# NET Plus Identity Guidance for Services

## Supported Service Provider Configuration Pages

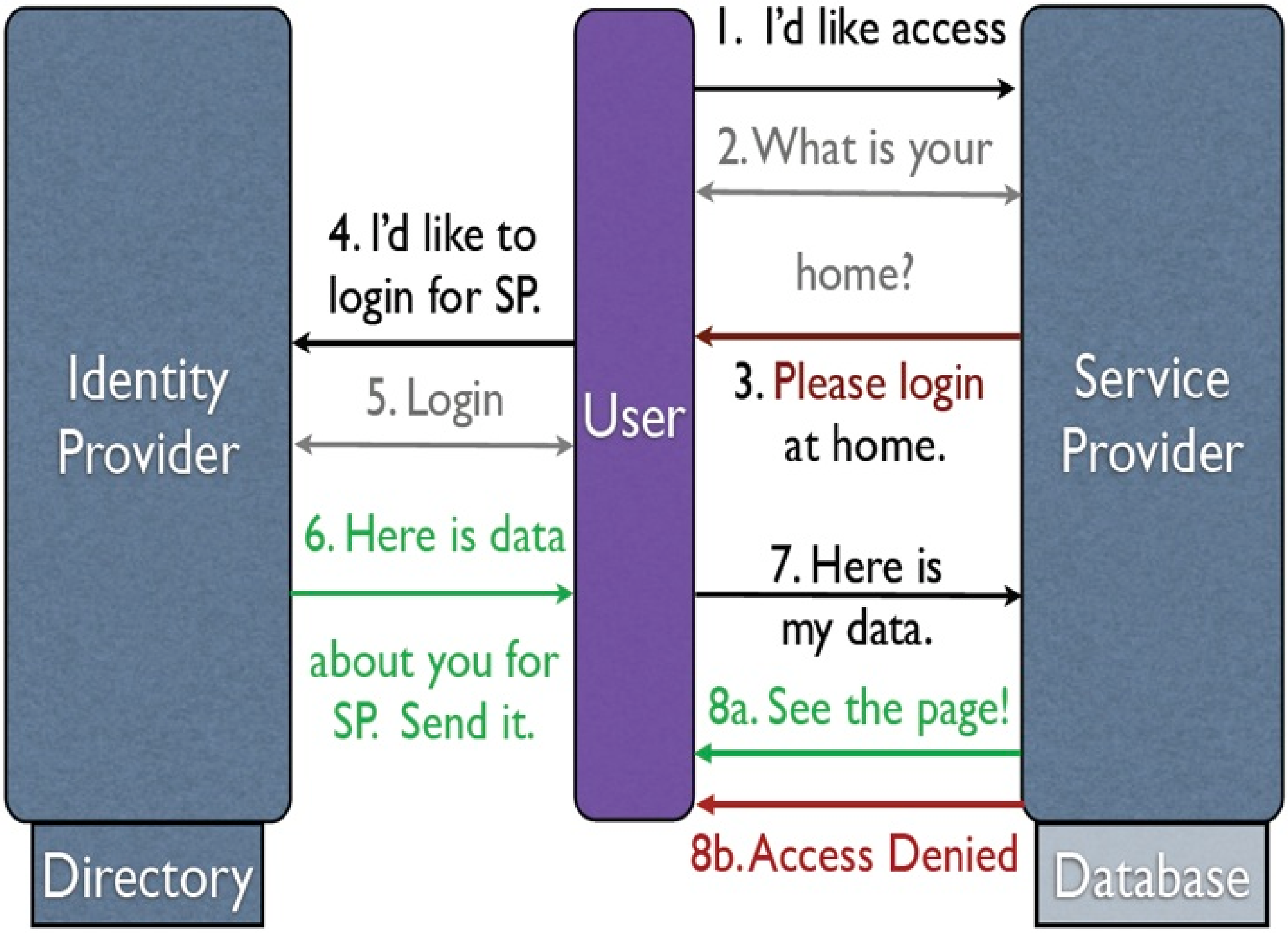
## Introduction

Internet2 NET+ offerings are built on top of identity services provided by participants in the InCommon Federation. InCommon supplies a secure common trust fabric and rules of the road to foster the sharing of services and user data by its membership. This document will help orient you with InCommon identity practices in preparation for becoming a NET+ provider.

InCommon identifies an authoritative identity provider (IdP) for each organization and will supply that organization with the information they need in order to connect to your service provider (SP). These IdPs are responsible for providing user data directly to you in a bilateral transaction that utilizes InCommon as substrate.

A federated identity transaction typically starts when a user is sent to their home IdP to authenticate, usually after attempting to access your application. The user authenticates as necessary after which user attributes are gathered about them. This information is packaged in a signed, encrypted SAML V2.0 assertion. These assertions are short-lived tokens generated dynamically each time a user wants to create a session at a service provider. Each assertion conveys authentication information and one or more attributes about the authenticated user.

The assertion is delivered to your SP according to the SAML V2.0 Web Browser SSO profile. The assertion is then processed by the SP to be used by the application to fulfill its identity requirements: for example, to authenticate the user, to create a local representation of the user, to authorize the user, etc. After the processing concludes, the user is granted access to the service.



InCommon maintains a signed document with a list of naming and trust information about all participants in the federation known as federation metadata. Each entry in the list is either IdP metadata or SP metadata. IdPs and SPs can trust and communicate with one another after they've exchanged metadata. By loading InCommon metadata, your customers will implicitly load your SP's metadata, while by loading InCommon metadata, you will load your customers' IdP metadata.

Each IdP and each SP are identified by a unique string vetted by InCommon known as an `entityID`. URIs provide a unique namespace for allocation of the `entityID`. These URIs are typically URLs for most InCommon members, but some InCommon members use a URN. These strings are not necessarily resolvable nor parseable. NET+ or InCommon staff can help you select the right `entityID` for your SP.

Because every service is different, so too is every identity integration. This document is intended to be generally useful and it is subject to revision. Portions of the following guidance, such as non-web integration or bulk provisioning, may not apply to your service.

This guidance has been reviewed by the InCommon Technical Advisory Committee.

# 1. Discovery and Authentication

Before a user can obtain an assertion, the user has to access their home identity provider with a destination service in hand. This may happen through your service interacting with the user, or through your service providing customers with enough information to help users initiate login events. Some deployments may support both approaches in order to allow flexibility in application access.

There is no uniform, "right" solution for initiating login. Making the right choice involves weighing the most elegant integration approach against the potential disruption of any existing user base. Getting discovery wrong will result in stranded users who would like to login for your application but are unable to figure out how to do so.

## 1.1. No User Interaction

These methods of session initiation rely on specific, unique ingress points to designate the right identity provider to use.

There is impedance to federated user login if the application is accessed in some common ways: for example, users arriving directly from a search engine at a generic public-facing page may be unable to get to the right home IdP. The user must get to defined access points such as a campus portal – a facility that some schools don't aggressively maintain or promote for general use by the entire population – or starting at the right URL from a perfectly executed web search.

These tend to work for cloud applications that serve limited populations that interact only within an organization, but not for applications with very broad user bases or where users from different organizations intermingle.

**1.1.1.** Dedicated URLs: The NET+ service exposes different URLs for each customer, such as `https://university.netPlusPartner.com/` or `https://netPlusPartner.com/University`. These URLs trigger generation of an authentication request for the user to carry to the IdP configured for that URL.

**1.1.2.** Exposed Session Initiators: A NET+ service can also expose dedicated URLs for session initiation. The SP would make public a session initiation base, such as `https://sp.netPlusPartner.com/Shibboleth.sso/Login`. Links for each IdP are then built from that base, for example, `https://sp.netPlusPartner.com/Shibboleth.sso/Login?entityID=https://idp.university.edu/`. These complete session initiation links are then exposed to users by, for example, a school's portal. This partially decouples your URL space from your login mechanism, leaving you with more flexibility to change your deployment without breaking customer integrations. This is NET+'s preferred approach within this category.

**1.1.3.** IdP-Initiated: Identity providers are capable of initiating access by sending the user to the SP with an assertion and destination URL in hand. This session initiation mechanism operates entirely outside the SP. This is mostly used by simple SP implementations that are not capable of issuing authentication requests themselves. Supporting this method in exclusion of all others will usually create usability issues regardless of how well it's done. Users like to bookmark, web search, and so forth, and none of these activities will be possible for NET+ users of your site if you only support IdP-initiated SSO.

There isn't much practical difference between these methods. Exposed session initiators and dedicated URLs are slightly preferable because they increase deployment flexibility by limiting the need for the IdP or third parties to know much about the SP.

## 1.2. User Interaction

Applications that serve diverse user communities as a collective population or want to accommodate user access from a variety of ingress points need to allow users to login after the user has already arrived at the application. The only way to consistently send the user to the right IdP to authenticate is to ask the user.

**1.2.1.** Discovery Services (DS): A discovery service exposes an interface to users by which they can choose a home IdP. Their selection is relayed to the SP, which then crafts an authentication request for the user to deliver to that IdP.

InCommon operates a discovery service for its members that you are welcome to utilize. This discovery service includes by default all members of InCommon. That list can be constrained by your SP to a subset of InCommon members who happen to be your customers; however, the list cannot be expanded by your SP to include customers who are not members of InCommon. If your user base is not InCommon or a subset thereof, you should probably not leverage InCommon's DS.

It is fairly trivial to run your own DS, but non-trivial to optimize the user experience. If you intend to operate your own discovery service, please familiarize yourself with some of the prior work done in this area and then run your ideas by the NET+ Identity team for additional feedback and suggestions. This is NET+'s preferred approach within this category.

**1.2.2.** Email Address: There is no good question to pose to a user to determine a user's home domain. However, most users will be able to respond to the question, "What's your email address?" The answer can be cleaved off at the @ and used to guess the user's IdP.

From InCommon's perspective, there are drawbacks to this approach:

- The approach of asking users to enter their email address on a page to begin a login process brings a significant security risk by paving the way for spear-phishing attacks.
- Users will often have multiple email addresses and exhibit confusion about which one they should use. Further, several schools either do not provide, or are contemplating not providing email addresses to students anymore.
- Users are confused about being prompted for their email address once and then prompted for a login name later, as they mentally tend to equate the two due to consumer site login practices.
- Users are confused by the lack of a password box alongside the email address, as they're not used to being prompted for just email address when initiating login. If a password box does exist, for example, to support local authentication as well with one interface, users will regularly enter their institutional passwords in that password box. This is another phishing hazard.
- The mapping of domains to IdPs is not bulletproof, and can be tricky. InCommon's identity techniques use organizations and IdPs as units of exchange. Accurately coupling domain names to InCommon registration and your set of customers becomes your problem to solve.
- Privacy is lost by prompting the user for their email address.

If you do need to use email-based discovery, consider borrowing ideas from some of the slicker implementations that have been done. (use abc@microsoft.com if you don't have federated access to trigger the behavior)

**1.2.3.** Buttons: Most consumer federated identity interactions are initiated by use of a branded button. This approach scales poorly to large numbers of customers or collaborators. We recommend against it in any deployment that must operate at scales greater than 6 IdPs.

## 1.3. Placement of Federated Identity Next to Local Authentication

Some NET+ deployments have existing user bases that authenticate using a username/email and password dialogue that is either embedded in pages or referred to via a login link. When the IdP discovery mechanism involves the SP interacting with the user, providing an adequate user experience for federated users with minimal disruption of local authentication takes some thought. For discovery mechanisms that don't interact with the user, the challenge is dealing with federated users that show up at your service directly and unauthenticated; your application's existing authentication interface generally doesn't need to be modified beyond adding some breadcrumbs to help these users get logged in.

For discovery mechanisms that do involve user interaction, from a federated identity perspective, it's great if you're able to treat the existing user base as just another federated user community that uses an IdP you operate for them. This ensures that discovery is the initial user interface when a user clicks "Login", maximizing the chances that federated users get back to their home IdP safe and sound. However, this is a confusing and additional step for local users, and often an unacceptable modification.

Next-best is placing local login behind a separate link. In this case, after requesting to login, the user will be presented with the option of federated identity or local identity. If the user selects local login, they will be presented with a username and password dialogue. If the user selects federated identity, they will be presented with a discovery interface from which to select a home IdP. The major challenge with this integration approach is presenting local login and federated login in a manner that users can successfully interpret, and there isn't much knowledge about how it's best done, beyond providing ways for users to backtrack if they select the wrong one.

A link to initiate federated login that is placed next to a username and password dialogue for local login is highly undesirable. This will routinely confuse and inadvertently phish users as they enter their home IdP username and password into the dialogue presented. This authentication attempt fails, of course, resulting in further user confusion. We ask NET+ vendors to please avoid this approach.

## 1.4. Authentication Quality and Levels of Assurance

When a user authenticates for your application, you can have confidence knowing that they used the same credentials that they use for other important services. Schools tend to do a good job of credentialing and authenticating users. Each IdP in InCommon is responsible for providing a statement describing how they perform a number of functions relevant to identity management in their Participant Operating Practices (POP) statement.

Level of Assurance (LoA) is another indicator that may be used to gauge the quality of credentialing and an authentication event: how certain you are that this user is who you think they are.

Qualification to assert authentication at a given LoA requires IdPs to meet a set of requirements defined by the InCommon Assurance Program. The InCommon Assurance Program is an Approved Federal Trust Framework under the Federal Identity Credential and Access Management Program and offers Federal-comparable profiles at NIST Level of Assurance 1 and 2, called Bronze and Silver respectively.

Unless your application is dealing with significantly sensitive information, you should maximize your customer base by accepting identity data whenever workable.

# 2. Attributes and Privileges

There is wide variation in attribute and privilege management capabilities amongst InCommon members. The best applications will accommodate this by offering multiple mechanisms to manage user information, privileges and provisioning.

Please bear in mind that your attribute requests are just that: requests. Every IdP is the ultimate arbiter of which attributes it will release to each SP. You should architect your application to require as few attributes as possible, and support as many attributes as practical.

## 2.1. Identifying User Organizations

Most SPs will expose the IdP's `entityID` to the application. This is the canonical way to know which IdP you're speaking with. The IdP `entityID` may be cross-referenced against the InCommon metadata in order to discover an organization name.

The flexibility of the InCommon identity system allows for a variety of IdP hosting scenarios. In the typical case, a single IdP is owned by, operated by, and authoritative for a single organization. However, some IdPs are authoritative for multiple organizations(e.g. school systems aggregated under a single identity management platform).

There is currently one additional indicator to help you ensure you're granting access to users from the intended organization. Some identity attributes include a domain context within which they are valid, a term known as "scope". For example, while `eduPersonAffiliation` may denote that a user is "staff", `eduPersonScopedAffiliation` conveys that the user is "staff@internet2.edu," a staff member of Internet2. If you are using Shibboleth software and InCommon, it will automatically validate that the asserting IdP is authoritative for the scopes of any scoped attributes received.

## 2.2. User Identifiers

InCommon IdPs can supply a variety of user identifiers which can be sorted into three main flavors: unique names, pseudonyms, and anonyms. A unique name takes a variety of forms but will uniquely identify a user. Pseudonyms, commonly referred to as "persistentId" or "targetedId" in shorthand, are opaque names for a user that persist as a user accesses your service time and time again. These names preserve a user's privacy through pseudonymity and correlation prevention. Anonyms, the default subject identifiers sent by most InCommon IdPs, grant anonymity by being uniquely minted for each session.

## 2.3. Commonly Available Attributes

InCommon participants will generally be capable of supplying user data that is stored in either eduPerson or inetOrgPerson (and structural components). Some attributes are defined in SAML 2.0 itself.

In practice, some attributes in the aforementioned object classes are more widely used than others. The most common attributes and loose definitions for them follow, ordered by privacy implications and, hence, usage patterns. The SAML 2.0 names for these attributes differ from these friendly names and can be viewed by hovering over the friendly name. Most SAML software will map the formal SAML 2.0 name to a friendly name for use by your application. Some of these attributes can be multivalued.

**Personally Identifying Information (PII)**: These attributes may identify an individual and all rarely, but can, change. `sn`, `givenName`, and `displayName` are not necessarily unique to an individual and change with greater frequency.

: A unique identifier for the user that is usually human-legible and is of the form userid@domain. Reassignment is permitted and reassignment practices vary substantially throughout InCommon.
: An email address for the user. Please don't use mail as an identity key, as it may be reassigned, frequently changed, or in a third party domain.
: A user's surname.
: A user's first (and second, third, etc.) name.
: A display name for a user.

NET+ recommends against the use of `cn` in favor of `givenName` and/or `displayName` because of the many and various interpretations and uses of `cn`.

**Pseudonyms**: These attributes are guaranteed to persistently-but-opaquely identify a user.

(deprecated name,
: A persistent, unique, opaque, non-reassignable identifier for the user. This identifier is generated once for each (user + IdP + SP), providing excellent privacy protection. Two drawbacks associated with the identifier are the lack of human legibility and relatively inconsistent support for it within InCommon. This should be supported as a primary identifier for your service, particularly for schools that feel uncomfortable supplying personally identifying information.

**Non-Personally Identifying Information**: These attributes help categorize users, may change over time, and are, except for rare situations with tiny subject sets, not personally identifying.

: A controlled vocabulary of attribute values that describe a user's affiliation with the school. Strong common definitions for these values don't exist because of heterogeneous definitions across higher ed. Schools are asked to populate it appropriately. Values student, staff, faculty, member, and library-walk-in are most commonly useful for NET+ services. Commonly multi-valued for some users.
: Of the user's `eduPersonAffiliation` values, this is the primary association with the organization. Less frequently used than `eduPersonAffiliation`.
: Same as `eduPersonAffiliation`, but scoped to a domain. This is often useful if a service is available to a specific community, for example, student@school.edu.
: An attribute that carries zero or more strings, each identifying either privileges a user has or groups a user is a member of. This allows IdPs to manage privileges with respect to your service in their native identity management infrastructure. Values should be URIs(either URL's or URN's) to ensure names don't collide, such as `https://netplusservice.org/attributes/isValidUser`. URI's don't need to resolve to something, although explanatory material could be placed at that location.

### 2.4. Custom Attributes

InCommon participants can define custom attributes when necessary. While the attributes can convey any data desired, including rich XML structure or base64-encoded objects, most custom attributes carry a simple string as a value.

The use of custom attributes raises the barrier to entry for customers who wish to use your service. Please use existing, widely supported attributes whenever feasible.

If you decide a specific custom attribute is necessary, ensure that the attribute is defined with a unique namespace (URL preferred) controlled by an organization that will be custodian for that attribute. Concisely and precisely define the attribute's purpose and value space.

We suggest consulting with the NET+ Identity team before requiring a custom attribute.

### 2.5. Privileges

InCommon members vary widely in their approach to management of privileges and permissions. Some schools would like to manage as many application privileges as possible centrally to leverage privilege management tools, consolidate privilege sets across applications, and improve auditing. Other schools intend to manage privileges entirely within applications or don't have centralized privilege management infrastructure yet. The ideal integration will accommodate this diversity.

**2.5.1.** The preferred approach to privilege management offers, in parallel, a privilege management interface integrated directly into your service and the ability to accept attributes that represent privileges. Default attribute name/value pairs and the privileges they represent can be defined. However, because of variability among campuses, in an optimal implementation, which attributes yield which privileges is configurable by each customer. A simple rules engine matches configured names and values to zero or more roles that are meaningful within your application. This configuration is the customer's responsibility.

XACML is not typically used within higher education to represent federated privileges.

**2.5.2.** A simplified implementation of the preferred approach using static attribute/privilege mappings is second-best.

**2.5.3.** A basic implementation will handle all privilege management within the application. This can create scalability and manageability issues for schools with large, heterogenous user bases.

## 3. Provisioning

Storing as little user information as reasonable within an application is an important NET+ design principle.

It's ideal if your application can provide full service to federated users without persisting any local user representation. If your application can do so, then you don't need any provisioning and you can move to the next section.

Provisioning for applications that need it can be roughly distinguished using the following question for demarcation: "Do you need to have a local representation of a user at any point before the user has accessed the application?"

If the answer to this second question is, "no, I don't need user data prior to that user authenticating," then it's recommended that your application implement dynamic, front-channel provisioning. This preserves privacy and limits the proliferation of user data, protecting users, schools, and you. Applications that must know users in advance will require back-channel provisioning.

### 3.1. Front-channel Provisioning

A typical front-channel provisioning workflow starts with the arrival of the user carrying a valid assertion from an IdP. The application will check to see whether the assertion contains a primary identifier associated with a known user.

If the user is recognized, the local user record is updated with information from the assertion if necessary, and the application uses either local or combined federated+local user data to grant the user access.

If the user is unrecognized, then a new user record is created using an identifier from the assertion as a key. After the record is created, the user is granted access.

### 3.2. Back-channel Provisioning

Back-channel provisioning is complicated by the lack of a successful, widely adopted standard. There are initiatives underway to change this status quo, such as SCIM and SPML. There are also successful standards in some education products, such as LTI, that may be usable for you.

The most common current approach is the implementation of custom provisioning APIs or web services. A back-channel provisioning interface should support both singular and bulk operations. Wholesale data transfers using LDIFs or CSV files are also not uncommon. NET+ prefers vendors to use standards whenever feasible and will encourage use of emerging standards as they are finalized and adopted.

### 3.3. Deprovisioning

When user information is stored by applications, it generally needs to be purged from time to time, a process known in the identity world as deprovisioning. Deprovisioning in a technical sense generally involves either deleting or decommissioning and archiving user data. Whether to delete or archive depends on the terms under which the application is offered and the nature of the user data, so we can offer no general guidance.

Federated user deprovisioning can either be triggered by aging, e.g. removing users that haven't been seen in a given amount of time, or by exposing interfaces to the customer to allow the customer to remove users.

Aging is a process of removing all users that have not accessed a service within a given period of time, e.g. the last 6 months. This shares the same challenges as provisioning in a federated environment, compounded by the fact that while attempt to access can be interpreted by an application as an attempt to provision, lack of recent access doesn't always imply an attempt to deprovision.

Exposing a deprovisioning interface, like back-channel provisioning, is hampered by the lack of good standards. Most deprovisioning interfaces are also custom and are frequently implemented alongside the provisioning interface.

## 4. Identity Event Data Collection

Each participant in a federated world possesses only a fragment of the complete picture of an event. Multiple participants, working in concert, can formulate complete pictures of most events. That makes federated identity event response a team sport in many circumstances. By contrast, as a collaborative framework, InCommon will rarely be involved and usually can't resolve a problem you may be having with any given provider.

Your customers may have specific audit, data retention, or incident response requirements that they need your help meeting, such as HIPAA and HIPAA BAAs. Work with them to be proactively aware of these.

Needs will vary by service and identity provider, but here are some typically useful data points.

### 4.1. Error Handling

You will eventually get bad data from an identity provider. This can be anything from insufficient attributes to syntax to a timestamp from the future. You may have errors of your own, too.

If your application can function anyway, it's generally appropriate to do so. If not, users need to be routed to the right resolution point for the problem with clear messaging, basic logs, and preferably an indexical reference to detailed logs.

When possible, add customization points to error displays for the customer to provide specific guidance to the user.

Using SAML software well, as described in Section 7, will make it much easier for you to do this. Poor error handling and students with free time can affect opinions of InCommon, SAML software, and your organization.

### 4.2. Auditing

Services should retain sufficient data to respond to audit-style questions. Typical questions are "how many users can use this service", "who has used your service", or "which users have privilege A". Specifics will be highly application dependent. Intuition is a guide, but some campus audit requirements are non-intuitive, so please be aware of the data you will need to retain proactively.

### 4.3. Event Response

Basic specific details should be retained about any identity transaction. These transactions are typically inbound SAML responses and mappings from SAML to your application, but may include outbound SAML requests.

For relevant messages, please retain at least specific timestamps, identifiers received, basic trust information, and IP addresses.

## 5. Logout

There is very little support for any form of federated- or single-logout capability within the InCommon Federation because of inherent challenges in web logout that are exacerbated by federation. Some schools expose a logout location to which you can redirect a user in order to clear their IdP session; others maintain a common page to display to users after logout occurs.

**5.1.1**: Exposing a configurable option that redirects the user (upon completion of local logout at the SP/application) to a URL of the customer's choice is a best practice.

**5.1.2**: Most applications consider logout to be a local process that terminates in a warning message to users asking them to close their browsers if they want to finish logging out.

This is an evolving aspect of InCommon and additional functionality may be implemented in the future.

## 6. Non-Browser Access

Many NET+ applications interact with customers entirely through web-based access, for example, HTTPS via a web browser. If your application interacts with customers in other ways, you will need to do more extensive integration work.

It's difficult to make specific recommendations due to the great variety of protocols, APIs, and applications in the world. As long as your interfaces are only intended to be used by clients that you develop, then you can pick the most appropriate integration for your application. NET+ has no strong preferences here, but this may change with time in the event that application interfaces become more cosmopolitan and applications are developed that work with a suite of NET+ services.

Please review the below guidelines and then discuss proposed solutions with the NET+ identity team, especially as this is an area of active standards work.

### 6.1. Non-Browser HTTPS

SAML 2.0 includes a profile called ECP designed to make it easy for native clients to communicate directly with the IdP over HTTPS to authenticate and obtain SAML assertions. These assertions are then played to a service provider or directly to the application and used to bootstrap into another session persistence mechanism.

This requires some implementation work to modify the client to use ECP, and some work to ensure your server can handle it as well. Examples are available.

ECP support is not universal amongst InCommon sites, but enabling it is typically not difficult for the IdP administrators.

OAuth 2.0 is a template explicitly architected for HTTP that can be used to build a protocol for obtaining and playing access tokens. OAuth supports a variety of token formats, including SAML 2.0 assertions and custom tokens.

### 6.2. Bridges and Other Protocols

If a client can be invoked by or integrate with a web browser, it's possible to bootstrap from the web session and typically short-lived SAML assertion into a separate session. This new session is persisted using an application-appropriate mechanism independent of any web browser cookie.

These integrations typically work via either extending or misappropriating fields in an existing protocol.

For example, a bridge into FTP could work by populating a database at the FTP server with nonces for usernames and a common password and supplying those nonces to clients invoked through the federated identity interface. Alternatively, a SAML assertion could be sent somewhere in the FTP protocol itself to a server in the know that could process it. The target protocol and your imagination are the only real limits.

New or custom protocols can bridge more elegantly by deliberately architecting federated identity support into the protocol.

Extensions to the IdP and custom bindings can be developed for native support of protocols, but in most cases this is not advisable because of development costs and the increase in deployment burden for your IdP partners.

## 7. Implementation

A preferred service validation process is outlined on the front page.

NET+ recommends the use of SAML 2.0 software that is compatible with the Interoperable SAML 2.0 Profile, such as Shibboleth or simpleSAMLphp, both of which are free and open source. Corporations can provide consulting services for these and other software products.

There are many commercial SAML 2.0 implementations, including ones from most major vendors, as well, but these may lack attribute or metadata features that your customers greatly value.

NET+ strongly discourages vendors from implementing SAML 2.0 themselves due to the inherent challenges in architecting and maintaining secure identity software.

Your development SP should be registered with InCommon to facilitate testing.

### 7.1. Imagine, For a Moment

So far, we've only thought about how to plug federated identity into an existing application. However, new possibilities are sometimes unlocked in the process.

Can you do anything new with authoritative user data? Can you leverage SSO to integrate better with the rest of a customer's infrastructure? How can this technology benefit corporate customers, too? Is there an opportunity for users, perhaps from different organizations, to collaborate in new ways?

### 7.2. Testing with an IdP

During the implementation process, it can be helpful to test against a known, functioning IdP where you have an account. A simple public service for testing new providers with a Shibboleth implementation is hosted by TestShib.

Once you're sure that your SP is functional, it's beneficial to set up and operate your own internal testing IdP so you can exercise specific aspects of your integration, including failure scenarios and user experiences.

### 7.3. Join InCommon

In order to join InCommon, please follow the processes described on the Join InCommon page. One of the organizations that is sponsoring your NET+ service validation will also sponsor your membership in InCommon.

InCommon aggregates metadata from individual providers and regularly publishes it as trusted, signed data. As one of the steps on the Join InCommon page, you will supply InCommon with the information needed to publish metadata describing your SP using the InCommon administrative interface.

You will also need to be able to consume the metadata supplied by IdPs published by InCommon on a regular basis.

Working with a new organization then involves little configuration work for both you and that IdP.

The IdP:

- Loads and trusts the metadata of the SP, which will happen automatically upon your acceptance into InCommon; and
- Configures the proper attribute release to your SP depending on your SP's requirements.

The SP:

- Will load and trust the metadata of the IdP, which will happen automatically when you trust the InCommon metadata;
- Ensures that discovery mechanisms provide a way for the IdP's users to find their way home; and
- Performs application-specific integration as necessary, e.g. linking a federated IdP to an internally identified customer environment.

After these steps have been performed, you should ensure that the authentication process is successful for a user from the organization's IdP before considering the integration successful.

## 7.4. Double-check with NET+

After you've parsed this document and, hopefully, constructed an integration plan for your application, we ask you to vet your thinking with the NET+ team before getting implementation underway. We'll include other participants, especially schools, in this process whenever beneficial or necessary.

## 7.5. Service Validation with Schools

After you've finished your implementation work and tested your SP successfully against a dummy IdP, it's finally time to continue with the schools that sponsored your participation in NET+ for real testing and service validation. Your sponsors determine precise requirements and exercise ultimate control over the validation process and its outcome to ensure their needs are met. Integration choices and roadmap, documentation quality, and other factors are frequently considered.

NET+, as a program of InCommon and Internet2, works in collaborative frameworks to ensure that identity requirements expressed by the service and the schools are met and ongoing improvements, if needed, are achieved.