

Federated Identity: everything you need to know but were too afraid to ask

Paul Millar

LSDMA WP1, 2013-05-17



The basics: 1-side intro

- Non-Federated (“normal”) login needs user to authenticate with the service
 - Typically done with username+password, Kerberos or X.509 client certificate.
 - Maintaining database of user credentials requires non-negligible effort
 - Especially over a distributed environment.
 - Even more so when user have a short-term relationship with service (“high churn”)
 - Idea is to handle authentication **separately** from the service
 - User identifies themselves to an entity they have a long-term relationship (e.g., “home institute”)
 - This entity states (“asserts”) that the user proved their identity, plus (optionally) some information about the user.
 - Some general problem areas in Federated Identity to keep in mind:
 - Group-membership **not** asserted by IdP must be asserted by some 3rd party.
 - When to decommission state? When has this user “gone away” ... really?
 - Lots of legal(-ish) issues on **trust**: an IdP trusting a service and a service trusting an IdP.
 - Commonly available solutions are **web-only**, data interaction is typically non-web.
-

The basics: SAML web-profile

- Vast majority of Federated Identity is based on SAML web-profile
 - Here is a typical interaction:
 - Using web-browser, a user navigates to desired service's web-page
 - As not currently logged in, she sees a "LOGIN via Federation" style of button.
 - When selecting button, redirected to "Where Are You From" (WAYF) web page
 - On seeing a selection of potential home institutes, selects appropriate one
 - On seeing the home institute web-page, identifies herself
 - typically user-name and password, but can be any method (Kerberos, X.509, ...)
 - On successfully identifying herself, web-browser is redirected back to service, transporting a set of SAML assertions to the service in doing so.
 - Things to note:
 - Requires a web-browser (non-web client would need to fake being a web-browser),
 - Choices (e.g., which home institute) may be recorded in cookies (easier next time),
 - Login could be fully automatic if session with home institute is still active (recorded in cookie)
 - No opportunity for 3rd party assertions (the service might be able to fix this),
 - Service may be able to "ping" IdP to learn if user still exists (this doesn't scale, though)
 - Similar to other forms for federated identity: c.f. OpenID, BrowserID, OAuth,
-

Scenarios

- There are **lots** of technologies that may be used in a Federated Identity solution
 - Some have “multiple modes” that support different ways of operating; for example, the CILogin service may be used from a web-browser, from a custom client, using ECP and using OAuth.
 - The solutions space is complicated:
 - Most technologies can be made to work, one way or another.
 - Difficult to get an overview of how they compare.
 - Introducing **scenario** as a way to group together comparable technologies.
 - Allows us to break the decision process into two levels:
 - First, which scenario(s) do we need to support?
 - Second, which technologies to support these scenarios?
-

What is a scenario?

- Just a convenient way of grouping similar technologies and their usage
 - See similarities between technologies.
 - Understand some fundamentals of how they distinguish.
 - The answer to three questions determines the scenario:
 - How does the client communicate with the service?
 - Is it directly (**D**) or via some Proxy service (**P**). A web-portal is an example of a proxy service.
 - How is the user identified to the service?
 - Is the client identified by a client X.509 certificate (**X**) or as a set of SAML assertions (**S**)
 - Which software is the users using?
 - Is the user forced to use a web-browser (**W**) or is the process generic (**G**), so it can be, in principle, anything. To illustrate a generic solution, either an existing application is updated via a plugin or a CLI is available.
 - The answer to these three questions describes a scenario, summarised by three letters:
 - D-S-G (user in direct communication with server, identified by SAML assertions, in a generic fashion),
 - P-X-W (using a web-portal, which requires a web-browser to work, and identified to service with X.509).
-

Placing technologies:

Client connects...	User is identified to service via...	Client application is ...	Examples (†)
Direct	SAML assertions	Web-browser	SAML web profile, OAuth 2.
Direct	client X.509 cert.	Web-browser	CILogin, SLCS, Confusa + TCS.
Direct	client X.509 cert.	Generic	D-X-W + cert-from-browser(*), SLCS client (**), CILogin client(*), CILogin-ECP(***)
Direct	SAML assertions	Generic	Moonshot, GSS-ECP(***)
via Proxy	SAML assertions		<i>Current solutions are too insecure, as proxy needs to authenticate as user to IdP.</i>
via Proxy	client X.509 cert.	Web-browser	CILogin-OAuth, Upload cert to MyProxy & MyProxy-OAuth EMI-STS.
via Proxy	client X.509 cert.	Generic	D-X-G + X.509 delegation D-S-G + EMI-STS

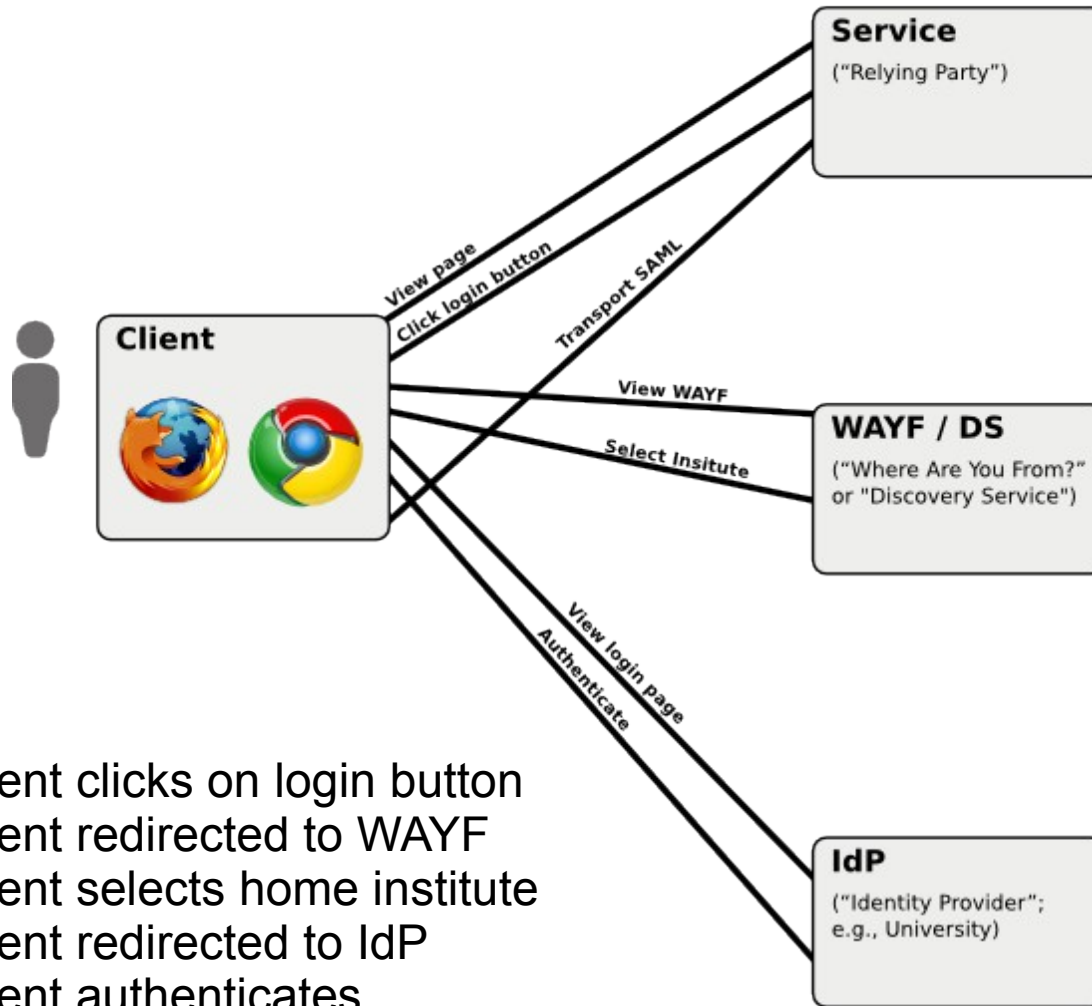
(†) Don't worry if you don't know all examples

(*) Require prior web-based interaction, so not completely generic.

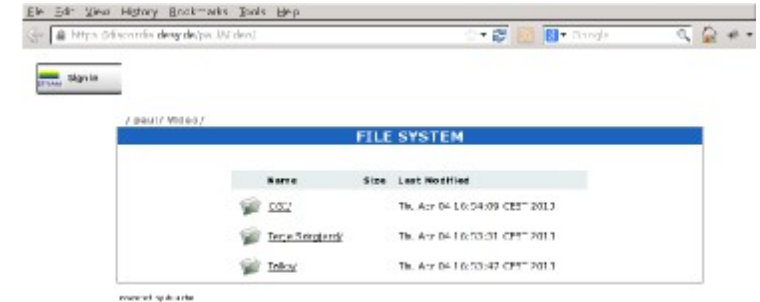
(**) The "SLCS client" option seems to be Shibboleth-specific.

(***) Requires IdP to allow ECP interaction, which is (currently) rarely.

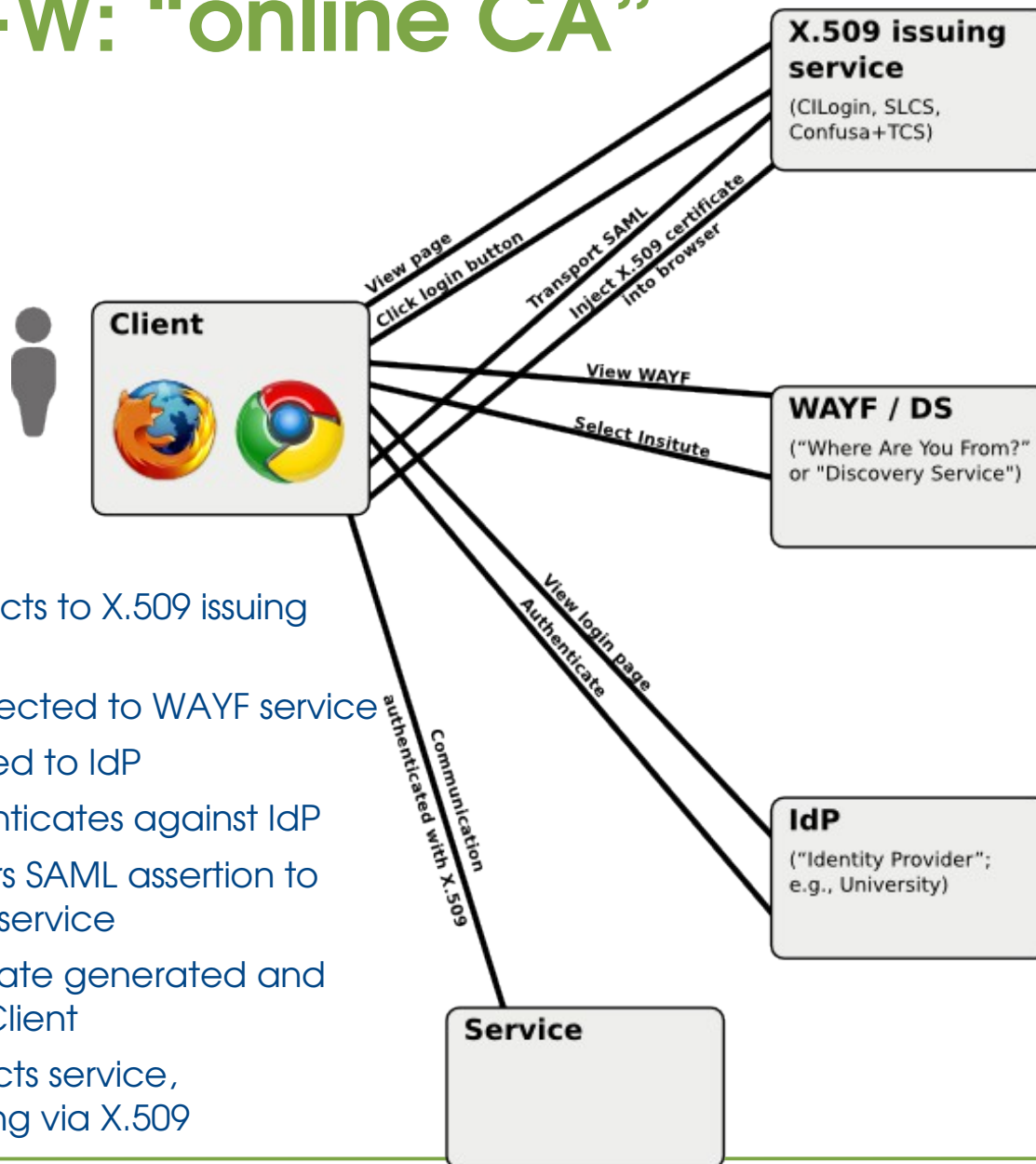
D-S-W: "SAML Web-Profile"



1. client clicks on login button
2. client redirected to WAYF
3. client selects home institute
4. client redirected to IdP
5. client authenticates
6. client redirected back to service, transporting SAML



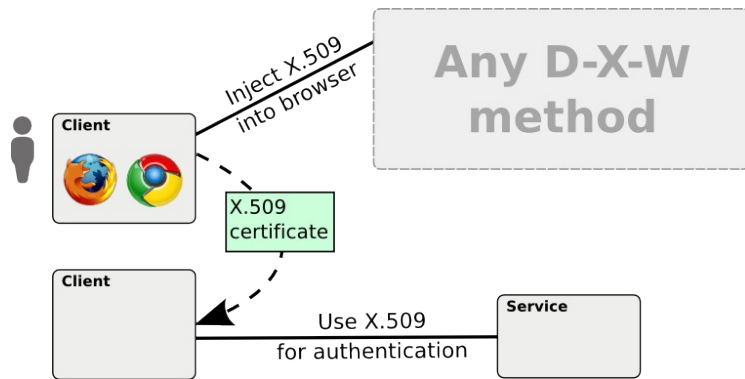
D-X-W: “online CA”



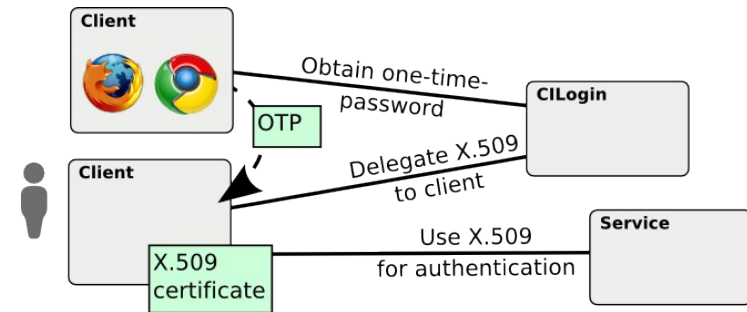
- Client connects to X.509 issuing service
- Client is redirected to WAYF service
- Client directed to IdP
- Client authenticates against IdP
- Client delivers SAML assertion to X.509 issuing service
- X.509 certificate generated and returned to Client
- Client contacts service, authenticating via X.509



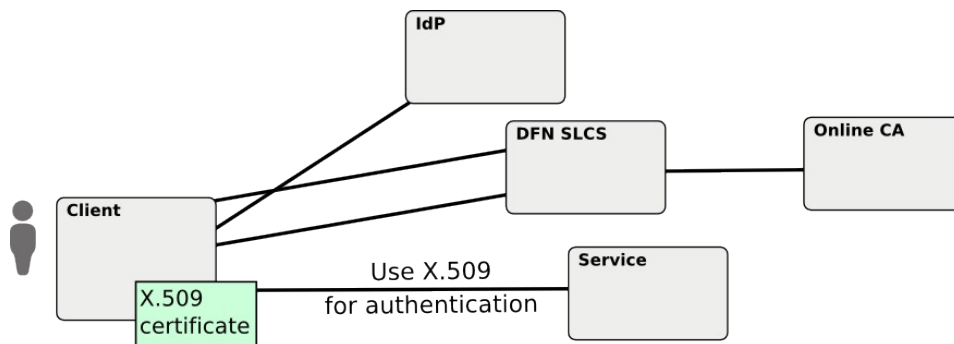
D-X-G: generic X.509



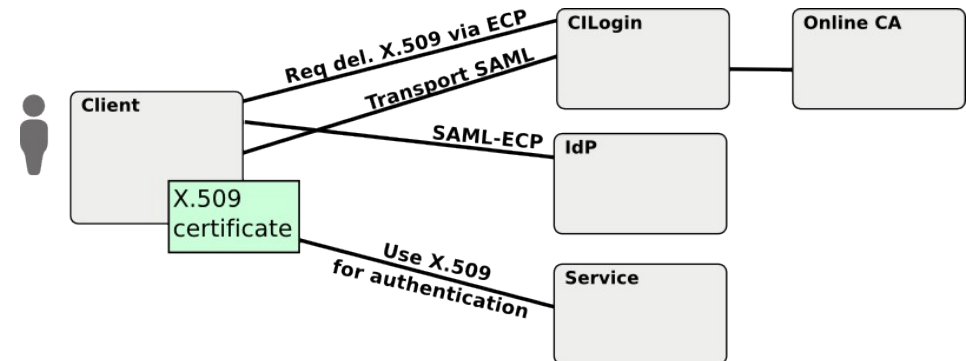
Cert from browser



CILogin client



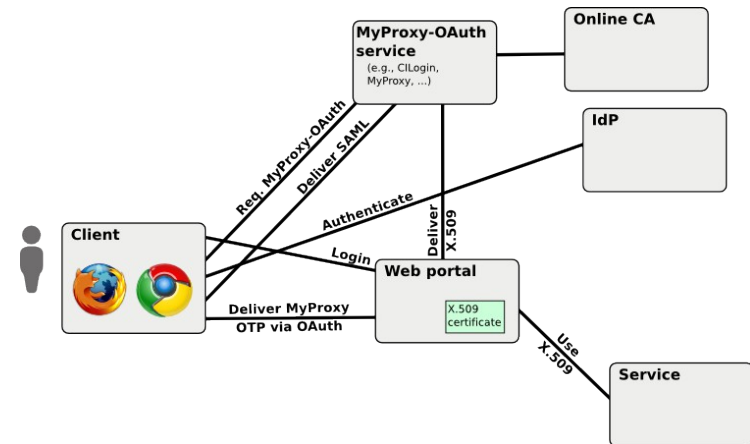
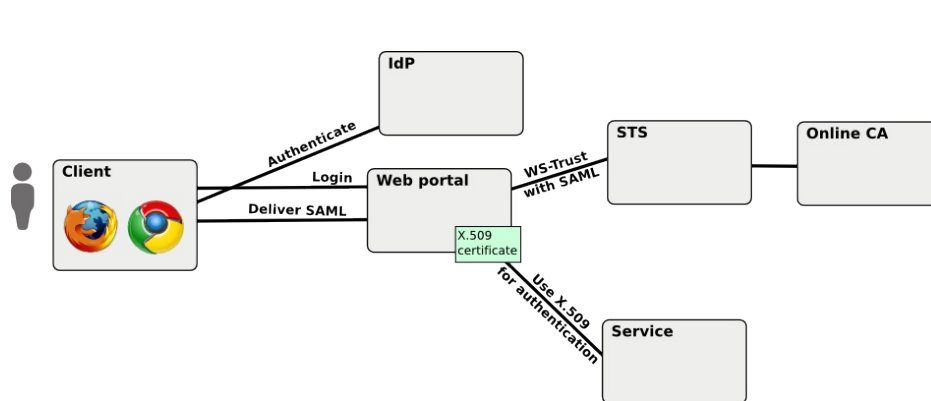
DFN SLCS



CILogin with ECP

P-S-^{*}: SAML Delegation

P-X-W: “web portal to the grid”



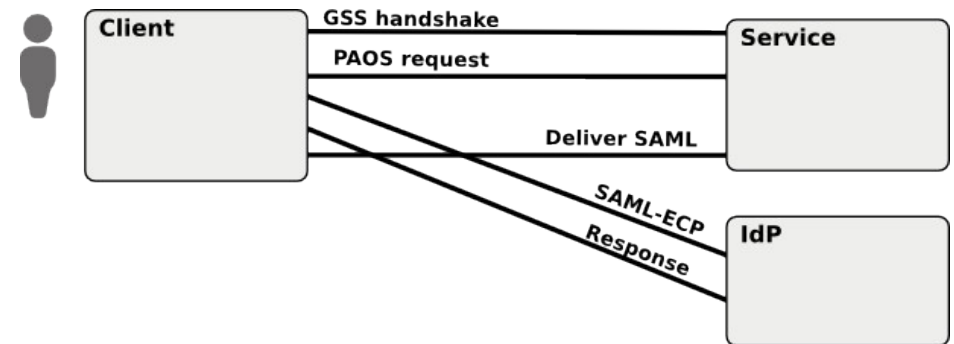
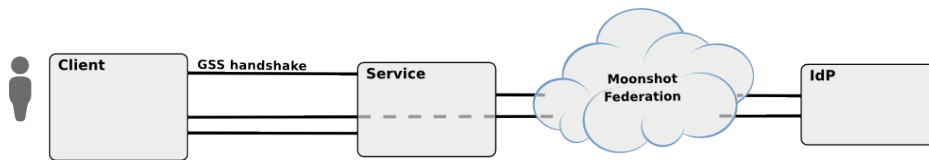
- SAML Web-profile to web-portal
- Use credential translations service (e.g., STS) to convert to X.509
- Use X.509 to contact service

- Web-portal redirects client to MyProxy-OAuth-like service
- Client logs into service using SAML Web-profile
- MyProxy-OAuth-like service translates SAML to X.509 certificate
- Client delivers MyProxy one-time password to Web-portal as 2nd leg of 3-leg OAuth
- Web portal receives delegated X.509 using OTP.

Globus Online is an example of a “web portal.” It can use the “MyProxy-OAuth” method (right diagram)

A web-portal might also drive a users work-flow: transferring data and starting compute as needed.

P-X-G: Generic proxy



- Client starts GSS handshake
- Service locates IdP through federation
- Service initiates login with IdP
- Service tunnels challenge/response between client and IdP
- Client delivers SAML to Service

- Client starts GSS handshake
- Service replies with PAOS ECP request
- Client forwards request to IdP, authenticating with the request
- IdP returns SAML assertion
- Client forwards SAML assertion to Service

Proposal: part a.

- Non-trivial fraction of our user community are already using Globus Online
 - Globus Online supports federated identity using the “MyProxy-OAuth” method, with CILogin
 - Demonstrator: we provide a MyProxy-OAuth service, which Globus Online uses to allow people to transfer files using their DFN-AAI, Umbrella, ... identity
-

Proposal: part b.

- FTP is commonly used transport for uploading and downloading files
 - Moonshot allows users to login with their institute credentials securely.
 - Demonstrator: at least two IdPs join moonshot pilot and at least two sites provide FTP services that allow data transfer, authenticated via moonshot.
-