# OpenCCC Identity Management

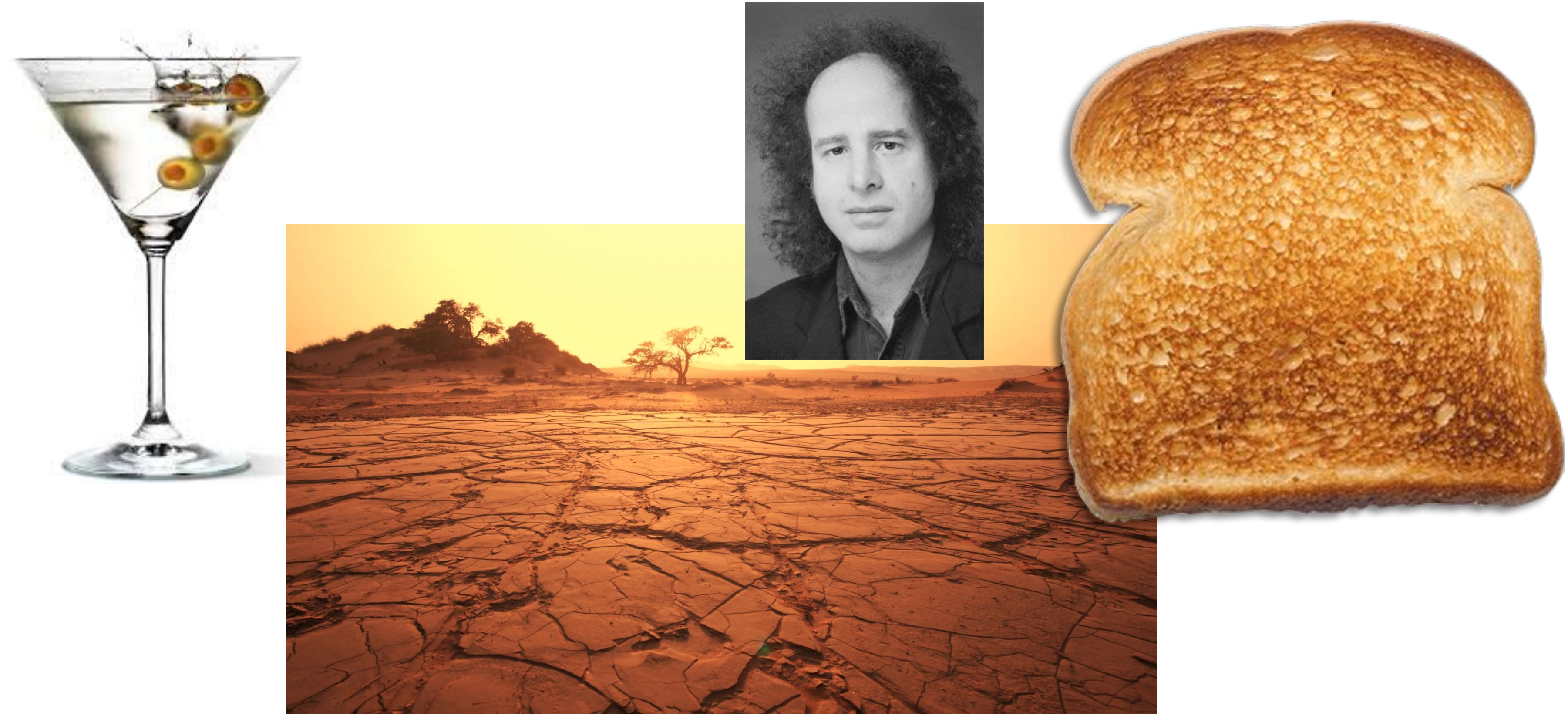
# About Me

- Charles Hasegawa, Software Developer with Unicon, Inc
- Software developer for 17+ years
- Technical Lead for the CCC MyPath Project

# Topics

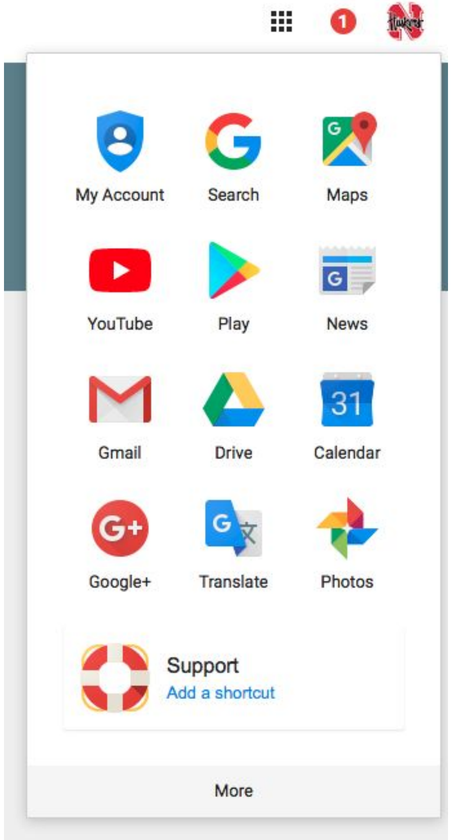
- SSO vs Federated Identity
- Authentication and Authorization
- SAML
- Identity providers and Service Providers
- Intro to CCC SSO Proxy

# What’s the common thing?



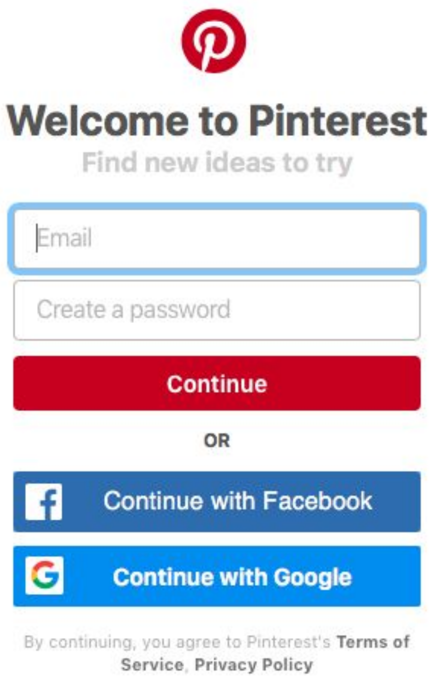
# What is Single Sign On?

- SSO, is short for Single Sign-On.
- SSO is the principle of delegating application authentication to a **single, trusted source**. This authentication is reusable as a user moves between different related applications.
- Without SSO, users have to log into every application, every time, even if it is the same username and password.
- With SSO, you can bring all of your organization's applications under a single login page (great for auditability, credential maintenance, etc)
- Think: your applications, your users...
- Think of it like a library card or work badge - it'll get you into any of the buildings, access to the cafeteria, the gym, etc. A specific ID that is valid for a limited scope of things.



# What is Federation?

- Federation is the principal of authenticating your users to 3rd party applications, OR allowing 3rd parties to authenticate their users to access your applications.
- **Each home organizations manages their own users' credentials.**
- Without federation, your users are creating accounts all over the place and you generally have no ability to help them with those accounts.
- With federation, you'd no longer need to create accounts for partners. When a user leaves, you no longer have to find all the partner systems where they have special rights.
- Examples: Typical OAUTH providers: Facebook Connect, Login with Google Plus, Login with Twitter, etc.
- Real life federated ID - Passports. Each country is a different issuer of standardized, trusted information



Welcome to Pinterest

Find new ideas to try

By continuing, you agree to Pinterest's **Terms of Service**, **Privacy Policy**

# Federated Identity Protocols

- **SAML 2.0**: very widely deployed in the enterprise world
  - **S**ecurity **A**ssertion **M**arkup **L**anguage
  - Markup Language means a form of XML
  - Widespread interoperability, but inconsistent quality (ie lots of things support SAML, but do so poorly)
  - SAML 1.1/Shibboleth 1.x
- **OAUTH 2.0**: Not really an interoperable protocol, more like of framework
- **OpenID Connect**: An OAUTH 2.0 profile; Basically SAML, but using JSON... probably the future.
- IMI/Infocard, WS-*, Liberty Alliance, etc.: “Feature Complete”, but not really used.

# SAML Response Payload

```
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" ID="_c8843789e9206297d3b67c19bc21c623"
IssueInstant="2014-05-29T19:38:20.811Z" Version="2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">https://ssodev.test.edu/</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
     <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI="#_c8843789e9206297d3b67c19bc21c623">
           <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                 <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="xs"/>
              </ds:Transform>
           </ds:Transforms>
           <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
           <ds:DigestValue>7TjFEONF009Xkq+rbDJLx1yCr3A=</ds:DigestValue>
        </ds:Reference>
     </ds:SignedInfo>
     <ds:SignatureValue><!--Signature--></ds:SignatureValue>
     <ds:KeyInfo>
        <ds:X509Data>
           <ds:X509Certificate><!--Cert Info--></ds:X509Certificate>
        </ds:X509Data>
     </ds:KeyInfo>
  </ds:Signature>
```

# JWT Token Payload

```
{
		sub: "pcollegeadmin@democollege.edu",
		 authSource: "CiMock",
		 azp: "94d6aded-be11-4da7-9b53-921796436fca",
		 eppn: "pcollegeadmin@democollege.edu",
		 roles: [
				"ROLE_USER",
				"ROLE_STAFF",
				"ROLE_PORTALCOLLEGEADMIN"
		 ],
		 iss: "https:\/\/login.ci.cccmypath.org\/f\/",
		 scopes: [
				"address",
				"phone",
				"eppn",
				"openid",
				"offline_access",
				"profile",
				"email"
		 ],
		 exp: 1520356529,
		 misCode: "ZZ1",
		 iat: 1520355929,
		 jti: "81e56fa1-ba65-4cee-aec1-b2b2d261f45c"
}.
[signature]
```

# SSO vs Federated ID

- SSO is really about **your** users and **your** applications, and using a long running authentication session to stay logged in.
- Federation is about sharing users and applications with 3rd parties while each owning organization manages the user's credentials.
- Federation is really a natural progression or extension of SSO.

Think how hard it would be to travel to another country if you had to apply for a new ID for each country you wanted to travel to - your passport allows you to move about.

# Authentication vs Authorization

- **Authentication**: validation of identity (you are who I say you are)

- **Authorization**: granting / denying permission to a resource or set of resources.

A student may have access to their own grades for the courses they are enrolled in, but they cannot see other student’s grades nor can they edit their own.

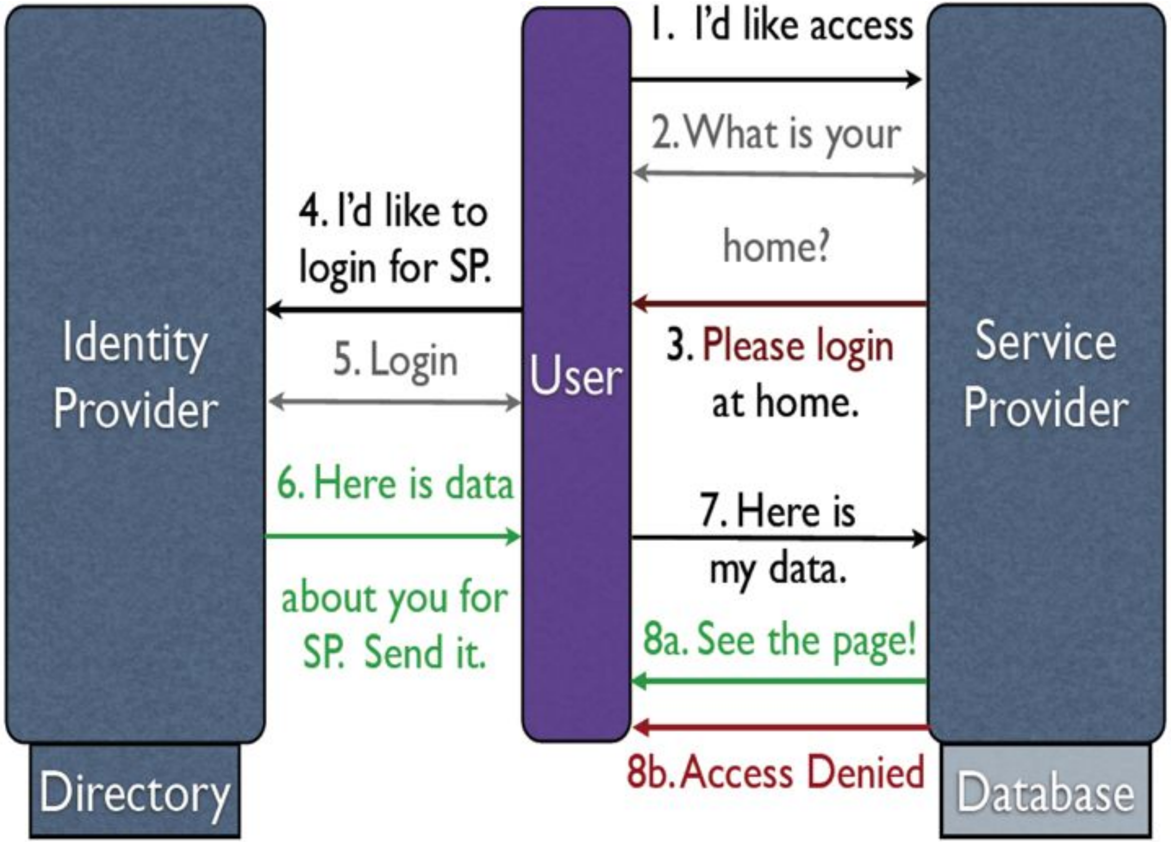
# Other Helpful Terms

- **Assertions or Claims**: facts that a client receives from a trusted source.
- **Attribute Release**: passing additional attributes to the client application along with the identity assertion.
- **Front Channel or Passive**: Actions that route through the user’s browser.
- **Back Channel or Direct**: Connections made directly between two services and don't involve the user/browser.

# Sessions

- **Application Session** - the length of time that a single application grants you access to its systems after you have authenticated.
- **SSO Session** - the length of time that your authentication source grants that your credentials are valid.

# SAML Walkthrough

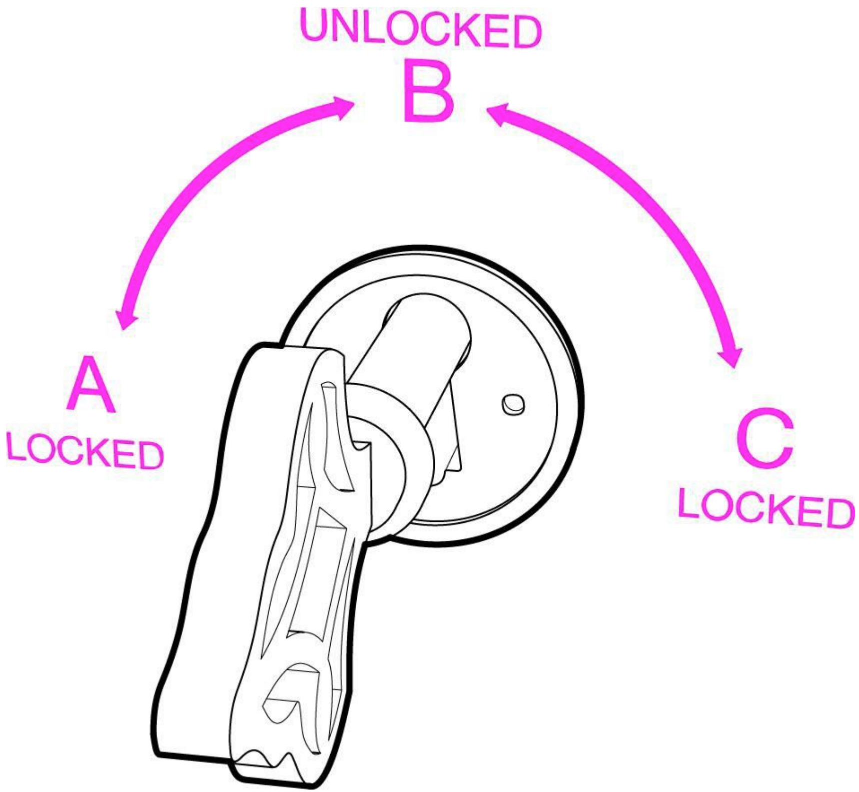


# Trusting the Metadata

- Metadata is used to allow IdPs and SPs to trust each other.
- Metadata is an XML that includes:
    - Signing/Encryption Public Keys
    - Endpoints (the valid place to handle the information)
    - Contact Info
    - Profiles supported

# Public and Private Keys (for non-geeks)

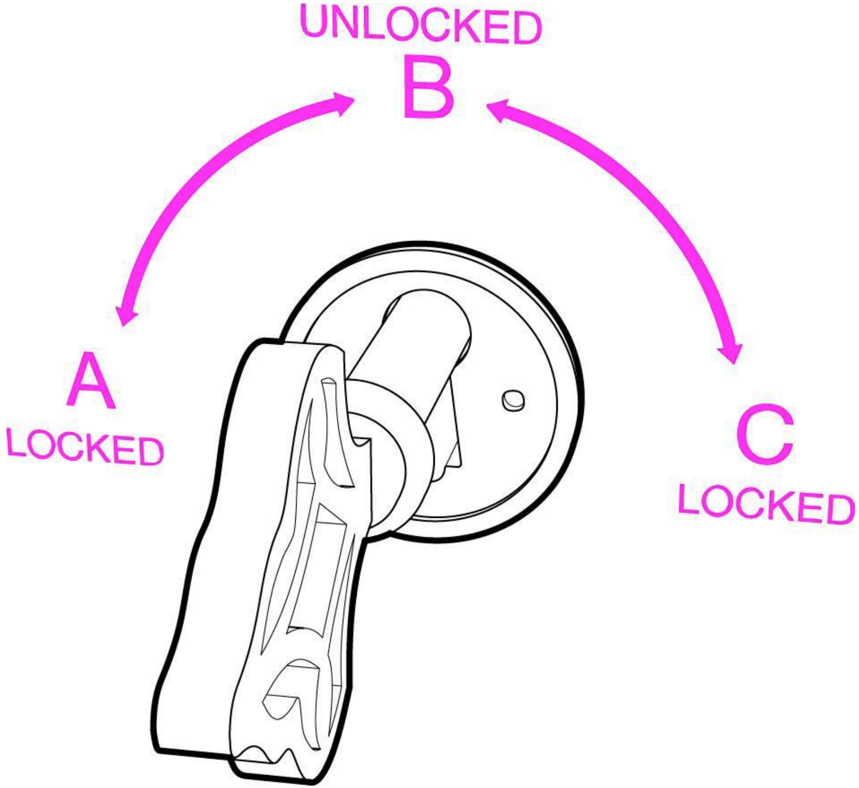
- Anna has a box with a special lock that has three states
- Two different keys work on the box
  - One turns clockwise (A-->B-->C)
  - One turns counterclockwise (C-->B-->A)
- Anna keeps the first (her private key)
- Anna makes a hundred copies of the second and gives them to everyone that she trusts to access her box (public key)



# Public and Private Keys (for non-geeks)

Imagine you need to leave Anna a very personal document. You put the document in the box and use a copy of her public key to lock it. Now the box is locked. The only key that can turn from A to B is Anna’s private key, the one she’s kept for herself.

*Remember, Anna’s public key only turns counterclockwise, so you turn it to position A.

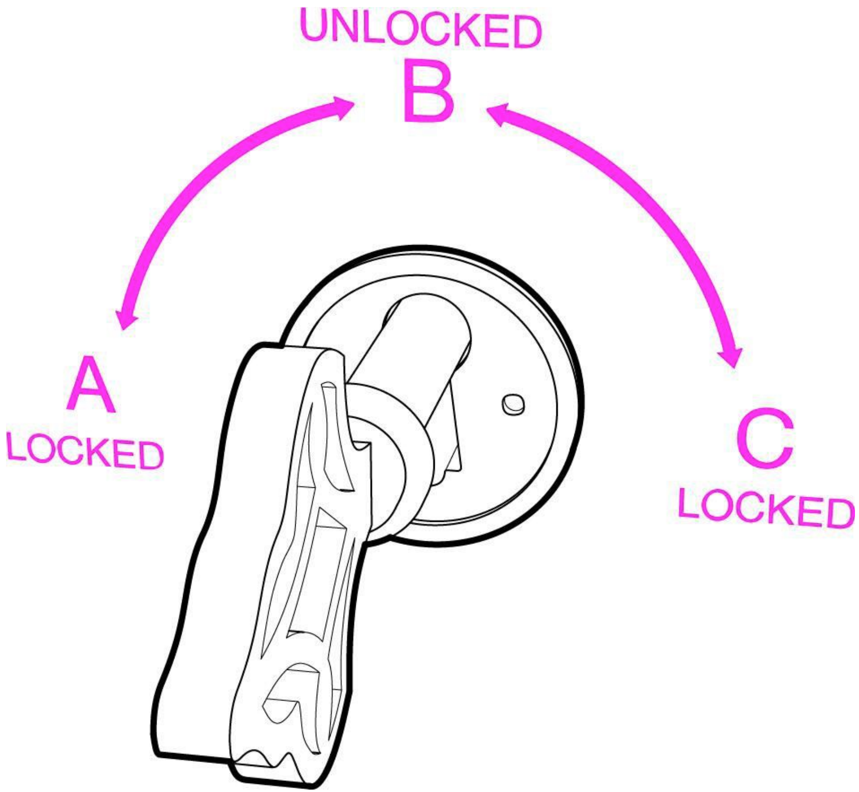


# Public and Private Keys (for non-geeks)

Suppose Anna puts a document in the box. And she uses her private key to lock the box, i.e. turn the key to position (C). Why would she do this? After all, anyone with her public key, can unlock it!

Someone delivers me this box and he says it's from Anna. I don't believe him, but I pick Anna's public key from the drawer where I keep all the public keys of my friends, and try it. I turn right, nothing. I turn left and the box opens! "Hmm", I think. "This can only mean one thing: the box was locked using Anna's private key, the one that only she has."

So, I'm sure that Anna, and no one else, put the documents in the box. This is what is meant by, **"digital signature"**.



# CCC SSO Proxy

- College applicants, students, staff and faculty will be using applications such as MyPath, Report Center, Hobsons and Canvas as well as a host of other CCC managed systems and external services.
- Each participating college will be required to provide a SAML compliant Identity Provider to authenticate their user population to these services.
- Makes adding SPs easy (only have to update the SSO Proxy)
- A central IDP proxy (SSO Proxy) will be invoked in order to route the students to their IDP and then to their target application. When moving to other Service Providers during the same session, requests will again be routed through the IDP proxy
- Service Providers will know which IDP the user came from (part of the released attributes)
- The SSO Proxy normalizes the attributes
    - Validate that student users have a CCCID
    - Ensure that attributes are in lower case
    - Ensure the minimum set of attributes were supplied (EPPN, eduPersonAffiliation(s), displayName)

# SSO Proxy - Ensure Student CCCID

- Determine if the incoming IDP assertion requires a CCCID
- Determine if there is already a CCCID associated with the asserted EPPN
- Find CCCID by having user login with their OpenCCC account

OR

- Create an OpenCCC account

# SSO Proxy - SLO

What about the digital trail of websites the user seamlessly accessed and left in their wake? The user didn’t have to log in manually, so it’s easy for them to be unaware that they have other sessions open. Afterall, authentication is only the first half of the story - end-users rarely log out of **each** session established during SSO.

The SAML protocol is a popular choice for enabling SSO and contains a protocol called SAML Single Logout (SLO). This additional protocol helps address the problem of orphaned logins. SLO allows a user to terminate all server sessions established via SAML SSO by initiating the logout process once. SLO is initiated from either the Identity Provider (IdP) or any of the involved Service Providers (SP).

SLO is however, not a panacea as it does have its own requirements and restrictions. For example, both parties (the IdP and the SPs) must support the SAML Single Logout protocol. The various logon sessions are not terminated if the SP does not support this protocol.

# SSO Proxy - SLO

The definition of SLO in the SAML specification is derided for being difficult to implement and fragile in the face of runtime errors or exceptions. The estimable folks that implemented the Shibboleth protocol wrote an illuminating round-up of their trials & tribulations when adding SLO to Shibboleth. These difficulties compound the fact that SSO vendors can view SAML single logout as a secondary feature that is often either poorly implemented or ignored and left out entirely.

A final consideration that must be addressed when contemplating SAML Single Logout is the varied nature of many SSO implementations. SAML SLO only works with SAML SSO installations (Such as SAML and Shibboleth), and will not integrate with other SSO protocols such as OAuth, CAS and WS-Federation. It is important to understand that in many SSO environments, a SAML Single Logout implementation will only mitigate issues of orphaned SAML sessions – not eliminate these issues entirely. As is the case when technology doesn’t adequately address an issue, the responsibility then falls to the end user. It’s easy to pass this off as a “user education” issue, but that rarely goes well.

Consequently, the best defense against orphaned logins and improper logout procedures is to require users to shut down all browser windows and tabs. However, policies and best practices tend towards subjectivity and administrative discretion as opposed to required end-user practices.

# References

- https://www.unicon.net/solutions/identity-and-access-management
- https://medium.com/@vrypan/explaining-public-key-cryptography-to-non-geeks-f0994b3c2d5
- https://wiki.shibboleth.net/confluence/display/CONCEPT/SLOIssues
- https://www.portalguard.com/blog/2016/06/20/saml-single-logout-need-to-know/