Government of Canada Credential Federation Application Integration Guide

Identity, Authentication & Authorization Services
Shared Services Canada

Version 1.3
EDRM# 1391007

Shared Services Canada
Services partagés Canada
Canada

Revision History

Date	Version	Description	Author
2011-06-14	0.1	Internal Draft	Doug Harris
2011-06-23	1.0	Incorporated Feedback	Doug Harris
2011-08-30	1.1	Minor Edits, SSC re-brand	Doug Harris
2011-09-09	1.2	Removed reference to ICM SSL certificates	Doug Harris
2012-02-14	1.3	Reference and align to Enrolment Design Guide	Doug Harris

Table of Contents

Table of Figures

1. Introduction

1.1 Purpose

The purpose of this document is to help Government of Canada Departments and Agencies to successfully integrate their program's on-line service applications with the shared credential and authentication services available within the Government of Canada Credential Federation (GCCF).

1.2 Scope

This guide includes an overview of the GCCF and its services, explains how federated authentication works, provides information to help departments make informed architecture and technology choices, and outlines all of the major steps required to successfully integrate an on-line service application into the federation.

1.3 Audience

This document is primarily targeted toward application architects, developers and testers who will be responsible for integrating an on-line service application into the Federation. It may also be of value to other technical stakeholders such a security, network and infrastructure architects.

1.4 References

[CATS]	Cyber Authentication Interface Architecture and Specification Version 2.0: Deployment Requirements
[GCCF-EDG]	Government of Canada Credential Federation Enrolment Design Guide, Version 1.2
[SAML2-Tech]	Security Assertion Markup Language (SAML) V2.0 Technical Overview
[SAML2-Glossary]	Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0

2. About the GCCF

2.1 Cyber Authentication Vision

In 2007-08, Treasury Board Secretariat (TBS) initiated the Cyber-Authentication Renewal Project, a federal interdepartmental initiative that defines a new approach to authentication for the Government of Canada (GC). This project was an initial step towards achieving a desired state for the GC that gives departments and agencies flexibility to determine appropriate authentication solutions for individuals and businesses (end-users) to authenticate to their online services while ensuring the privacy and security of Canadians.
The Cyber-Authentication Renewal Project defined a new approach to authentication that includes a provision for multiple assurance levels commensurate with risk, standardized components and interfaces to be used as building blocks for authentication solutions, and

recommendations for a renewed policy framework on authentication. The focus of this project is external facing to citizens and businesses.
Figure 1 below shows the strategic vision for Cyber Authentication as an evolution from epass through the interim GC Access Key solution towards a standards-based solution that will see the government of tomorrow operating in an environment that supports the use of client-chosen credentials provided by multiple providers across multiple jurisdictions, through multiple service delivery channels, irrespective of the technology used.

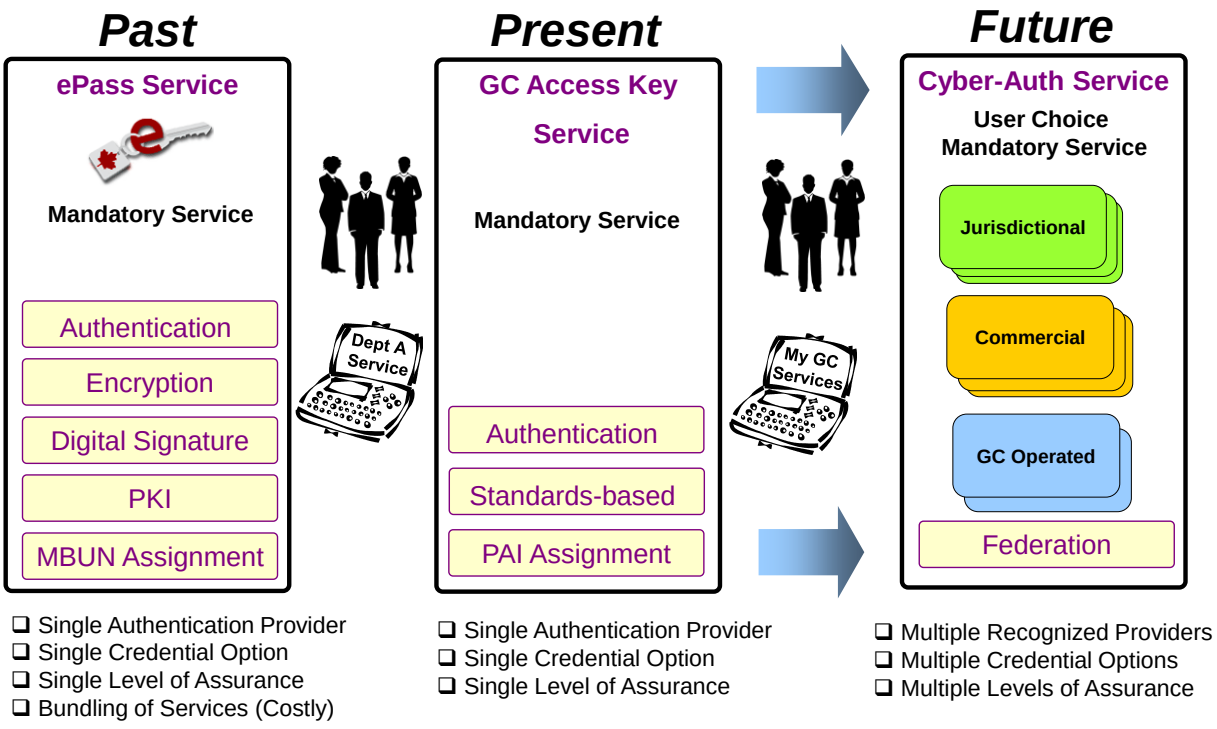

Figure 1: Cyber-Authentication Vision

2.2 The Federation Model

Figure 2 below illustrates the overall structure of the Government of Canada Credential Federation:

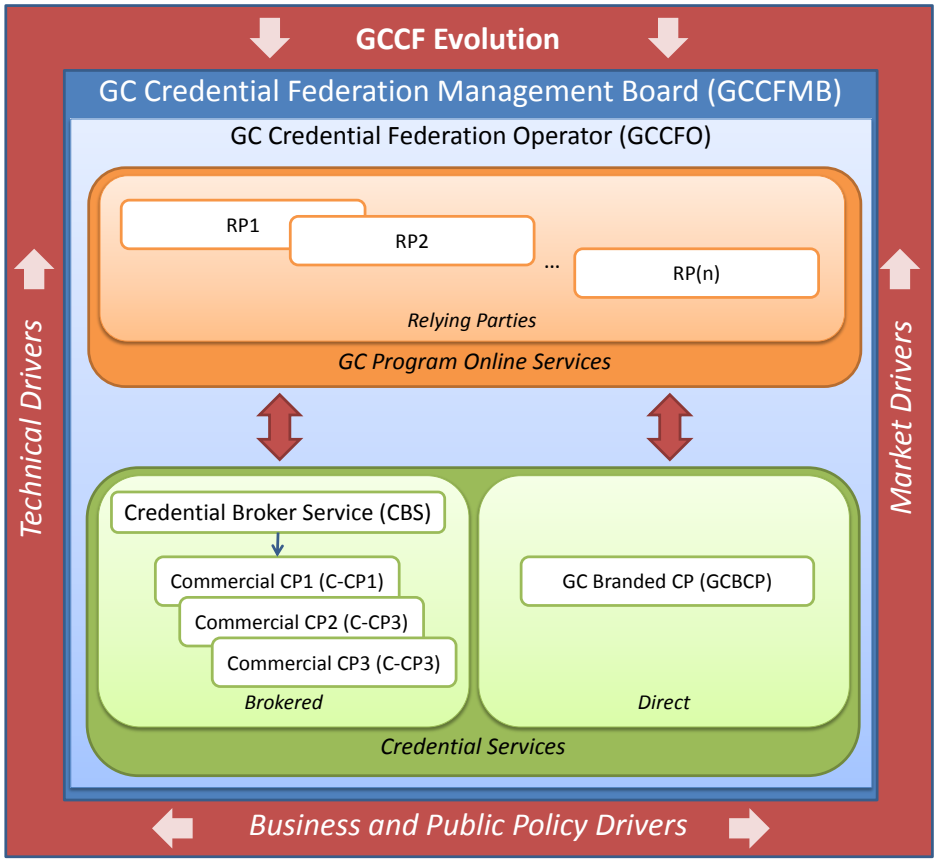

Figure 2: GC Credential Federation Model

2.2.1 The Federation Management Board

The Chief Information Officer Branch of Treasury Board Secretariat serves as the GC Credential Federation Management Board (GCCFMB). This is the governing body that develops federation policy and standards and also approves member's admission into the federation on behalf of the member community.

One of the most important standards developed and maintained by the GCCFMB is the "Cyber Authentication Technology Solutions Interface Architecture and Specification Version 2.0: Deployment Profile", often referred to as [CATS]. Based on the family of Security Assertion Markup Language (SAML) standards developed and maintained by the Organization for the Advancement of Structured Information Standards (OASIS), [CATS] defines the standard interface that on-line service applications use to invoke the authentication services offered by credential service providers. Because [CATS] is based on a widely supported industry standard, departments and agencies are free to choose from many available products and technologies in order to successfully connect their applications to the federation.

2.2.2 The Federation Operator

The Information Technology Services Branch (ITSB) of Shared Services Canada (SSC) serves as the GC Credential Federation Operator (GCCFO). The Federation Operator supports the day-to-day functioning of the federation by:

- supporting the mechanisms whereby federation members can be certain they are interacting with other legitimate federation members
- ensuring members are certified for compliance or compatibility with federation standards and providing means for reliably conveying the certifications that have been issued to each federation member
- aiding in problem resolution and/or technology compliance testing with or among members
- entering into contracts for credential services available to community members
- serving as the Point of Contact for concerns or complaints about improper conduct or failure to comply with standards on the part of a federation member.

2.2.3 GC Program Online Services

GC Program Online Services are those GC programs that provide services online to individuals and businesses. Within the federation model these are referred to as Relying Parties (RPs).

2.2.4 Credential Services

Credential services provide credentials to individuals and businesses and leverage those credentials to offer authentication services to GC Program Online Services.

2.2.5 GC Branded Credential Service (GCBCS)

GCBCS is a credential management and authentication service operated by the GC that will provide GC-branded credentials to allow individuals and businesses to authenticate for access to GC online services.

2.2.6 Credential Broker Service (CBS)

The CBS allows users to only select their preferred commercial credential to authenticate and mediates all authentication requests and responses between users, the approved C-CPs and the RPs. In future this may be extended to allow selection of GCBCS as well.
For the purpose of GC online services, the CBS interacts only with GC approved C-CPs. The service provider of the CBS will choose which 'approved' CPs will be accessible and the GCCFMB will approve their joining the GCCF. This is in alignment with the GCCFMB as the second step in positioning the GC to adopt a broad range of authentication options in the future.

2.3 Important Concepts

There are a number of important concepts that underlie the Government of Canada's approach to authentication:

A *level of assurance* is a specific measure of certainty (level of confidence), which may be relied upon by others, that a statement or fact is true. *Authentication* is the process of establishing truth or genuineness to generate an assurance.

Identity Authentication (sometimes called *identity proofing*) is the process of establishing confidence in the validity of an individual's claimed identity. This authentication process generates a level of *identity assurance*: the level of confidence that the person really is who they claim to be.

Credential Authentication is the process of establishing confidence that an individual has control over their rightful credential (i.e. not stolen), and that the credential has not been compromised. This authentication process generates a level of *credential assurance*: the level of confidence that the individual has maintained control over what has been entrusted to them (e.g. username/password, token, etc.) and that the credential has not been compromised (e.g. tampered with, modified, etc.).

Regardless of the type of assurance (identity or credential), levels of assurance are always measured to one of exactly four standardized levels that are numbered from 1 (little or no confidence) through 4 (very high confidence). These concepts are described in much more detail in Treasury Board policy instruments such as the Guideline on Designing Authentication Requirements as well as in Communications Security Establishment Canada documents such as (ITSG-31) - User Authentication Guidance for IT Systems.

3. Program Application Responsibilities

3.1 Identity Management

Credential Service Providers that belong to the GC Credential Federation only provide credential authentication assurances that contain a Persistent Anonymous Identifier (PAI); they do not provide any personally identifying information to GC online service applications. Applications are responsible for performing identity authentication and for capturing, storing and managing any personal or program-specific identity information about users. This means that GC Online Service applications must provide new users with an enrolment capability that allows them to create an account within the application and then associate a GCCF credential with that account that they can use to log in. Because the Cyber Authentication program will be offering a growing number of credential choices over time, applications must also provide existing users with a means to replace the credential that they currently use with a new different credential.

3.2 Authorization and Access Management

Authorization and Access Management is about allowing authorized users to access information and functionality from your application while preventing unauthorized users from doing so. Authorization and Access Management is not a service provided by the GC Credential Federation (GCCF); Departmental applications are responsible for implementing their own access control. This boils down to primary functions:

1. Requiring users to authenticate before allowing them to access information or functionality, and

2. Ensuring that authenticated users can only access the information or functionality for which they are authorized.

While the credential providers of the GCCF provide the means to perform authentication, your application is responsible for preventing access by unauthenticated users as well as for initiating the authentication process when needed by sending the user to a credential provider. There are a number of approaches to implementing access control. Depending on the design of your application you may choose to:

a) Implement your application behind a commercial web access management product. One advantage to this approach is that most of these products include support for the SAML protocol and can be easily integrated into the GCCF.
b) Use the access control capabilities of your web application framework. All modern frameworks including ASP.NET and J2EE included access control capabilities that you can leverage in your application. This approach can provide a greater amount of flexibility, although such an approach may also be more complex to develop, test and manage.

Whatever approach you choose, your access control implementation is a critical component responsible for protecting the privacy and trust of users.

3.3 Session Management

There are two main considerations when discussing authentication and session management:

1. Authentication is an activity that happens at a point in time. After authentication of a user has taken place there is a certain period of time during which an application assumes that the person using the application is still the same person that authenticated. The duration of this period of time often depends on factors such as the sensitivity of the application's information or functions and how active the user has been while using the application. Applications implement various types of session timeouts to either automatically end the session or to perform a re-authentication of the user when it is no longer considered safe to assume that the person who authenticated is still at the keyboard on the other end.
2. Users need to be able to explicitly log out and end their session so that they can feel comfortable leaving the computer unattended. This is particularly important when they are using a shared computer.

Performing logouts and managing session timeouts can become rather complex in a federated environment where the user may have multiple sessions active with multiple federation members' sites at the same time. Even in the simplest scenario where a user logs on to one departmental service, the user will have one session with the credential service and another one with the departmental application. If the user accesses multiple departmental applications using the same credential then there are additional sessions involved.

The GCCF has adopted the following principles in its approach to managing distributed sessions:

1. Departmental applications are responsible for managing their own session timeouts and should be free to implement whatever session timeout rules they consider to be appropriate.
2. Credential services must provide departmental applications with information about when the user was last authenticated, and must allow departmental applications to force re-authentication of the user at any time.

3. When a user initiates a logout from any federation site it must result in a global logout such that all of that user's sessions are terminated at all sites. This minimizes the risk that a user will unwittingly abandon a shared computer with an active session. Departmental applications are therefore required to initiate the global logout process if a user logs out of their site, and are also required to honour global logout notifications received from other federation members by terminating any active sessions.

3.4 Web Servers, Network Zones and Perimeter Security

In the founding years of the Government of Canada's Secure Channel, many security services such as site hosting, internet connectivity, firewalls and other perimeter safeguards were included as part of the shared authentication service. Under the current cyber-authentication model, these other security services are no longer bundled with the authentication service; leading to greater flexibility and reduced cost. This implies that departments and agencies are responsible for implementing the necessary perimeter safeguards to protect their networks and web applications.

4. SAML Overview

The Security Assertion Markup Language is the family of XML-based standards that many governments, including the GC, have adopted as the basis for creating an integrated and interoperable federation of applications and credential services. There are numerous available sources of information on SAML:

- The SAML article on Wikipedia provides a good high-level overview as a starting point.
- For additional detail there is also a good SAML Wiki Knowledgebase located at saml.xml.org.
- OASIS has published an excellent technical overview of SAML that is available at http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.pdf

SAML 2.0 is a flexible standard with numerous available features that many or may not be required by any particular federation. All federations must therefore decide on which required features of SAML will be used, define the details of how those features will be used, and finally identify all of the features that are not required and will not be used. The result of such an exercise is a more-detailed subset of the standard that is called a *profile* of SAML. The GCCF's [CATS] standard is a deployment profile of SAML 2.0 that is based on version 2.0 of the eGovernment implementation profile of SAML V2.0 which was developed by the eGovernment Working Group of the Kantara Initiative (http://kantarainitiative.org/confluence/display/eGov/Home). This relationship is shown in Figure 3 below:

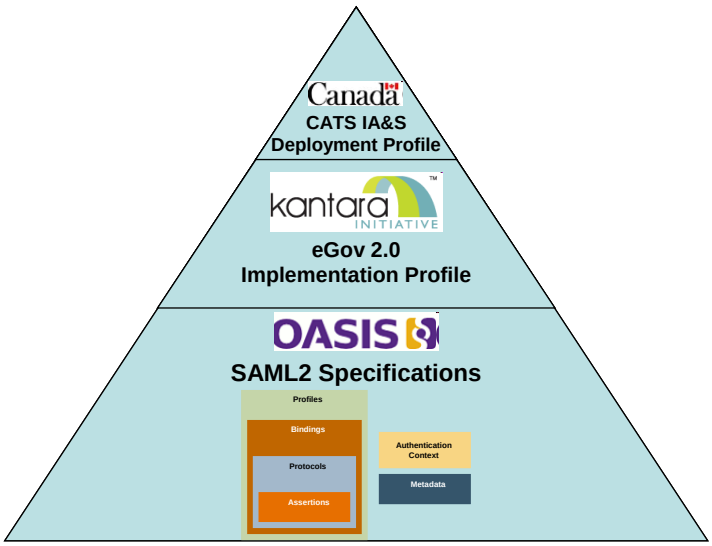

Figure 3: The foundation building-blocks of CATS

These documents make up the authoritative technical specification of how GCCF members are expected to integrate their applications into the federation. The objective of this section is not to duplicate this authoritative information, but to provide a less formal introduction to how SAML is used in the GCCF.

4.1 Bindings

In a nutshell, applications use SAML to integrate with credential services by sending and receiving XML messages to and from those services. In SAML, *bindings* define different ways that these messages can be sent from one system to another. These different ways of exchanging messages between two systems each have their own benefits and limitations. The GCCF uses the following three bindings to transport these messages:

4.1.1 HTTP Redirect Bidding

With the HTTP redirect binding, the message sender encodes and compresses the XML message into a URL-friendly string, and then sends the user's browser an HTTP message that tells it to redirect the user to one of the recipient's URLs with the encoded string appended in a query string parameter named "SAMLRequest". There are a number of important things to note about this approach:

- This is an indirect form of communication between the sender and receiver. It is the user's browser that actually delivers the message so there is no need for the sender to make a network connection with the receiver.
- The user starts on the sender's web site, but as the message is delivered, the user is transferred to the receiver's web site.
- Because there is a practical limit on how long a URL and query string can be, this binding is only appropriate for delivering smaller messages.

4.1.2 HTTP Post Binding

The HTTP post binding is similar in many ways to the HTTP redirect binding. The difference is that instead of passing the message in as a query string parameter in a URL, it is passed in an HTML form field. With this binding, the message sender sends an HTML page to the user's browser that contains an HTML form. That form includes a field (named "SAMLRequest" and normally hidden) that is pre-populated with the XML message to be delivered. The target of the form points to a URL of the message recipient so that when the form is submitted the message is sent to the recipient. The page typically includes JavaScript that automatically posts the form to the recipient system immediately so that the user never sees the page and no user involvement is required to deliver the message. Important things to note here are:

- As with the HTTP redirect binding, it is the user's browser that actually delivers the message, and the user ends up being transferred from the sender's web site to the receiver's web site. No network connection between the two systems is required.
- Form fields can contain much more data than a URL query string, so this approach is better than HTTP redirect for sending larger XML messages.
- Inevitably there will be some users that choose to disable JavaScript in their browsers. To accommodate this, the HTML page with the form needs to provide the user with a submit button they can use to manually submit the form to the recipient.

4.1.3 SOAP Binding

With the SOAP binding, the message sender uses the Simple Object Access Protocol to pass the XML message directly to the receiver, and possibly to receive a different response message in return. With SOAP:

- The messages are sent directly between the systems involved, so network connectivity between the two is required.
- The user and their browser are not involved. The user need not even be present anymore.

4.2 Protocols

SAML *protocols* describe how two systems exchange a pair of messages: a request message in one direction followed by a response message in the other direction. The purpose of each protocol is to perform a particular function. In the GCCF, the following protocols are used:

4.2.1 Authentication Request Protocol

The authentication request protocol implements the login function. When an application wishes to authenticate a user, it sends an authentication request message to the user's preferred credential service provider (CSP). The CSP then attempts to authenticate the user and returns an authentication response message to the application. If authentication was successful then the authentication response contains information about the user's credential within an XML data structure called a SAML *assertion*. If authentication was not successful then the authentication response contains a status code indicating the reason for the failure.

In the GCCF, authentication request messages are sent from applications to CSPs using the HTTP redirect binding, and the resulting authentication response message is returned from the CSP to the application using the HTTP post binding.

4.2.2 Single Logout Protocol

The single logout protocol is used to co-ordinate the global logout process across multiple sites. Logout functionality is particularly important in the context of the GCCF, since many individuals use public or shared computers to access GC online services. When a user clicks on a logout button they may have several active sessions at several sites (they will at least have two: one at a department site and one at the CSP site) and it is important that all of those sessions are terminated so that the user's personal information cannot be accessed by the next person that uses the computer.

When the user clicks a logout button on an application page, GCCF rules require that the application must send a logout request message to the CSP. The CSP then returns a logout response message once the user is logged out of the CSP. If the user has logged on to multiple applications during their session then the CSP will send additional logout request messages to all of those other applications to make sure that those sessions are terminated as well. Applications must therefore be able to send and receive both logout requests and logout responses.

In the GCCF, most logout request and response messages are sent and received using the SOAP binding, however applications are also permitted to use either the SOAP binding or the HTTP redirect binding to send logout requests to a CSP.

4.2.3 Name ID Management Protocol

Some credential service providers in the GCCF allow for a credential to be revoked or permanently disabled. Once a credential has been revoked it can never again be used to authenticate to a GC online service and some applications may wish to flag, clean-up or remove any user account information associated with it. In the GCCF, applications can optionally register with these credential service providers to receive notification of credential revocations in the form of Manage Name ID messages.

In the GCCF, Manage Name ID messages are sent using the SOAP binding.

4.3 Entity Identifiers

Every SAML implementation in the federation is assigned a unique name called an Entity ID. Entity IDs are used for many purposes including identifying the issuer and intended recipient of SAML messages. Entity IDs in the GCCF use the syntax of a URL in the federation member's DNS domain (e.g. https://spc-ssc.gc.ca/cyberauth/testenv1), however there is no requirement for members to host a web page at the location specified in the URL. Federation members are generally free to define whatever Entity IDs they wish to use for their various development, testing and production environments, subject to review and approval by SSC.

4.4 Persistent Anonymous Identifiers

The GC Authentication Federation has adopted the term Persistent Anonymous Identifier (PAI) to refer to the mechanism whereby federation members such as relying parties and credential

service providers refer to a specific credential holder. This term has been adopted to more closely align GC terminology with that used in the SAML industry standards:

> ***Persistent Identifier:*** *is a persistent opaque identifier for a principal that is specific to an identity provider and a service provider or affiliation of service providers. Persistent name identifiers generated by identity providers MUST be constructed using pseudo-random values that have no discernible correspondence with the subject's actual identifier (for example, username). The intent is to create a non-public, pair-wise pseudonym to prevent the discovery of the subject's identity or activities. Persistent name identifier values MUST NOT exceed a length of 256 characters. A given value, once associated with a principal, MUST NOT be assigned to a different principal at any time in the future.*

The Government of Canada's first shared credential service, called *epass,* employed a similar concept that was referred to at the time as a Meaningless But Unique Number, or "MBUN". The MBUN identifier was a specific example of a PAI but it had a number of limitations that PAIs do not:

- The MBUN was specific to the *epass* credential. A PAI on the other hand is specific to an identity provider and a relying party or affiliation of relying parties. This means that it is possible for a credential provider to use different PAI values with different departments for the same credential. This capability can provide an additional level of privacy protection as different departments are unable to correlate their respective user databases using the common identifier.
- The MBUN was a static identifier: once an MBUN was created for a credential and mapped to an individual's identity by a program it could never be changed. In contrast, with PAIs the SAML standards provides additional flexibility to support the re-mapping and de-mapping of PAIs over time. These features may be used by the GC in the future to address challenges such as credential revocation and user migration.

4.4.1 Sensitivity of the PAI

The fundamental design concept behind the PAI is to "prevent the discovery of the subject's identity or activities". Nevertheless a PAI is considered Personal Information under the Privacy Act in so far as it is ultimately linked to an individual. This implies that safeguards are required to protect PAIs wherever they are associated with other personal information that could be used to identify an individual. Three examples of where this occurs are:

- Within departmental mapping databases where PAIs are mapped to identity information via program identifiers. Departments are responsible for protecting these mapping databases with acceptable safeguards in accordance with MITS.
- Within the user profile databases of credential providers, where PAIs are mapped to userids, recovery questions and answers, or other data elements that might help to identify the credential owner. These profile databases must be protected by the Credential Service Provider in accordance with applicable policy, legislation and contractual requirements.
- Within electronic communications that involve the individual: there are some cases where the PAI is included in electronic messages involving the individual's computer. For example, SAML messages exchanged between federation members using the HTTP post or redirect bindings are exchanged via the user's browser. This creates an opportunity to associate an individual's PAI with the IP address of their computer. MITS

therefore requires that communications with the user's computer over the Internet that include the PAI should be encrypted. In the case of SAML messages, the GC CATS specification mandates the use of TLS or SSL3 with CSEC-approved algorithms to accomplish this.

4.4.2 Uniqueness of the PAI

It is important to note that PAIs from different CSPs are not guaranteed to be unique. Both the PAI and the EntityID of the CSP are required to uniquely identify a credential.

4.5 Assertions

As mentioned in section 4.2.1 above, a CSP responds to authentication requests from applications by returning an authentication response which contains an XML data structure called an *assertion*. Figure 4 shows an example of a SAML assertion.

```
                    <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
                                    ID="s2de01458dac159fc3ab88a185e7a8aa5f460184bd"
                                    IssueInstant="2011-06-06T15:38:11Z" Version="2.0">
CSP Entity ID ->       <saml:Issuer>https://idsim1.gccatslab.net</saml:Issuer>

                       <saml:Subject>
                          <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
Credential PAI ->            ALpIfX6E8M5xERYBQx0n1T20iNQn
                          </saml:NameID>
                       </saml:Subject>

                       <saml:Conditions NotBefore="2011-06-06T15:28:11Z"
                                        NotOnOrAfter="2011-06-06T15:48:11Z">
                          <saml:AudienceRestriction>
                             <saml:Audience>https://rpsim1.gccatslab.net</saml:Audience>
                          </saml:AudienceRestriction>
                       </saml:Conditions>
  Authentication
    Timestamp ->       <saml:AuthnStatement AuthnInstant="2011-06-06T15:38:11Z"
                                            SessionIndex="s20fb4844f77839594038e04401e722d7545"
Session Limit ->                            SessionNotOnOrAfter="2011-06-06T23:38:11Z">
                        <saml:AuthnContext>
                             <saml:AuthnContextClassRef>
Assurance Level ->              urn:gc-ca:cyber-auth:assurance:loa2
                             </saml:AuthnContextClassRef>
                          </saml:AuthnContext>
                       </saml:AuthnStatement>
                    </saml:Assertion>
```

Figure 4: Example Assertion

The assertion contains a number of data elements that applications can use. The most important of these are:

1. The <Issuer> element contains the unique name (Entity ID) of the CSP.
2. The <NameID> element contains the CSP's unique identifier (PAI) for the credential.
3. The <AuthnInstant> element contains the date and time that the user last authenticated by entering their credentials. This may be some point in time earlier than when your application sent the authentication request since it is possible that the user was already logged in to the CSP.
4. The optional <SessionNotOnOrAfter> element specifies a future point of time after which the CSP's session with the user can expire.

5. The <AuthnContextClassRef> element specifies the level of credential assurance supported by the authentication process that took place.

Applications use the contents of the first two elements, the CSP's Entity ID and the credential's PAI, to uniquely identify the credential that the user authenticated with. These are the two data elements that need to be associated with (i.e. mapped to) the user's account in the application.

Applications may wish to examine the timestamp included in the <AuthnInstant> element to determine how recently the user actually provided their credentials to the CSP. GCCF rules allow a CSP to passively satisfy authentication requests (without interacting with the user) for up to 20 minutes after a user authenticates. This provides the user with a single sign-on experience.

CSPs may provide a date and time value in the SessionNotOnOrAfter attribute to let applications know how long the CSP is willing to keep track of its own session with the user. While applications are free to implement whatever session timeout rules that they wish, if the duration of the application session exceeds the duration of the CSP session, then the CSP will no longer be able to co-ordinate the global logout process. In order to mitigate the risk of this happening, GCCF rules require that CSPs may not expire their session with the user until at least 8 hours after the user initially authenticates.

4.6 Security

There are two fundamental security requirements that apply to the exchange of SAML messages between applications and CSPs: Privacy and Trust. Privacy is concerned with ensuring that any personal information contained within the SAML messages is protected from eavesdropping through the use of encryption. Trust is concerned with ensuring that a message recipient only recognizes genuine SAML messages that are received from a legitimate sender through the use of digital signatures.

4.6.1 TLS Encryption

The GCCF uses two complementary methods for protecting the content of SAML messages from potential eavesdroppers using encryption. The first method used Transport Layer Security (TLS) to encrypt all HTTP network traffic.

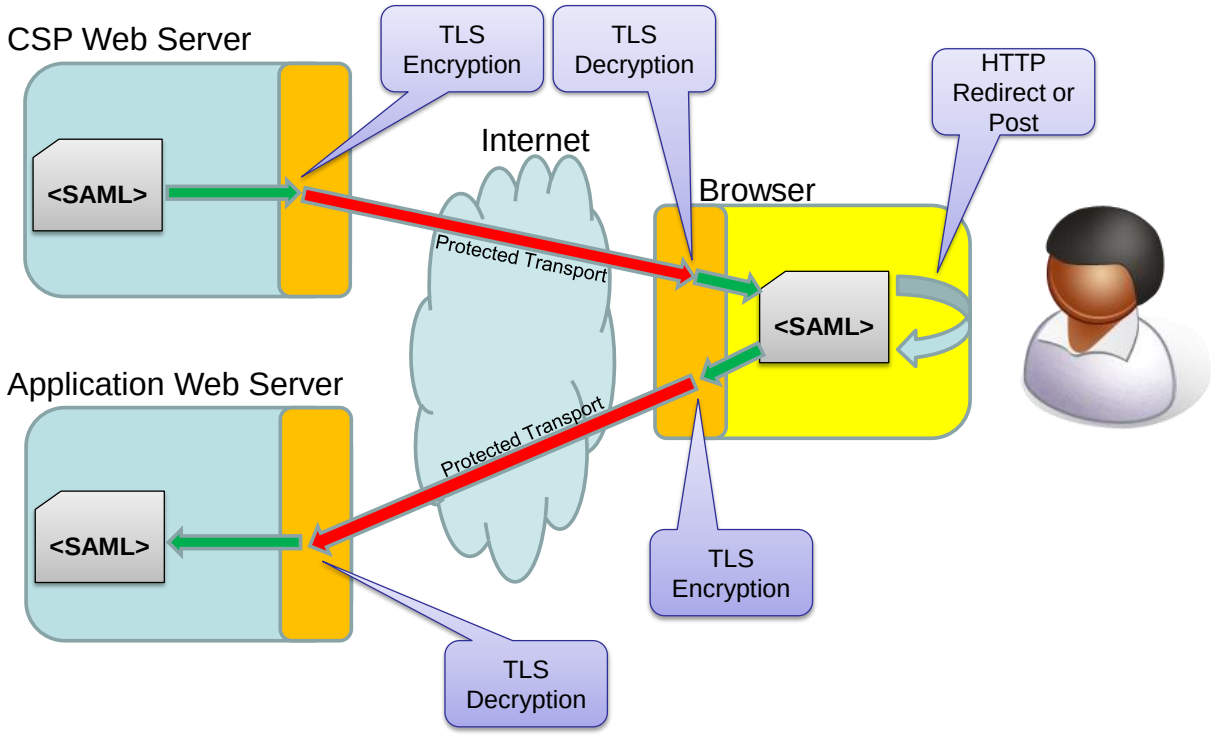

Figure 5: TLS encryption with the HTTP redirect and post bindings

Figure 5 shows what happens when a CSP sends a SAML message such as an authentication response to an application using the HTTP redirect or post binding. Once prepared for transmission, the SAML message is packaged into an HTTP message that then passes through the TLS module of the CSP's web server. The TLS module then encrypts the HTTP message before sending it to the user's browser over the Internet.

When the HTTP message arrives at the user's browser, the browser uses its own TLS module to decrypt the HTTP message into its memory. Once the message has been decrypted into the browser's memory, the browser can then read and process the message. Normally the browser immediately re-sends the message to the application's web server. The HTTP message is therefore re-encrypted by the browser's TLS module and sent to the application's web server where it is decrypted by the web server's TLS module and then processed.

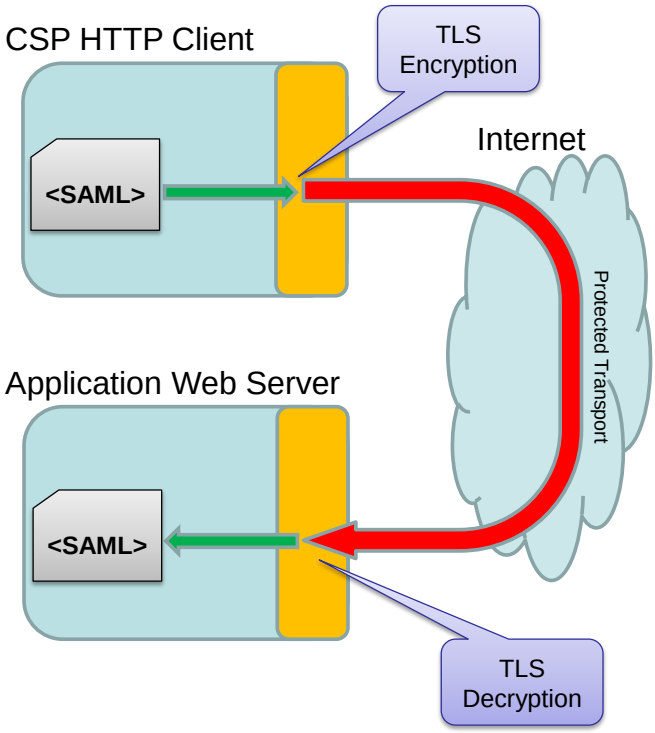

Figure 6: TLS Encryption with the SOAP binding

Figure 6 shows the somewhat simpler scenario that occurs when the SOAP binding is used. In this case the SAML message is encrypted and decrypted only once since the user's browser is not involved in the exchange.

4.6.2 XML Encryption

In addition to using TLS to encrypt HTTP traffic, the GCCF also uses XML encryption to further protect personal information in the SAML messages themselves. This provides an additional layer of security to protect information while it temporarily resides in the memory of the user's browser. Specifically, XML encryption is applied to:

1. The Assertions contained within Authentication Response Messages, and
2. The PAI contained within Logout Request Messages

Figure 7 shows an example of an assertion that has been protected using XML encryption.

```
<saml:EncryptedAssertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
                       Type="http://www.w3.org/2001/04/xmlenc#Element">
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <xenc:EncryptedKey>
            <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
            <xenc:CipherData>
               <xenc:CipherValue>
maLpKFN5K9CNV57qfUettT1L9sb83wFc0kkQc3T1xNrVG9qX2EXN6Eo4Umhi3jeIB6H6em79oU4m
xTq0W8pqy3tpAII1mIr4CBZemiJTcQE85LH/u1ZCszNX3mI5kEW3xwA/R+rv3W7nAGyboLSJ9BZk
WFbNbi2hIrQtauHt0xg=
               </xenc:CipherValue>
            </xenc:CipherData>
         </xenc:EncryptedKey>
      </ds:KeyInfo>
```

```
      <xenc:CipherData>
         <xenc:CipherValue>
f3KNoh+bqhoPaEoK3bwANwQJVKXnM8JVl0t2dRbspsTGnmd7WEuAoL59kjMurMMLkszU8ZDICdck
zbwYYCU1b1jbuJysf2uEhHi4uCDHWhVMP2t5Utj/1r699ThnSHJ9mPS7OC6gTFE4vfQ3gA8dPQb2
Kj/2bd7NrWaQLNFu75xEChhWwANcQp6XB1lYPnhroeKVwUopfljjEa7fKgrJBGbUeLm9uq95uNIF
eMzNQVEFRYmVcYGAiLJkphqj/z+7FvK6JKHdoDCRFMEaShe4+KhHqApZyEmAiu2Hi1IMePX//Omt
bk6OPquK1BB2AWF2ggG7tHyipdGrpJzZlprAwqZQnaLColHEHHAejVjTBlGDVpCyERj/OPAB1xfN
Pa4FLz4quXjzaqn+tdbXp91YGxgI5vdI0oYJs3xopWJIb0m6IuDOoKV0aXbJ2wtxFBmNebdBhWXG
HlJ33drtAb5x/7VwdaFR3p3eIruRHA5gJWMogK8BXzazoS2mvcadYuvzyJTbbf/j/9eWK6z7fB+i
SvheaMoqs25lx+1sdg2fdBkb2c3cJ03V7cs/DJeKtVKSslp0MrLpfw91ykR6IGwypVgnazsmn8jj
TVSIuqUmvfNY1q90Q6QrgXtXtPGR60C/1IpNSLrZ1NYWMMYNMTUpz5MjGp16VfpNkEGEg8DvKmd2
q4gT0XSi7m5goUae6Sz3Yua7rhKG0y8/Y8p6bFItmeN8ujyn/Pcq/FsplVGzn0aQWQj3vi23TmiV
+wfCA68qdqFVTHz53BkWdwMDT6DrdrK734Se+OnjMHWkNA1BKX6/eC8om/6fwNi2v222vLJMnE3/
ZCZqC1AElxhBHI2yUOWWleRGh++Bnp/M11nhcttLlvlzYEuTHjlVEZ2OopM9aaKNWG9jBZktqlPr
YPHJDHmKSlLq+j9UItWQpZMfEzwliq7MpWwuDxsg/eQsefLQ6R1K9A/Fd5qS04GWcje280X4eMSm
lG6VBkADFtlyklKZuWFIixNtiS/k34gc91ZHkkLahdZ2Tj2WmkBb/5q7lFwjZFUN5bjtb7/LVVpp
tFXdWS/BssjUy3rVSk/o045nkWG5ULRtthk9/FrsfnnX2go/4mi5vkKZIiI8jEXgQ7NpTNoP8zdt
6TGZhTkq3qvWsGN+WBE6kmvxd/HChHUMPSh9rrYJAQva/T83Wzicl9NY0aN2EWFE7sO1LGn1VEky
zpbAFzWdQeLtqHhxnDkkWc5yuLLDnZQYzmfDEdXQ+DlFmT4rb/jp9fhLPZxgCqPmjoRhqArqko6S
cZNldf5BikAStRSbT6UeTs97CnqdaWzBefvUqpPvrsvXbdVUTVCHy4KTVufKMkc0/7exFxvKqqtx
SE++MUJYQ6Wlt3F16yf48/Mnz6cwFT0SYEX433XG/JHIM4znGpanN6hcIGiYw7GgjcWvDpJOVVvc
TGxg1YHM49miN177zv8CqhUcFWQjadw27c2Wdgsf04cp+4Nnhj827sCTXySajIrzFTjmtCk80kx8
woC7UMD7fpM1Yj4LvF8A9/+ro9nBxakQZxLqtHLovJaQ1eWRWr+3nSk3t1sUlETI6oPnCFPDAKZx
znME8OvguJY1Kp+SsijiVqh+bWN5KBd+pygAOeoAeXKPYMHw1eMljUX1kQ6pro2en+GVoIDGFTPY
PaaTypGS/q47RaVUdBZJ5v7mlf1JeCFEmC4xBlEY3H4qTbcOvhbfl/sSS8pcpYLcqaq6TGSDVjc2
bsWU87nEF7LZJAoaY+r6e4ukzbzEzF6CynMI08usymotIhyzb31l7DuTJZ//hEpaiCt1VsAJMzRT
cWviUvayahWC3Yn1RINgahSMduvn48IGqkSMlB68RMNFesYdND7w
         </xenc:CipherValue>
      </xenc:CipherData>
   </xenc:EncryptedData>
</saml:EncryptedAssertion>
```

Figure 7: XML Encryption applied to an Assertion

4.6.3 Signing and Verification

In order to prevent an attacker from using a forged SAML message to access or disrupt an application or a CSP, GCCF rules require that all SAML messages must be digitally signed by the sender. Message recipients are in turn required to verify the digital signature on all messages before accepting them.

4.7 SAML Metadata

Much of the work involved in configuring the SAML integration between an application and a CSP involves exchanging information about each other's various URLs. Your application needs to know which of the CSP's URLs you should send SAML messages to, and the CSP needs to know what your URL is for receiving responses. Another requirement is for the CSP and application to exchange information on each other's PKI certificates so that one can successfully sign or encrypt information for the other.

In order to make this easier, the SAML standard includes a standardized XML format that is used for exchanging this type of metadata information.

5. Architecture and Technology Options

There are a wide variety of both commercial off-the-shelf and open source software products to choose from that can help you integrate your GC online service application into the GCCF using SAML. The various features and functions offered by each of these products can vary significantly. This can make it difficult to decide which product or products are right for you.

The purpose of this section is to help you to evaluate your architecture and technology options by providing general information about the major architectural components that you need to implement, and then give an overview of the different types of available products and what capabilities they provide in terms of those major components.

5.1 Reference Model

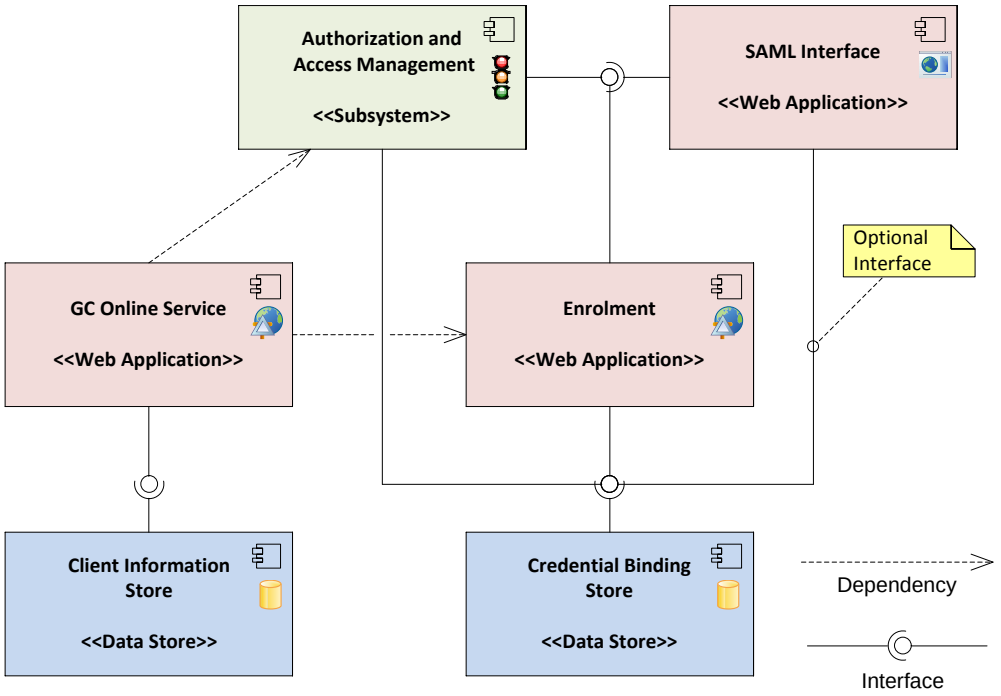

Figure 8: Reference Model

The logical component diagram in Figure 8 shows the six major components of a GC online service that is ready to integrate with the GCCF. These components are:

5.1.1 Your GC Online Service Web Application

This is the actual web application that delivers GC services to your clients. It depends on all of the other components to make certain that it is dealing with the right client when delivering services over the Internet.

5.1.2 An Authorization and Access Management Subsystem

This component is responsible for allowing authorized users to access information and functionality from your application while preventing unauthorized users from doing so. Authorization and Access Management was described in more detail back in section 3.2.

5.1.3 A Client Information Store

The client information store is the directory or database where you store information specific to Individuals for the purpose of providing them with services for which they are authorized.

5.1.4 A User Enrolment Web Application

Enrolment is the process by which a user establishes (or re-establishes) an account in your online service application. There are many possible ways of doing this. You may create a new account to go along with the new credential or the user may have an existing account to which you add the new credential. You may wish to authenticate the identity of the user prior, during or after the enrolment process. However you decide to implement your enrolment process is up to you, but you will need some kind of web application that creates a mapping that links the PAI of the user's credential together with the issuing CSPs Entity ID to their account in your repository. Additional information on this subject is available in a companion document: The GCCF Enrolment Design Guide.

5.1.5 A Credential Binding Store

This data store maintains the associations between each Individual's Credential(s) and their record in the Client Information Store. Implementing this separately from the Client Information Store with its own distinct and segregated security safeguards reduces the risk of a privacy breach if either of the two data stores is compromised in some way.

5.1.6 A SAML Interface Web Application

Finally, you will need to stand up a web application to implement your SAML interface to the federation. Remember that all of the bindings used to exchange SAML messages use the HTTP protocol, so even though it serves up SAML and XML messages instead of user interface pages, this component is still just another web application.

5.2 Architecture Options

Before you start looking at the various commercial and open source products and technologies, you should look at each of the major components in the above reference model and ask yourself the usual questions as to how you want to implement each:

1. Do you want to implement the component using custom, commercial off-the-shelf or open source software?
2. What kind of team will be implementing and supporting the solution? Will it be mostly system administrators, mostly developers or a mix of both?
3. How should each component fit within your multi-tier architecture? Which components do you want to implement in which tier?
4. Are you looking for a comprehensive all-in-one solution from a single software provider, or do you prefer to select and integrate best-of-breed components?
5. What are your platform preferences in terms of operating systems, web servers, application platforms (e.g. Java, .Net), directories and databases?
6. What existing assets are available for re-use within your environment?

7. Do you want to receive notice of revoked credentials via Manage Name ID messages? (This may narrow your choices as not all products support this type of SAML message.)

5.3 Technology Options

To assist you in your research, this section defines a few high level categories into which most software products and technologies can be grouped.

5.3.1 Web Access Management Products

Web Access Management (WAM) products provide Authorisation and Access Management functionality for web applications. Authorization rules are normally defined based on the URL of the resource that a user is trying to access, for example a different URL pattern may be used to distinguish between protected and unprotected pages. All of these products also implement some kind of User Account Repository, typically through the use of a relational database or LDAP directory that you must obtain separately. Many WAM products also have their own a SAML Interface, although this is sometimes sold separately as an add-on and not included in the base product. Many vendors that offer a WAM product also sell a complementary Standalone Federation Product (see the next section).

Some examples of Web Access Management products are:

Product	SAML Interface	URL
CA SiteMinder	Optional (FSS)	http://www.ca.com/us/internet-access-control.aspx
Entrust GetAccess	Included	http://www.entrust.com/internet-access-control/index.htm
ForgeRock OpenAM	Included	http://www.forgerock.com/openam.html
IBM Tivoli Access Manager	Standalone product sold separately	http://www-01.ibm.com/software/tivoli/products/access-mgr-e-bus/
Novell Access Manager	Included	http://www.novell.com/products/accessmanager/
Oracle Access Manager	Standalone product sold separately	http://www.oracle.com/us/products/middleware/identity-management/oracle-access-manager/overview/index.html
RSA Access Manager	Standalone product sold separately	http://www.rsa.com/node.aspx?id=1186

5.3.2 Standalone Federation Products

Standalone Federation Products provide a SAML Interface but are intended to be integrated with a separate Authorization and Access Management solution: either a full-featured Web Access Management product such as those described in section 5.3.1 above, or sometimes with the more basic access management capabilities provided by web servers themselves.

Standalone Federation products normally include some ability to integrate with an existing User Account Repository, but few provide a full featured repository of their own.

Some examples of standalone federation products that are designed to work in concert with a full-featured WAM product are:

Product	URL
IBM Tivoli Federated Identity Manager	http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr/
Oracle Identity Federation	http://www.oracle.com/us/products/middleware/identity-management/oracle-identity-federation/overview/index.html
Ping PingFederate	https://www.pingidentity.com/our-solutions/pingfederate.cfm
RSA Federated Identity Manager	http://www.rsa.com/node.aspx?id=1191

Other examples of more lightweight standalone federation products that may use a web server (or reverse proxy server) to perform basic authorization and access control are:

Product	URL
CA Federation Manager	http://www.ca.com/us/products/detail/CA-Federation-Manager.aspx
IBM Tivoli Federated Identity Manager Business Gateway	http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr-bg/index.html
Ping PingFederate Express	https://www.pingidentity.com/our-solutions/pingfederate-express.cfm
Shibboleth	http://shibboleth.internet2.edu/

5.3.3 Application Frameworks, Plugins and Toolkits

If you prefer to implement your own User Account Repository and Authentication and Access Management subsystem (perhaps by leveraging features that are included in the application development platform you are using), then there are a number of application frameworks, plugins or toolkits that you can use to just implement a SAML interface. Some examples of these are:

Product	URL
The OpenAM Fedlet for Java	http://www.forgerock.com/openam.html
The OpenAM fedlet for .NET	http://www.forgerock.com/openam.html
SimpleSAMLphp	http://simplesamlphp.org/
Microsoft Windows Identity Foundation SAML 2.0 Extension	http://blogs.msdn.com/b/card/archive/2011/05/16/announcing-the-wif-extension-for-saml-2-0-protocol-community-technology-preview.aspx

5.3.4 Security Token Services

If you have developed your application using technologies that do not support SAML, but do support federated authentication using a standard other than SAML (for example WS-Federation) then you may wish to implement a Security Token Service (STS). An STS can act as a translation broker between your application's non-SAML interface and the SAML interfaces of GCCF Credential Service Providers.

Several of the products listed earlier can also be configured as an STS. As well, there are some "pure" STS products on the market. Some examples are:

Product	URL
IBM Tivoli Federated Identity Manager	http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr/
Microsoft Active Directory Federation Services	http://www.microsoft.com/windowsserver2008/en/us/ad-fs-2-overview.aspx
Oracle Identity Federation	http://www.oracle.com/us/products/middleware/identity-management/oracle-identity-federation/overview/index.html
Ping PingFederate	https://www.pingidentity.com/our-solutions/pingfederate.cfm
RSA Federated Identity Manager	http://www.rsa.com/node.aspx?id=1191

6. Application Integration

Once you have decided on your architecture and chosen the products you will use, the next step is to implement your solution and integrate your application into the GCCF.

6.1 Join the GCCF Technical Community

With over 20 member departments and agencies, the GCCF has a thriving technical community of architects, developers and system administrators with real-world experience of what it takes to successfully integrate an application into the federation using different products and technologies. As part of its role as Federation Operator, SSC operates an on-line discussion forum where this community can share knowledge, develop and maintain re-usable solutions, and collaborate to solve problems. If you haven't already, you should contact your SSC Client Implementation Team contact in order to get access to this forum, leverage its valuable content and collaborate with your peers across the federation.

6.2 GCCF Technology Toolkits

There are a number of technology toolkits that have been created by the federation technical community to help new applications integrate into the GCCF using specific products or technologies. These are intended to save you time and effort that you might otherwise expend if you had to "start from scratch" with your implementation. These toolkits may include valuable resources such as:

- **Installation and Configuration Guides** that help you to successfully install the product and quickly configure it to comply with GCCF standards,

- **Sample Code** that you can re-use to customize or extend the product to implement commonly required or desired functions, and
- **Page Templates** to assist you in designing certain user interface functions that are commonly required.

Information on what toolkits are available and how to get them can be found in the GCCF technical community forum.

6.3 User Interface Elements

Much of this document so far has focused on the SAML interface between your application and the federation. In order to provide your clients with a smooth (and CLF-compliant) authentication experience, there are also a number of user interface elements that you need to implement.

6.3.1 Departmental Choosing Page

If you refer back to Figure 2 back in section 2.2, you will see that the GCCF architecture model shows two arrows that link GC Online Services with Credential Services. The first arrow links to the Credential Service Broker while the second arrow links to the GC-Branded Credential Service. These two connections support a user's ability to choose if the wish to authenticate to your application using either a commercial credential or a GC credential. In order to support this functionality your user interface will need to include a page that presents the user with these two choices. In the GCCF architecture this is referred to as the departmental choosing page.

Some products that can be used to implement a SAML Interface come with a built-in choosing page that you can customize; others expect you to implement this page yourself. The GCCF technical community's online forum is an excellent place to look for information on how to implement a choosing page with the product you have chosen.

6.3.2 Session Language Passing

The Government of Canada's Official Languages Act and Common Look and Feel Policy require a way to communicate the user's current language preference whenever their browser is redirected from one site to another. This is one of the more interesting problems that the GC has had to deal with since this GC-specific requirement is not explicitly addressed by the SAML standard or any other industry standard. The solution that the GCCF decided upon utilizes a session cookie named "_gc_lang". Sites are required to set this cookie to the user's preferred language ("eng" or "fra") before sending them to another site (for example, using either the SAML HTTP redirect or post binding) and are required to read the cookie to ensure that pages are displayed to the user in the correct language. Since the HTTP and cookie standards do not allow sites in different DNS domains to see each other's cookies, SSC registers a DNS CNAME alias for each federation site in a common DNS domain (fjgc-gccf.gc.ca) so that everyone can access this common domain cookie.

Sample code for reading and writing the common domain language cookie is available in the GCCF technical community's online forum.

6.3.3 Departmental Name and Logo

In order to provide a more consistent user experience, some credential services can include your department or agency's name and logo on select user interface pages. If you wish to make use of this feature you will need to place the logo on one of your web servers and provide the URL of the logo (it must be HTTPS) together with your department or agency name in your SAML metadata.

6.3.4 Error Pages

In the event that any kind of SAML error occurs, you will need to implement error pages that conform to the GC CLF standard. This may involve some customization of the product you used to implement your SAML interface. Examples for your product may be available in the GCCF technical community's online forum.

6.4 Trust Infrastructure

In order to support all of the encryption and digital signature safeguards describe in section 4.6 above, you will need to obtain a number of digital certificates:

6.4.1 SSL Certificates

You will need to purchase a commercial SSL certificate for your web server(s). In addition to enabling the use of SSL/TLS to protect the pages of your application's web site, this certificate can also be used to protect SAML messages that are exchanged.

6.4.2 Signing and Encryption Certificates

You will need a PKI encryption certificate issued by the GC Internal Credential Management (ICM) service. This will be used by CSPs to encrypt assertions for your application.

You will also need a PKI signing certificate issued by the GC Internal Credential Management (ICM) service in order to digitally sign your authentication and logout requests.

6.4.3 Certificate Revocation Checking

Your SAML interface is required to regularly check the status of the ICM certificates that are used to sign the SAML messages that it receives. This involves configuring your SAML product to connect to the ICM LDAP directory where certificate revocation lists are published. Information on configuring your product is likely available in the GCCF technical community's online forum.

6.5 SAML Metadata Exchange

As introduced in section 4.7 above, SAML metadata is a standard XML file format that significantly reduces the amount of work required to configure your application's SAML interface to integrate with a Credential Service Provider. While the details of the configuration process may vary from product to product, the fundamental steps that take place are the same:

1. You start by obtaining the CSP's SAML metadata file from SSC and import it into your application's SAML interface product
2. You then perform any additional configuration of your SAML Interface product that is required
3. Next, you produce your own SAML metadata file and send it to SSC where it will be validated before being forwarded to the CSP
4. Finally, the CSP will import your SAML metadata and perform any additional configuration of their SAML interface product

Once the above steps are complete, your SAML interface should be able to successfully exchange SAML messages with the CSP.

7. Transition to Production and Operations

In a federated environment, maintaining users' trust and their confidence in the convenience and security of their online interactions with the GC is everyone's responsibility. Government of Canada privacy and security policies and standards need to be applied consistently by every department and agency in the federation.

7.1 Privacy Impact Assessment

Your application's user account repository almost certainly meets the Privacy Act's definition of a personal information bank. The Privacy Impact Assessment of your online service must therefore address your identity authentication and user enrolment processes.

7.2 Certification and Accreditation

Departments and agencies must have their GC online service applications certified and accredited before the GC Credential Federation Management Board will approve them for operation within the Federation.

7.2.1 Compliance Self-Assessment

Comprehensive testing of the integration interfaces between applications and CSPs plays an important role in ensuring that user's confidence is maintained. SSC maintains a standard compliance toolkit to facilitate the thorough and consistent testing of all integration interfaces between GC online service applications and the federation. This toolkit includes a standard set of test cases and some helpful testing tools intended to make this testing as easy as possible. GCCF rules require that applications perform compliance and security testing and fill out a compliance traceability matrix (included as part of the toolkit) before they go live, and then to repeat the testing process on an annual basis.

7.3 Configuration and Change Management

You will probably maintain a number of non-production environments for development and testing to support your normal configuration and change management procedures. All Credential Service Providers in the GCCF provide at least one testing environment to support these non-production environments. Non-production SAML interfaces will need their own unique SAML

Entity ID to uniquely identify them. They are also expected to use non-production ICM certificates issued by the ICM Client Acceptance Testing Environment (CATE). You do not need to obtain separate ICM CAGTE certificates for each non-production environment however, they can be re-used by as many non-production requirements as you need.

7.4 Help Desk Integration

In order to assist users with both application-related and authentication-related difficulties, the help desk that you provide to your end users will need to integrate with the help desks operated by the Credential Service Providers.

8. Glossary of Acronyms

CATE	Client Acceptance Test Environment
CATS	Cyber Authentication Technology Solutions
CBS	Credential Broker Service
CIOB	Chief Information Officer Branch, Treasury Board Secretariat
CSP	Credential Service Provider
GC	Government of Canada
GCBCS	Government of Canada Branded Credential Service
GCCF	Government of Canada Credential Federation
GCCFMB	Government of Canada Credential Federation Management Board
GCCFO	Government of Canada Credential Federation Operator
HTTP	Hypertext Transfer Protocol
ICM	Internal Credential Management
LDAP	Lightweight Directory Access Protocol
LoA	Level of Assurance
MBUN	Meaningless But Unique Number
MITS	Operational Security: Management of Information Technology Security
OASIS	Organization for the Advancement of Structured Information Standards
PAI	Persistent Anonymous Identifier
SSC	Shared Services Canada
RP	Relying Party
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SP	Service Provider
SSL	Secure Sockets Layer
TBS	Treasury Board Secretariat
TLS	Transport Layer Security
URL	Uniform Resource Locator
WAM	Web Access Management
XML	Extensible Markup Language