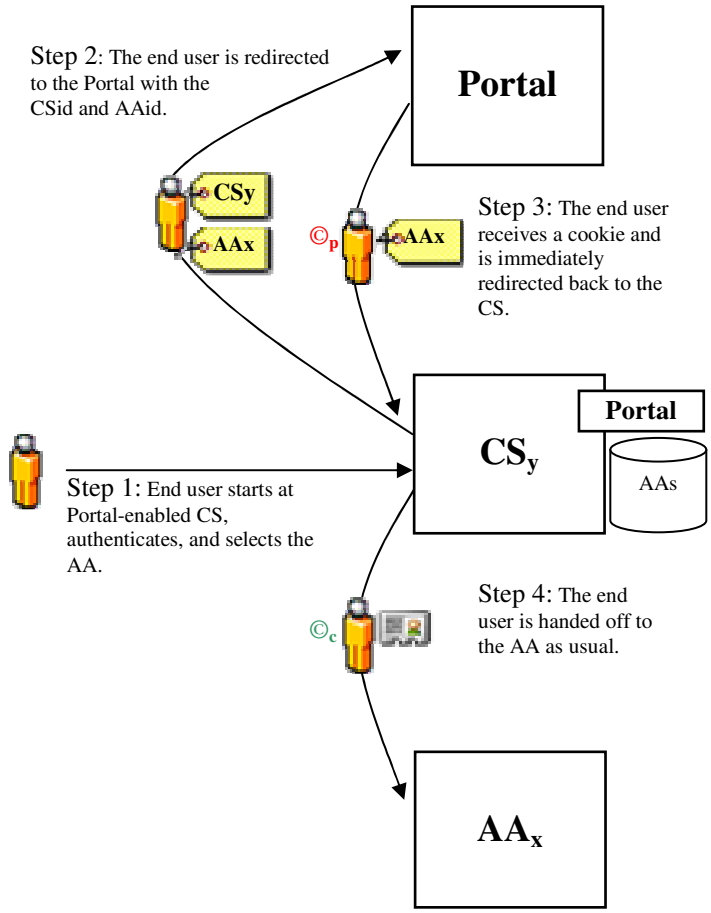
One example is for a CS to present the end user with AAs that will accept their credentials. CSs, such as banks, may be able to add value by suggesting AAs that are relevant to a particular end user or related to the business the end user is engaged in during a particular browser session. A CS that has downloaded the metadata about AAs is considered Portal-enabled if it has the ability to present the end user with applications that are accessible with their credential and can redirect them through the Portal to the application. The following sections describe use cases where other sites have been Portal-enabled.

2 Portal Functions at the Credential Service

It is possible for a CS to provide some Portal functions in this architectural framework. Figure 11 shows the sequence of events for this case. In Step 1, the end user starts at the CS, perhaps conducting routine business. The CS has integrated metadata on AAs into its site and presents the end user with a list of applications that can be accessed with their credential. When the end user selects one of the applications, the CS redirects the end user to the Portal with the AAid and the CSid, which is shown in Step 2. Since the end user arrives at the Portal with both AA and CSids there is no need for the Portal to interact with the end user. The end user simply receives a transaction cookie and is passed back to the CS, as shown in Step 3. In Step 4, the CS has already authenticated the end user, so it can immediately pass the end user to the AA as described in the Base Case.

While it would be possible for the CS to initiate the hand-off to the AA directly, the end user must be sent to the Portal in order for single sign-on to work properly. If the CS passed the end user directly to the AA, then subsequent visits to other applications would not be automatically authenticated. Single sign-on requires the Portal cookie as well as the CS cookie, so the end user must be passed through the Portal even when not interacting with it.

Figure 11: User Starts at Portal-enabled CS



Explicit support for this scenario in the architecture encourages CSPs to advertise the availability of government applications. It also provides an easy mechanism for CSs to show the value of their credential to their end user base. The end user benefits from easier availability and access to government applications.

3 Portal Functions at the Agency Application

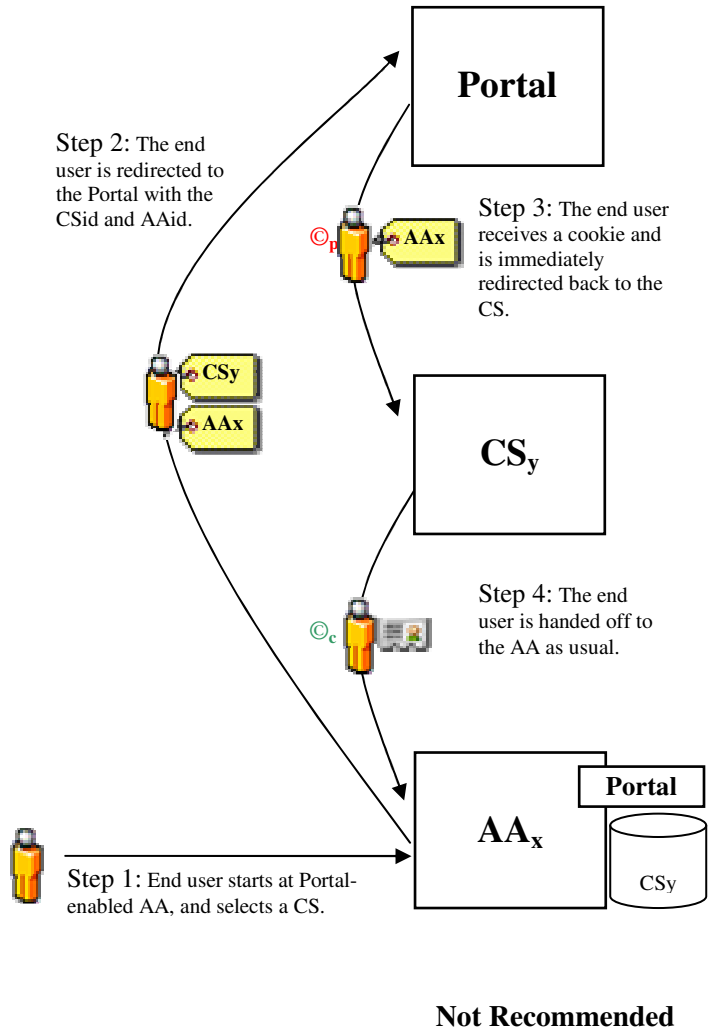
It is also possible for an AA to provide some Portal functionality. If an AA downloads the metadata for CSs, it could provide end users with a list of potential CSPs.

Figure 12 shows the sequence of events for the Portal-enabled AA case. During Step 1, when the end user starts at the AA, which has integrated the metadata for CSs, the application can then present the end user with a list of appropriate CSs directly from the AA site. After selecting the CS, the end user is redirected to the Portal with the AA and CSids, as shown in Step 2. Once again, because there is no interaction with the Portal, the end user simply receives a cookie and is passed along to the CS, as shown in Step 3. Finally, the end user is then authenticated and passed back to the AA, as described in Step 4.

This scenario is not recommended because it can interfere with single sign-on. If the end user had already authenticated to a different AA earlier in the browser session and then accessed the Portal-enabled application, the end user would have to select the CS a second time at the Portal-enabled AA. If the AA simply redirected the end user to the Portal as described in figure 2, the end user would not be required to make the selection a second time. This scenario is presented because it may provide utility to some agencies in certain circumstances and requires no additional functionality in other architectural components.

If this were the end user's first authentication, then subsequent access to other AAs would provide single sign-on.

Figure 12: User Starts at Portal-enabled AA

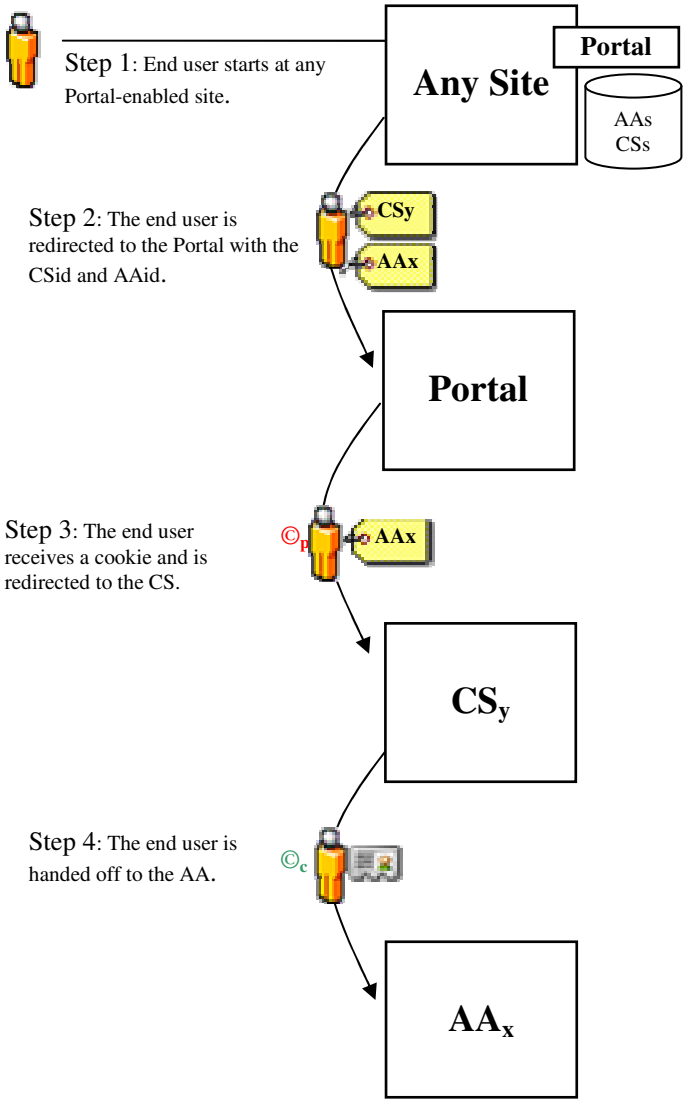


4 Other Possibilities

The ability of the Portal to accept incoming AA and CSids supports a variety of scenarios that allow for flexibility in the end user experience. For example, it would also be possible for a commercial Portal to download all of the metadata and provide E-Authentication Portal functionality, simply redirecting the end user to the E-Authentication Portal once the end user selected a CS and AA. An industry association could also integrate the metadata into its website offering members easy access to industry-related applications, again redirecting the end user to the E-Authentication Portal once the end user selected a CS and AA. Agency websites could offer similar functionality, highlighting the applications provided by the agency. These scenarios, and others, are possible and ultimately benefit the end user and the government by increasing the exposure of E-Government applications.

Figure 13 depicts the sequence of events for these scenarios. In Step 1, the end user starts at any site that has integrated the Portal metadata and selects a CS and AA. In Step 2, the end user is redirected to the Portal with the CS and AAids as described in the previous cases. The Portal can then immediately redirect the end user to the CS without any interaction, as shown in Step 3. The sequence then continues as described in the Base Case; where the end user authenticates to the CS and then is passed to the AA in Step 4.

Figure 13: User Starts at another site



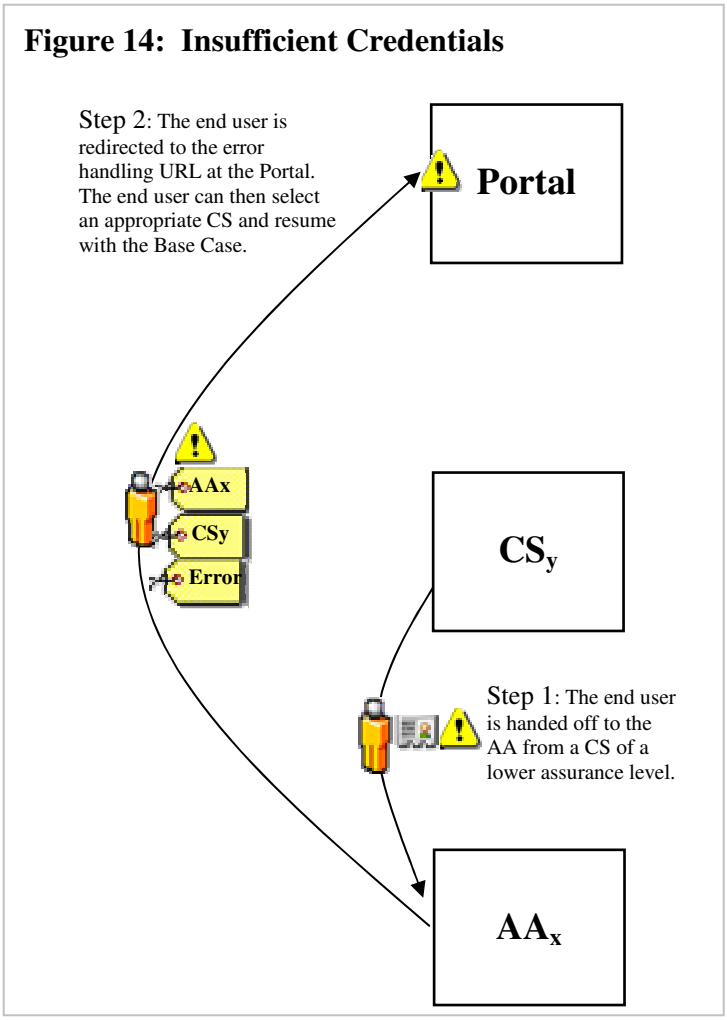
Appendix B: Exception Handling

Any good design should have multiple layers of risk mitigation. The hand-off from the CS to the AA is a crucial part of the ASC; therefore multiple approaches have been employed to ensure smooth hand-offs. The use of standards, the testing of products, the role of the Portal, and the governing authority are all designed to ensure successful hand-offs. This section describes a final fail-safe mechanism that provides a consistent and helpful end user experience during hand-off exceptions.

The Portal has a Uniform Resource Locator (URL) that any CS or AA can redirect an end user to in the event of any problems. The URL supports a number of standardized error codes that can be included on the query string. This approach helps to minimize the usability and exception handling burden on CSs and AAs, and ensures consistent exception processing throughout the architecture.

Figure 14 shows the sequence of events for a hand-off failure from a CS to an AA. Step 1 shows an example exception, where a CS attempts to make an assertion to a higher assurance level AA. Such an exception would require improper configuration at both the Portal and the CS, and would fail due to security mechanisms used in the hand-off[3]. When the AA detects the problem, the end user is redirected to an error handling URL at the Portal with the AAid, CSid, and error code on the query string, as shown in Step 2. The Portal may then provide the end user with a page explaining the event and allow for selection of an appropriate CS for the AA, or to select an appropriate AA for the CS. In addition, the Portal will log the event for review by administrators. If the CS-AA combination is valid according to the Portal database, then the combination may be disabled for further sessions until administrators can fully debug the problem.

Figure 14: Insufficient Credentials



Supported error codes are provided in the scheme interface specifications and may expand over time. As experience with end users increases, the design and content of the Portal can be updated and improved without any modifications elsewhere in the architecture.

[3] Details on scheme security mechanisms are provided in the adopted scheme document and the interface specifications for each particular scheme.

Appendix C: Secure Email

Secure email is supported through the ASC using the same certificate validation techniques described in section 4. Any Secure/Multipurpose Internet Mail Extensions (S/MIME) capable e-mail software product can be used to process signed and encrypted email.

The following sections describe three options for validating the status of the certificate attached to the message once the message has been processed. Each organization will have to determine which approach is appropriate for their needs.

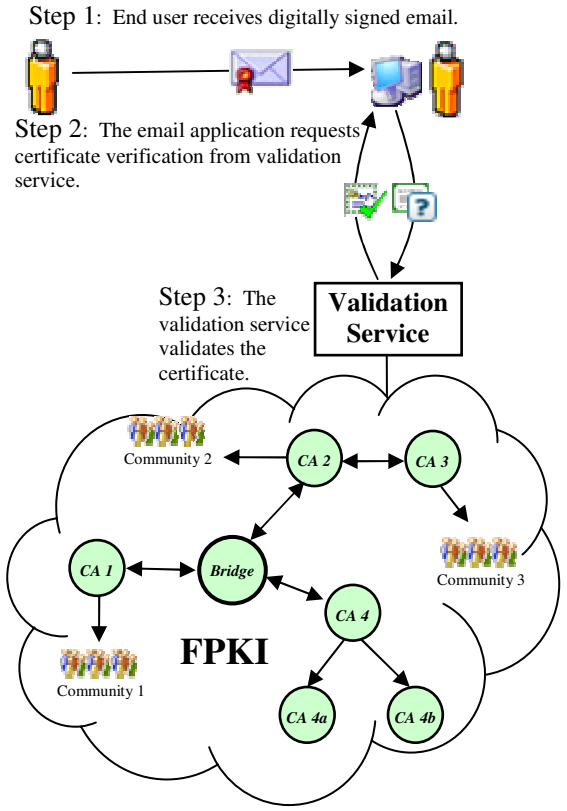
1 E-Authentication Validation Service

Figure 15 shows the use of the E-Authentication validation service[4] for certificate status checking. In Step 1, the end user receives digitally signed email. In Step 2, the email software on the end user's computer requests certificate verification from the validation service to determine the certificate's status. This is the same validation service depicted in section 4, which determines the certificate status through the bridge[5]. In Step 3, the validation service validates the certificate.

In this approach, the end user's e-mail software must be capable of processing S/MIME, and be capable of using a certificate status protocol supported by the validation service. As described in section 4, these protocols may vary over time; examples include XKMS, OCSP, and SCVP.

This option is not appropriate for relying parties that trust CAs that are not cross-certified with the bridge.

Figure 15: Validation Service

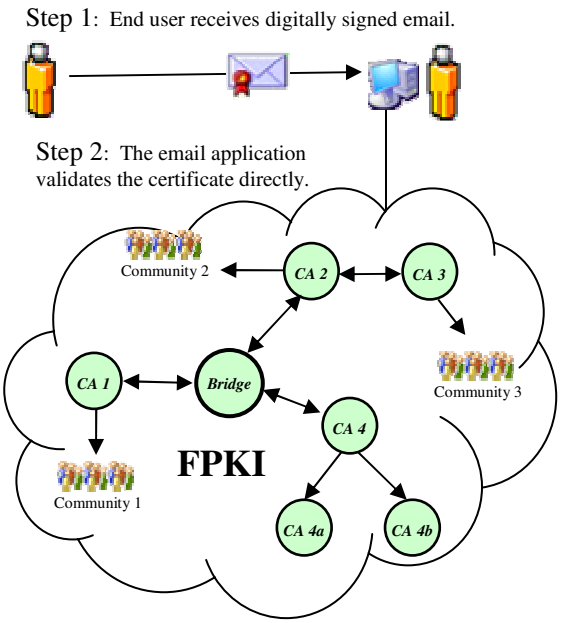


[4] Validation Service software is sometimes identified as Certification Status Authority (CSA).
[5] The bridge is referred to as the Bridge Certification Authority (BCA).

2 Desktop Validation

Another possibility is to run certificate validation software on the end user's desktop. The capability could be part of the email software, or be a software addition. Figure 16 depicts this use case. In Step 1, the end user receives a digitally signed email. In Step 2, the end user's desktop then validates the certificate through the bridge individually.

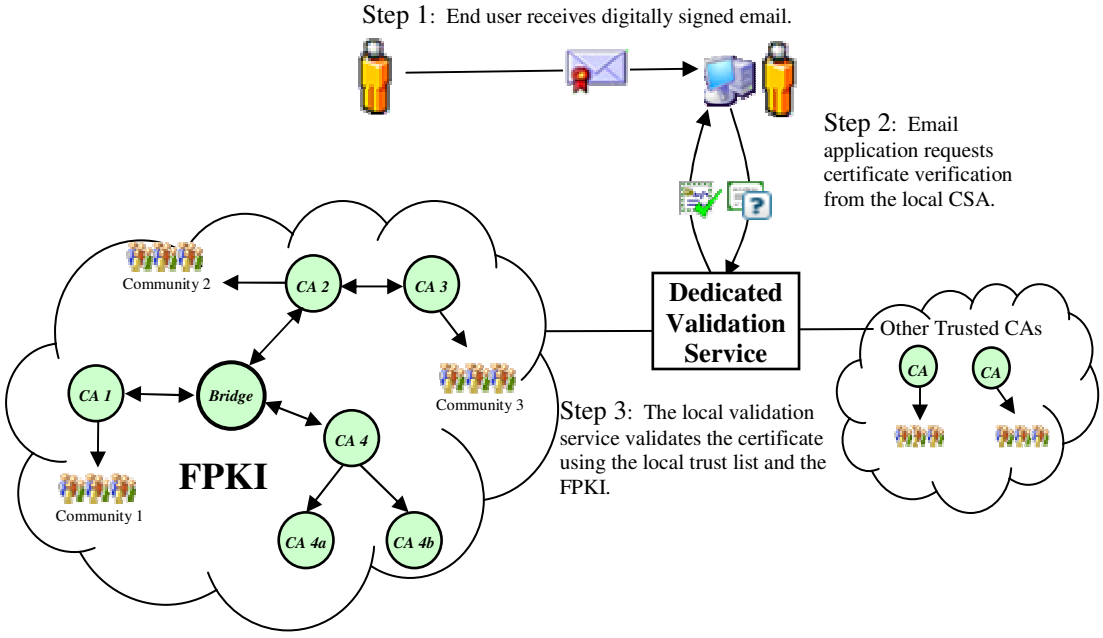
Figure 16: Desktop Validation



3 Dedicated Validation Service

For organizations who trust CAs that are not trusted government-wide, a dedicated validation service may be a good choice. The local validation service may be configured to trust other CAs as appropriate for that organization. Email software used in that organization would then be configured to rely on the dedicated validation service. The validation service must still be capable of interoperating within the bridge to ensure other E-Authentication credentials can be accepted. Figure 17 depicts this use case. In Step 1, the end user receives a digitally signed email. The email software then requests certificate verification from the local validation service, as shown in Step 2. In Step 3, the local validation service validates the certificate using the local trust list and FPKI.

Figure 17: Dedicated Validation Service



4 Other Approaches

The previously described options can be combined or varied to create other options. For example, a dedicated validation service could use the ASC validation service for any certificate it could not validate on its own, or a desktop validation engine could be configured with locally trusted CAs.

The key point is to ensure that e-mails are signed and encrypted using S/MIME, and that certificates comply with X.509 version 3. If those two standards are followed, then a variety of certificate validation approaches may be employed.

Appendix D: Electronic Forms Applications

1 Introduction

Some E-Government business is being performed with electronic forms applications rather than through on-line web forms. These applications do not have the same characteristics as browser-based applications. This appendix shows an approach for using the ASC with forms applications.

Many forms applications support the use of PKI certificates to sign the form. The form server component can then validate the certificate (and thus the signature) using one of the certificate validation approaches described earlier in this document. This approach is straightforward and widely supported by forms software, but requires end users to have certificates. Section 2 below describes certificate-based submission of forms.

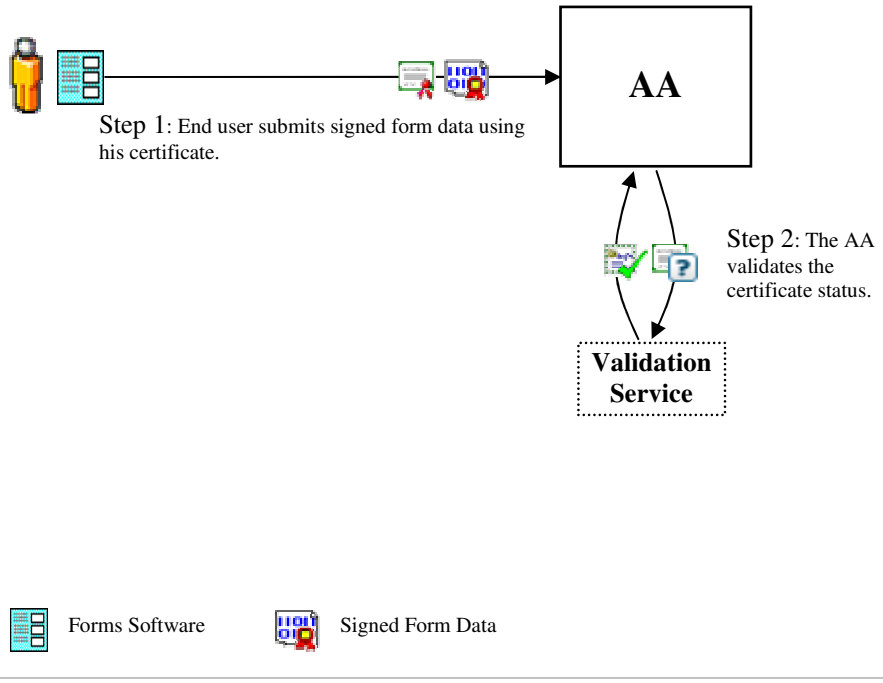
The use of PINs or Passwords to submit forms in a federated environment is more complex due to NIST restrictions against sharing secrets. The forms software cannot submit third party passwords to AAs directly. The approach described here uses the existing architecture by authenticating through a browser window, which requires no special interface for CSs, and maintains the scalability and privacy features of the ASC. This is the recommended approach for pin/password authentication of form data submissions using the ASC. As standards evolve, the ASC may be updated to accommodate additional approaches. In this approach, the browser window is not required until the form data is submitted. End users are still free to download forms without authenticating and may complete the forms offline. When the end user submits the form data, the form application opens a browser window and redirects the end user to the Portal, which in turn redirects the end user to their CS for authentication. Section 3 below describes this approach in more detail.

2 Certificate-Based Form Submission

This section describes how electronic forms work with certificate-based authentication. A significant number of electronic forms products support digital signatures using certificates, so this is the most straightforward way to submit signed forms.

Figure 18 describes the sequence of events for this case. Using the form software, the end user signs the form data, and then submits it to the AA along with the certificate, as shown in Step 1. After verifying the signature, the AA uses a validation service to validate the certificate, as shown in Step 2. A number of approaches can be used to validate the certificate, see the previous appendix or section 4 for more information.

Figure 18: Certificate-Based Form Submission

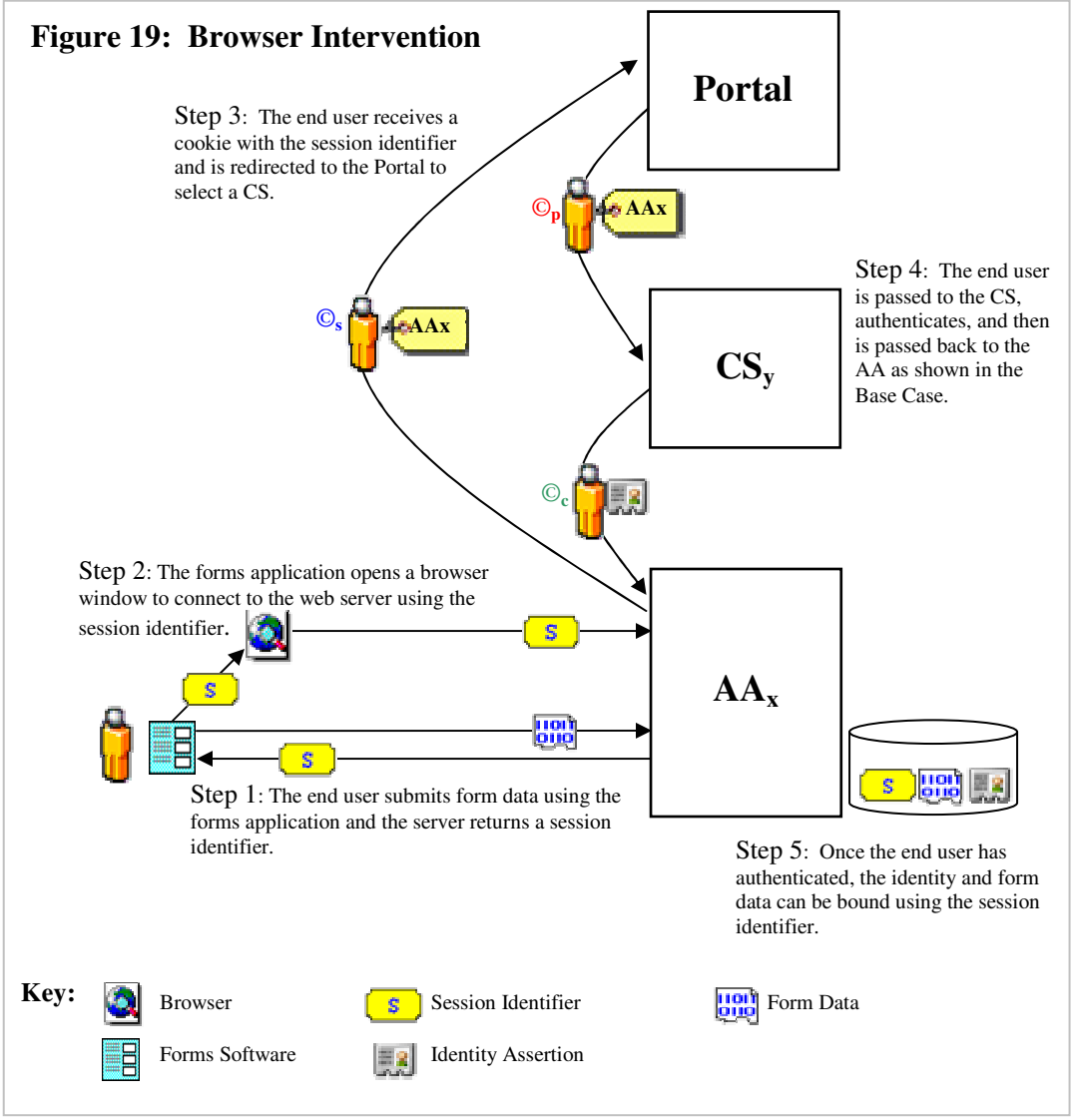


3 Browser Intervention

Figure 19 shows the flow of events for authenticating a form submission not using digital certificates. When the end user submits the form data using a forms application, the server returns a session identifier, as shown in Step 1. The forms application then opens a browser window with the AA's URL and the session identifier in the query string, as shown in Step 2. In Step 3, the end user receives a cookie with a session identifier from the AA and is redirected to the Portal with the AAid on the query string. The end user then selects a CS and is authenticated, as described in section 3 (Assertion-Based Authentication) and shown in Step 4. When the end user is handed off from the CS, the AA reads the session identifier from its cookie, allowing the application to bind the asserted identity to the form submission, as shown in Step 5.

Allowing a browser window to intervene at the time of authentication allows forms applications to leverage all E-Authentication CSs and minimizes the need to customize forms applications.

Figure 19: Browser Intervention



Appendix E: Glossary and Acronyms

Term	Description
Agency Application (AA)	An online service provided by a government agency that requires an end user to be authenticated.
Agency Session	The period of time the AA will trust an end user before handing him or her off to the CS for re-authentication. AAs do not have access to authentication session information; they must maintain their own session with an end user and decide how long an end user remains authenticated once the transaction has started.
Assurance Level	Level of trust, as defined by the OMB M-04-04.
Authentication Session	The period of time that an end user remains trusted by a CS (CS) after the end user authenticates. A CS typically does not require an end user to re-authenticate for every page requested; the end user continues to be trusted for some period of time after each authentication. The allowed period between re-authentication is referred to as the authentication session.
Authorization	An authenticated end user's right to perform transactions or access data of an application. AAs maintain full control over authorization.
Browser Session	The period of time the end user's browser is open. The browser session begins when the end user opens their browser and ends when it is closed. All session cookies are terminated when the Browser session ends.
Certificate	X.509v3 digital certificates in a Public Key Infrastructure (PKI) for authentication can be used at any assurance level.
Certificate Arbitrator Module (CAM)	Application-level router that routes certificates from relying party programs to the issuing CAs for validation. By interfacing directly with the CAM, a relying party application will be able to interact with multiple CAs.
Certification Authority (CA)	A certification authority 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.
Claimant	A party whose identity is to be verified using an authentication protocol.
Composite Application	An application that relies on remote services to complete its transactions.
Credential	Digital documents used in authentication and access control that bind an identity or an attribute to a claimant's token or some other property, such as an end user's current network address. Note that this guidance distinguishes between credentials and tokens, while other documents may lump tokens with credentials.

Term	Description
Credential Assessment Framework (CAF)	Based on technical and policy guidance from Office of Management and Budget (OMB) and National Institute of Standards and Technology (NIST), the CAF provides a structured means of delivering assurances to Federal agencies as to the veracity, and thus dependability of identity credentials and tokens. This assurance is achieved by evaluating and assessing CSPs and their credential-issuing service(s) against criteria established in the CAF.
Credential Service (CS)	A service of a CSP that provides credentials to subscribers for use in electronic transactions. If a CSP offers more than one type of credential, then each one is considered a separate CS.
Credential Service Provider (CSP)	An organization that offers one or more CSs. Sometimes known as an Electronic Credential Provider (ECP).
E-Authentication Portal (Portal)	A website that helps end users locate the CSs and AAs they need to complete their transactions. The Portal also maintains information about CSs and AAs referred to as metadata, which includes technical interface data as well as descriptive information. When the end user opts into single sign-on, the Portal assigns a session cookie.
End Users	Any citizen, government employee, contractor, or business that uses an AA. One of the principal goals of E-Authentication is to make the end user experience as simple as possible by improving the availability and ease of use of credentials.
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.
Federal Bridge Certification Authority (FBCA)	Allows PKIs to trust digital certificates issued by other entities that have been policy mapped and cross-certified with the FBCA. See http://www.cio.gov/fpkipa/.
Federal Enterprise Architecture (FEA)	Component-based architecture that facilitates expansion of E-Government by identifying opportunities to collaborate, consolidate, and leverage IT investments across government. The architecture includes several reference models, including Performance (PRM), Business (BRM), Service Component (SRM) and Technical (TRM).
Federal Public Key Infrastructure (FPKI)	Employs a BCA to harmonize policies and procedures for CAs. See http://www.cio.gov/fpkipa/.
Governing Authority	Established by the government to issue certificates that allow Agency Applications to retrieve SAML assertions from Credential Services over a client and server authenticated SSL channel, effectively controlling which entities can participate.
Hint List	A list of CAs sent to browsers during the TLS/SSL handshake. The browser uses the list to help the end user select the certificate to use for authentication. The E-Authentication Hint List consists of the names of every CA that is reachable from the FBCA.
HyperText Transfer Protocol	Underlying protocol used by the World Wide Web. HTTP

Term	Description
(HTTP)	defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
Liberty Alliance Alliance	See http://www.projectLiberty Alliance.org/specs/
Online Certificate Status Protocol (OCSP)	An on-line protocol used to determine the status of a public key certificate. See [RFC 2560].
Personal Identification Number (PIN)	A password consisting only of decimal digits.
Project Management Office (PMO)	The PMO is the organization that handles E-Authentication program management, administration, and operations.
Relying Party	An entity that relies upon the subscriber's credentials, typically to process a transaction or grant access to information or a system.
Scheme	Schemes, such as SAML and Liberty Alliance, specify protocols and standards for federated identity mechanisms for different entities to share identities without requiring the end user to manage multiple accounts.
Scheme Translator	Supports interoperability among different authentication schemes by translating between CSs and AAs using different schemes. A scheme translator may be called a Step Down Translator when used to translate from certificate schemes to assertion schemes.
Secure Sockets Layer (SSL) (See also: Transport Layer Security)	Protocol for transmitting private documents via the Internet by using a private key to encrypt data that's transferred over the SSL connection.
Secure/Multipurpose Internet Mail Extensions (S/MIME)	A standard that extends the MIME to support the signing and encryption of e-mail transmitted across the Internet.
Security Assertion Markup Language (SAML)	XML-based framework for ensuring that transmitted communications are secure. SAML defines mechanisms to exchange authentication, authorization and nonrepudiation information, allowing single sign-on capabilities for Web services.
Session Cookie	Small transient file that contains information about an end user that disappears when the end user's browser is closed. Unlike a persistent cookie, a transient cookie is not stored on an end user's hard drive, but is only stored in temporary memory that is erased when the browser is closed.
Shibboleth	See http://shibboleth.internet2.edu/
Simple Certificate Validation Protocol (SCVP)	Allows a client to offload certificate handling to a server. The server can provide the client with a variety of valuable information about the certificate, such as whether the certificate is valid, a certification path to a trust anchor, and revocation status. SCVP has many purposes, including simplifying client implementations and allowing companies to centralize trust and policy management.
Simple Object Access Protocol (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

Term	Description
	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.
Single Sign-on	After initial authentication with a CS during a browser session, the end user is seamlessly logged into any other AA of equal or lower authentication levels. For privacy considerations, the end user is required to take an explicit action to opt into single sign-on.
Token	Something that the claimant possesses or knows (typically a key or password) that can be used to remotely authenticate the claimant's identity. Technically, the token includes an end user id and password that ensures token uniqueness within a credential domain.
Transport Layer Security (TLS)	An authentication and security protocol implemented in current browsers and web servers. TLS is defined by [RFC 2246] and [RFC 3546]. TLS is similar to the older Secure Socket Layer (SSL) protocol and is effectively SSL version 3.1.
WS-Federation	See: http://www106.ibm.com/developerworks/webservices/library/ws-fed/
XML Key Management Specification (XKMS)	Defines a Web services interface to a PKI. This makes it easy for applications to interface with key-related services, like registration and revocation, and location and validation.

Acronym	Abbreviation For
AA	Agency Application
AAid	Agency Application Identifier
ASC	Authentication Service Component
Authz	Authorization
AWG	Architecture Working Group
BCA	Bridge Certification Authority
BRM	Business Reference Model
CA	Certification Authority
CAF	Credential Assessment Framework
CAM	Certificate Arbitrator Module
COTS	Commercial off the Shelf
CS	Credential Service
CSA	Certification State Authority
CSid	Credential Service Identifier
CSP	Credential Service Provider
E-RA	Electronic Risk & Requirements Analysis
FEA	Federal Enterprise Architecture
FOC	Full Operational Capability
FPKI	Federal Public Key Infrastructure
FPKI PA	Federal Public Key Infrastructure Policy Authority
HTML	HyperText Markup Language

Acronym	Abbreviation For
HTTP	Hyper Text Transfer Protocol
MD SSO	Multi-Domain Single Sign-On
NIST	National Institute for Standards and Technology
OCSP	Online Certificate Status Protocol
OMB	Office of Management and Budget
PII	Personally Identifiable Information
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PMO	Project Management Office
PRM	Performance Reference Model
RA	Registration Authority
RC	Release Candidate
S/MIME	Secure/Multipurpose Internet Mail Extensions
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SCVP	Simple Certificate Validation Protocol
SP	Special Publication
SRM	Service Component Model
SSL	Secure Sockets Layer
SSO	Single Sign-on
TBD	To Be Determined
TLS	Transport Layer Security
TRM	Technical Reference Model
URL	Uniform Resource Locator
XKMS	XML Key Management Specification
XML	Extensible Markup Language

Appendix F: Document History

Document History

Status	Release	Date	Comment	Audience
Draft	0.0.1	11/03/03	Initial Draft, limited release, do not distribute	Limited
RC1	1.0.0	05/17/04	1) Added text describing that this is a technical document and provided link to web site for additional information. (AWG Bullet 10) 2) Definition and Acronym section added (AWG Bullet 12 & 64)) 3) Added text on browser capability requirements (AWG Bullet 14) 4) Added text mentioning human interaction via web browsers. (AWG Bullet 37) 5) Added text describing how E-Authentication corresponds with FEA. (AWG Bullet 44) 6) Composite Application was defined and added to the glossary. (AWG Bullet 45) 7) Added text describing the adoption of new schemes being based upon support of E-Auth use cases. (AWG Bullet 48) 8) Added text describing the consideration of other scenarios besides human/ browser to system. (AWG Bullet 51) 9) Added text describing how SAML assertions can qualify at level 3, but not at level 4. (AWG Bullet 53 & 54) 10) Added text reflecting the changes to the NIST Guidance. (AWG Bullet 55) 11) Added sentence describing “explicit opt-in”. (AWG Bullet 57) 12) Added a diagram for AA requiring a higher assurance level than is presented. (AWG Bullet 58) 13) Added text describing how logout works. (AWG Bullet 59) 14) Added text describing the difference between E-AuthN and E-AuthZ. (AWG Bullet 63) 15) Added text describing that when an AA has two interfaces (admin/user) it is defined as two AAs. (AWG Bullet 65) 16) Hint list metaphor added to replace trust list metaphor. (AWG Bullet 66).	Limited

			17) Spelling of E-Authentication changed to E-Authentication. 18) Added a document Editor section. 19) Changed Protocol Translator to Scheme Translator (AWG Bullet 84) 20) Added Glossary section to describe document terms. (AWG Bullet 12 & 64)	
RC2	1.0.0	05/28/04	1) ESP was deleted and replaced with CS. 2) Definition of End User was reworded. 3) Section 3.5 changed to section 3.4.2 and titled as Scheme Adoption. 4) Portal Translator (PT) deleted and changed to Scheme Translator (ST). 5) EVS acronym deleted and replaced with VS. 6) A footnote was added for validation service to clarify that it is sometimes referred to as a Certification Status Authority (CSA). 7) Second paragraph in section 1.3 was reworded to specify the difference in alignment of humans and browser authentication between the Tech Approach and SP 800-63. 8) In section 1.4, Single Sign-on high level requirements were reworded for better clarity. 9) http://www.cio.gov/E-Authentication changed to http://www.cio.gov/eauthentication. 10) Added document reference callouts to the documents diagram in section 1.1. 11) Updated appendices b and c based on AWG input. Insufficient credentials appendix was updated to take a more generic tact on exception handling. This resulted in the removal of an appendix from the SAML document and an update of the interface specifications as well. 12) The signed form appendix now more clearly distinguishes between certificate based and assertion based approaches. 13) Added a footnote to section 4.2 referring the end user to the website for more information on hint lists. This page now needs some formatting help. 14) Changed the hint list definition in the glossary. 15) Added an acronym list to Appendix E.	Limited
RC3	1.0.0	06/28/04	1) Text formatting changed for consistency with	Limited

Interface Specs and Adopted Scheme documents.

2) Browser session specified as session type on section 3.2.

3) Hint list definition was reworded.

4) Bridge Certificate Authority changed to Bridge Certification Authority (BCA).

5) Certificate Authority changed to Certification Authority (CA).

6) Proper document titles (NIST SP 800-63, OMB M-04-04) added to the document diagram.

7) Definition link added for Liberty Alliance Alliance, Shibboleth, and WS-Federation in the glossary.

8) Certification Authority definition added to the glossary.

9) E-Authentication Service Component changed to Authentication Service Component (ASC).