Location Independent Collaboration In Higher Education Networks (LICHEN) Project

Application Oriented Authentication and Authorisation
for Virtual Organisations based on the UKERNA
Location Independent Networking (LIN) Infrastructure

Project Report 2a:
Proposal for Janet Roaming Service (JRS)-based Virtual Identity Provider (IdP) for Shibboleth

Editor: Tim Chown (University of Southampton)

Contributors:
Josh Howlett (University of Bristol)
Mark O'Leary (University of Manchester)
David Mills (University of Southampton)

Version 0.3

1st March 2006

The LICHEN project was funded by the Joint Information Systems Committee (JISC)

Contents

CONTENTS **2**

1 INTRODUCTION **3**

2 INTEGRATING SHIBBOLETH, THE JRS AND LICHEN **3**

2.1 JRS COMPONENTS 3
2.2 SHIBBOLETH COMPONENTS 4
2.3 WHERE DOES SHIBBOLETH MEET THE JRS? 5
2.4 USING SHIBBOLETH FOR NETWORK LAYER ACCESS 5
2.5 USING THE JRS AS A SHIBBOLETH AUTHENTICATION SERVICE 6
2.5.1 Attribute requirements 7
2.5.2 JRS integration requirements 7
2.5.3 Case (A): JRS integration without attributes 7
2.5.4 Case (B): JRS integration with UKEduPerson attributes 9
2.5.5 Case (C): JRS integration with bespoke attributes 9
2.5.6 Requirements on the UK federation 9
2.5.7 Requirements for managing the Virtual IdP (VIP) 10
2.5.8 Requirements on participating JRS sites 10
2.5.9 Development effort 10

1 Introduction

In this complementary report to the LICHEN development reports, we draw on experience of the LICHEN project to make a proposal for how the JANET Roaming Service (JRS) and Shibboleth 'worlds' could be integrated such that the JRS could be used as a Shibboleth access control back-end through deployment of a Virtual Identity Provider (IdP).

This report is currently in draft status. It would be hardened by end of March 2006 subject to approval to develop the proposal.

The LICHEN project members propose to use remaining (already allocated) LICHEN project budget to prototype this proposal in the period April-July 2006. The feasibility would thus be identified by the time of Shibboleth federation availability to Early Adopters, in advance of any open service to other institutions. The bulk of the development work would be done at Southampton, by David Mills, who also did the bulk of the LICHEN development work. Specific development items are listed towards the end of the proposal.

2 Integrating Shibboleth, the JRS and LICHEN

In the LICHEN-specific reports of this project we have not broached the subject of the possible integration, coexistence and interworking of Shibboleth with either the JRS or any LICHEN servers deployed 'on top' of the JRS.

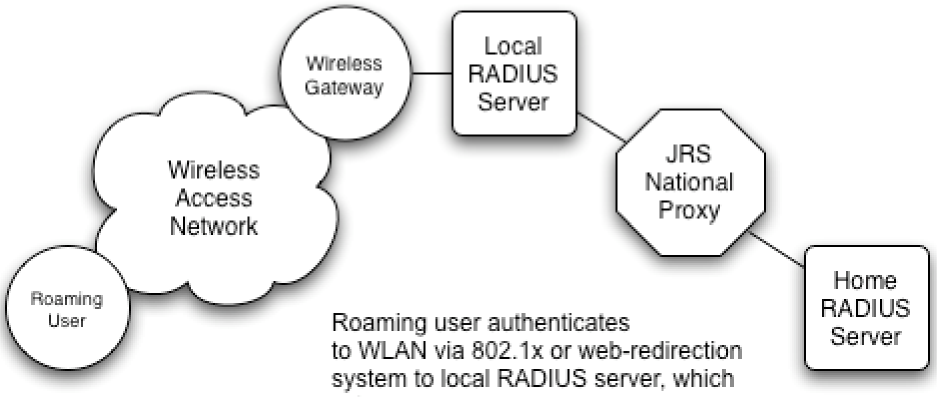
We believe the most fruitful path forward is to utilise Shibboleth as it stands for resource access control, including policy control at the service provider, and use the JRS as the back-end. We reuse the lessons learnt in introducing specific LICHEN policy servers, by introducing a RADIUS interface to the standard Shibboleth IdP, such that it becomes a proxy, or Virtual IdP (VIdP), relaying authentication and attribute requests over the RADIUS-based JRS infrastructure.

In this section we discuss this potential synergy between the JRS and Shibboleth. We describe the problems of trying to use Shibboleth as a network layer authentication service, and the potential advantages for offering the JRS as a Shibboleth authentication (and authorisation) back-end.

2.1 JRS components

The JRS is now entering production service with around 25 participating (university and FE) sites. The system supports roaming users gaining wireless (or wired) access while visiting participating JRS sites.

The JRS uses the RADIUS protocol to relay authentication requests from a local wireless network device or gateway to a site's local RADIUS server, as shown in the Figure. The local device or gateway may be a web-redirection gateway (such as a BlueSocket box) or an 802.1x authenticator. Local users are authenticated locally, visiting users have credentials relayed to their home RADIUS server for authentication via the national RADIUS proxy (managed by UKERNA).



Roaming user authenticates to WLAN via 802.1x or web-redirection system to local RADIUS server, which refers non-local credentials via National RADIUS proxy to user's home site

Figure 2-1: JRS components

This system has proven to be robust and secure (enough) in the 12-18 month trial period (when the JRS was known as the LIN).

JRS gives a method for relaying authentication credentials via a well-established Internet protocol (RADIUS). RADIUS can also relay attributes, but the JRS does not use these; it uses per-site authentication only. It's an 'all trust all' model.

2.2 Shibboleth components

There are four core components of Shibboleth; the client (user), the service provider (SP, the resource owner), the identity provider (IdP, usually the user's home site) and the WAYF (Where Are You From) service.

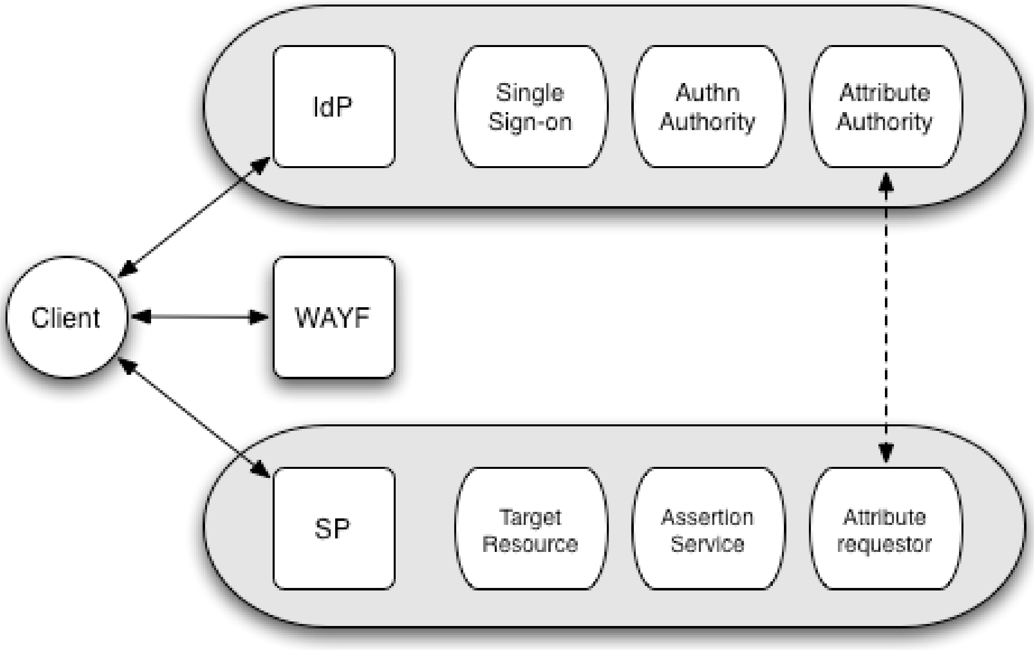


Figure 2-2: Overview of Shibboleth components

The interactions between these components can be complex, but can loosely be summarised as follows. The client attempts to use a Shibboleth 'controlled' web resource at the SP. The SP redirects the client to the WAYF, so the client can select (declare) their home site (IdP). The client then authenticates with their IdP; if successful, further attribute information may be requested from the IdP by the SP, upon which to base a final access decision.

2.3 Where does Shibboleth meet the JRS?

Shibboleth is clearly gaining some traction in the UK academic community, helped by pilot work by the SDSS[1]. There is significant JISC funding to kick-start Shibboleth deployment projects and to support early adopters.

At the same time, the JRS is also gaining traction. Indeed, there are already at least 25 UK sites that are members of the JRS, and who have taken part in the pilot phase of the deployment in the past 12-18 months. This has been successful, and there is every indication that the JRS will grow to many more sites.

So that leaves us with an interesting situation. On the one hand there is Shibboleth as a framework for access control for (web-based) applications, on the other we have the JRS for network layer access control (aimed at Wireless LANs, but with utility for wired networks also).

Is it possible to push Shibboleth down to the network layer, or to push the JRS up to the application layer?

The LICHEN model offers one method for using the JRS for application layer access control; the LICHEN server manages (simple) policies, while the JRS provides the authentication. LICHEN allows a much richer set of applications to be used than Shibboleth currently does, e.g. PAM, or indeed any application with a RADIUS API.

At present, we believe LICHEN is usable as an application layer access control system, although there are caveats as described earlier in this document. The security model is different (arguably weaker, because the authentication, unlike Shibboleth, is not performed to the user's home site infrastructure) and LICHEN does not include Shibboleth's privacy hooks.

At this stage we discuss the JRS as a Shibboleth authenticator, and do not include LICHEN as an 'intermediate' in that authentication process. We feel that LICHEN has its merits for certain applications that Shibboleth cannot (yet) serve, but that there is significant value in allowing the JRS to be used as a Shibboleth authentication (and authorisation, via attributes in RADIUS) back-end.

2.4 Using Shibboleth for Network Layer Access

The problem with using Shibboleth for Wireless network access control is one of scalability.

To use Shibboleth for network access control, the user would need web access to both the WAYF service, and their home institution web authentication service. This would require that the wireless network gateway be configured with a set of IP addresses for all those home web servers, as well as the 'well-known' WAYF server. This doesn't scale, and is similar to the reason why VPN-based access control for

[1] http://sdss.ac.uk/

visiting wireless LANs (you can only get out from the WLAN to your home VPN server) was rejected in favour of web-redirect and 802.1x in the JRS.

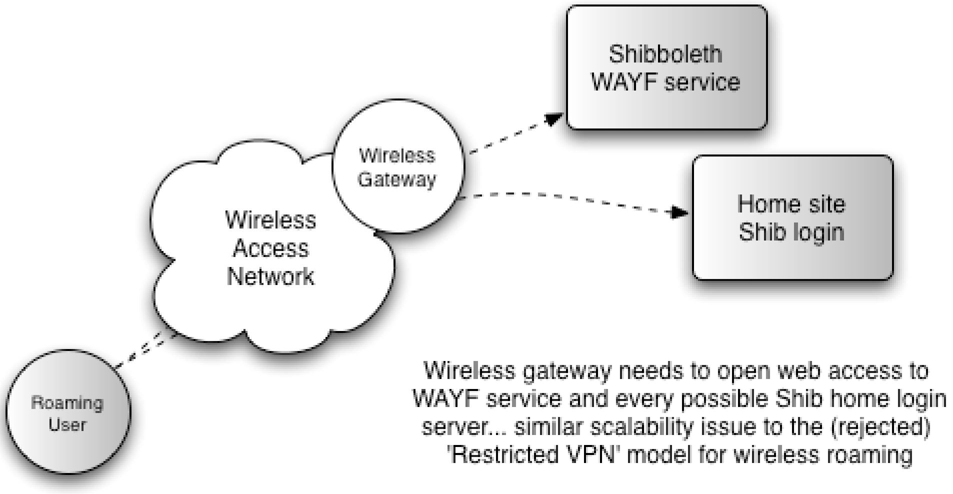

Shibboleth WAYF service
Wireless Gateway
Home site Shib login
Wireless Access Network
Roaming User
Wireless gateway needs to open web access to WAYF service and every possible Shib home login server... similar scalability issue to the (rejected) 'Restricted VPN' model for wireless roaming

Figure 2-3: Using Shibboleth for wireless roaming

We thus do not see any future in pushing Shibboleth down to the network admission control layer.

2.5 Using the JRS as a Shibboleth authentication service

One might reasonably ask why the JRS should be considered as an authentication back-end. One answer is that many sites already have a RADIUS presence, either for JRS or some other network access application, but few have a Shibboleth presence. Having surveyed the Shibboleth early adopter list, and the JRS site list, we believe that of the 25 initial JRS sites (and 5 committed sites) and 30 or so Shibboleth sites, the overlap is only 11 sites. That means that there is plenty of potential for JRS sites to utilise such a facility.

As a member of the JRS, a site's users can be authenticated while visiting other JRS sites via the visited site's RADIUS server and the JRS RADIUS infrastructure.

However, if the 'visited' site is a web server capable of referring authentication requests via RADIUS to the JRS (as per the LICHEN server model) then that server can proxy any web login credentials into the JRS.

Those credentials in the JRS as it stands are rather limited, in that no user attributes are involved, just a user name and a password; the authentication is made via the JRS infrastructure to the user's home RADIUS server, and a binary success indicator is returned. There is no qualification on the user, in the way of meeting certain attribute requirements.

The basic model for bolting the JRS on as a back-end to Shibboleth would be to provide a trusted proxy server that could be selected by a user from a WAYF service. That server would be a proxy to the JRS authentication infrastructure, presented as a (potentially customised per site) web login screen. For this discussion, we refer to the proxy as a Virtual IdP (VIdP).

2.5.1 Attribute requirements

Before we discuss how we could use the JRS as a Shibboleth authentication back end, we should consider the issue of attributes. One of Shibboleth's strengths is its ability to handle (assert) attribute values between the IdP and the SP.

A UKEduPerson attribute set for the UK Shibboleth federation has been defined, based on EduPerson, and has laid out a (limited) set of recommended attributes. These may be sufficient for many – if not most - applications, but there is nothing to prevent an IdP and a SP agreeing their own additional bespoke attributes.

There seem to be three classes of attribute-linked authorisation:

Class (A)

No attributes required; just being a UKEduPerson is enough; in this case authorisation is in effect authentication;

Class (B)

The basic recommended UKEduPerson schema is sufficient; authorisation decisions can be made on a common set of pre-agreed attributes;

Class (C)

Additional bespoke attributes are required; this is the more complex scenario.

It's not clear at this stage what 'share' of the potential application space each of these classes currently represents.

Given these three classes, the question is now how the JRS can support those classes. Class (A) is essentially what the JRS offers now, so could be utilised as is. Class (B) would require some specific injection of attribute data in RADIUS (from the VIdP to the JRS site), where that set of attributes is well-known. Class (C) would present a significant challenge, because bespoke changes are almost certainly be needed for each additional attribute type.

While Class (C) would be problematic, one would imagine that most IdP's would want to maintain only a common, standardised set of attributes, for the sake of their own administration effort. We thus target Class (A) and Class (B) with the proposed VIdP service.

2.5.2 JRS integration requirements

There are certain highly desirable design goals in the JRS integration process.

The most important of these is to minimise (indeed avoid) any code changes being required to Shibboleth for the SP or the WAYF service.

The integration should not add any new security concerns to the Shibboleth environment, nor break the privacy model that it enjoys.

The system should also be simple to deploy (low administrative overhead) for the end sites (who already support the JRS), and be intuitive for the end users.

2.5.3 Case (A): JRS integration without attributes

In this scenario, there are no attributes used, and in effect authorisation is authentication.

The user is able to choose a WAYF selector that will redirect them to the VIP, just as they would do for a regular Shibboleth IdP. The VIP is, in effect, a single sign-on service that can be managed on behalf of the participant JRS sites by an appropriate trusted body (e.g. UKERNA, who manage the JRS). By 'participant' sites we mean JTS sites that request that the WAYF service use the VIP (and JRS) until they deploy their own Shibboleth IdP.

A question arises as to whether to present 'The JRS' as the selector in the WAYF service, or the institution by name. We feel it is better to use the name of the JRS institution, because it allows the site to be identified in the exchange with the VIP, and the proxy server can set the context and provides a seamless path to migrate to use of a Shibboleth IdP later (the user will be presented with the same choice by the WAYF).

The ability for the context to be set for the proxy would allow a custom appearance to be provided (for example, the Institution's branding) for each individual JRS organisation using the VIdP.

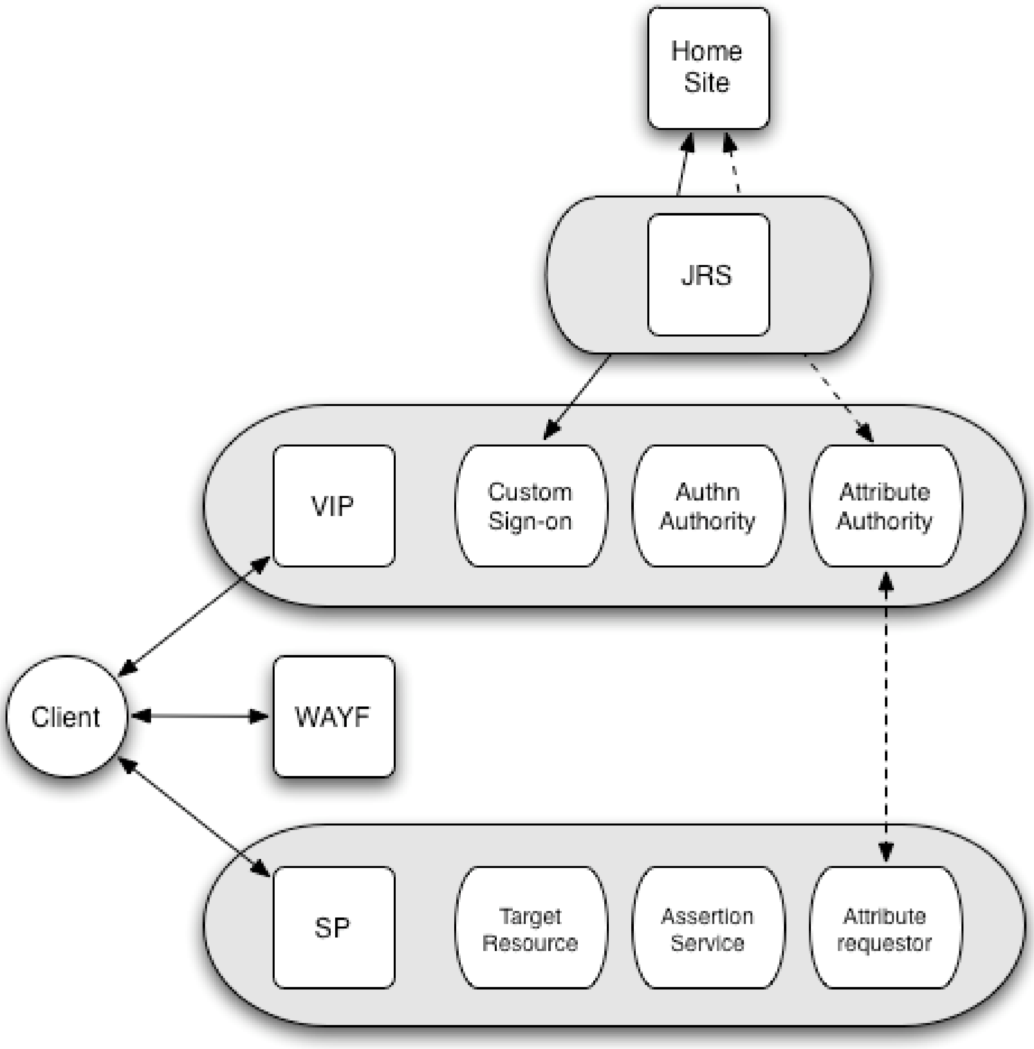


Figure 2-4: Adding the JRS as a Shibboleth VIdP

There are some specifics to consider for this model. For example, in the event that the SP does request attributes, the request needs to fail gracefully.

2.5.4 Case (B): JRS integration with UKEduPerson attributes

In this scenario, we need to support the core set of attributes required for the UK Shibboleth Federation. It is not completely clear what the agreed set is, but documentation at SDSS suggests it will include *eduPersonScopedAfilliation*, *eduPersonTargetedID*, *eduPersonPrincipalName* and *edupersonEntitlement*. We need to be able to carry such attributes between the VIP and the SP.

When the VIP receives a request to provide an attribute in response to a SP's requirements, instead of looking up the attributes locally (as a regular IdP would do), it can make RADIUS requests over the JRS for the data. One way to support this is to use Shibboleth connectors, where custom Java packages can be invoked. The CustomDataConnector element in principle allows Java code to be executed to enable this.

In a standard Shibboleth deployment, the AA for the IdP will query attributes over a (local) protocol such as LDAP. LDAP is not designed to be an 'Internet' protocol, thus our proxy model needs a (secure) method for the VIP's AA to request attribute values from the JRS site it is representing. A Java connector that supports RADIUS is thus required, ideally one that allows TTLS to the home site, to offer a secure end-to-end channel for the attribute lookup. JRadius[2] supports use of arbitrary attributes within the TLS tunnel, and would thus appear a suitable package to use.

Given the use of TLS, we then need to consider the PKI/CA implications. In the Shibboleth world, the federation is a CA that provides server certificates to IdPs and SPs; in the VIP model, each JRS site that joins the VIP would issue a certificate (its CA's root certificate) to the VIP.

An attribute release policy (ARP) could be implemented via configuration of the JRS site's RADIUS server.

It may be useful to recommend a common set of LDAP attribute value names for JRS sites to use, e.g. UKEduPerson_*. Administrators may choose to wildcard certain attributes, e.g. the scoped affiliation.

Note that Shibboleth uses handles rather than IDs (for privacy reasons) so the VIP needs to maintain a mapping of which handle maps to which ID, and only present the handle to the SP.

2.5.5 Case (C): JRS integration with bespoke attributes

We do not consider this case (yet!).

2.5.6 Requirements on the UK federation

A policy decision would be required as to whether JRS-enabled sites could in effect become Shibboleth capable by using the proposed VIP. This could be done on a site by site basis.

Another policy decision would be required as to whether other specific non-UK eduroam-enabled sites could be listed in the WAYF selection.

[2] http://jradius.sourceforge.net/

2.5.7 Requirements for managing the Virtual IdP (VIP)

A trusted entity is required to maintain and configure the proxy server(s); this would perhaps be UKERNA, who manage the JRS service.

- An SSL certificate would need to be obtained for the VIdP server.
- The proxy would need to be run on an appropriate enabling platform, e.g. Apache2
- The VIdP needs to receive a certificate from each JRS site using it.

2.5.8 Requirements on participating JRS sites

A JRS-enabled site that wished to gain early Shibboleth application would need to:

- Confirm that it is wants its WAYF selector to point to the proxy server
- Provide customisation information for the proxy web server (potentially some logo(s) and/or text/names/etc.
- Provide it's CA's root certificate to the VIP (for the JRadius TTLS connector)
- Needs to configure RADIUS to query its back-end (e.g. LDAP) for attribute values

2.5.9 Development effort

The following development is required:

- Production of a VIdP web interface, allowing customisation per JRS participant using the VIdP
- Identification of the JRS site from the exchange between the Client and the VIdP after the WAYF has been contacted
- Configuration of the VIdP web server to use *mod_auth_radius* (or an equivalent, maybe some CGI) for the JRS-based authentication
- Development of a JRadius hook for the connector service to retrieve attribute data over RADIUS
- Testing and documentation of the system