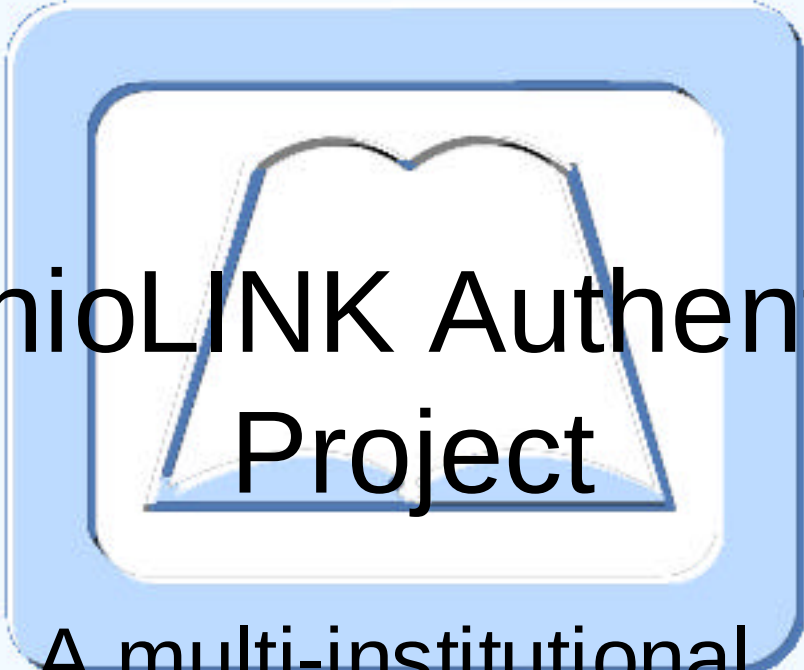




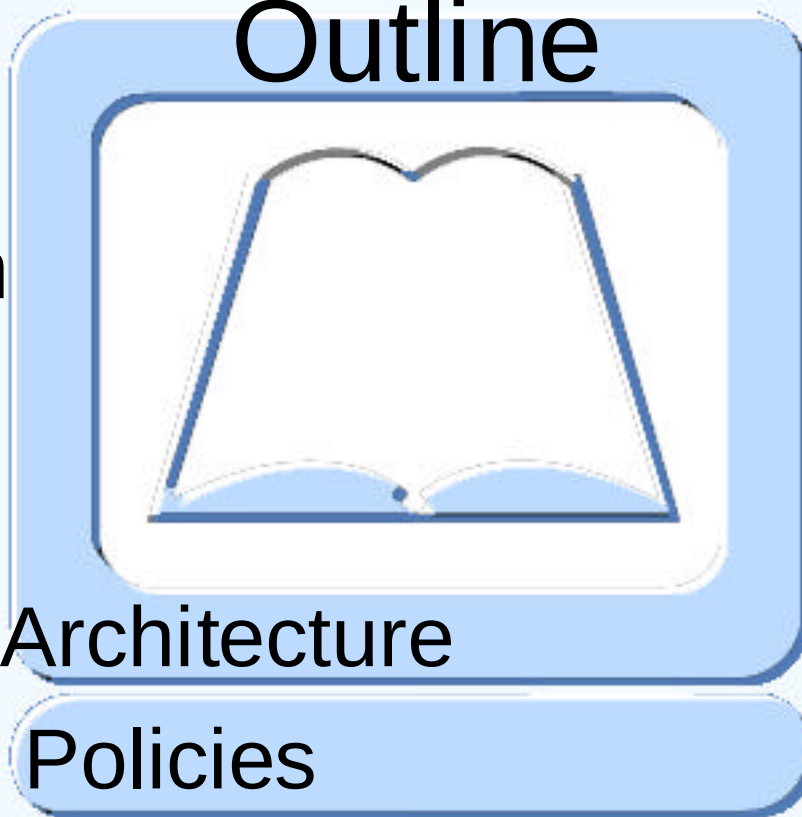
The OhioLINK Authentication Project

A multi-institutional,
cookie-based authentication system for
local and remote resources



Outline

- Problem
- Solution
- Process
- System Architecture
- Security Policies
- Adapting the System to Handle Remote Resources



OHIO LINK

Problem

- IP authentication no longer enough
 - Outsourcing of campus dialup services
 - Faster Internet access through commercial ISPs
 - Cable modems, ADSL
 - Distance education programs
 - New services require user specific Ids

Solution for Local Resources

- Authenticate users in OhioLINK patron databases if they come from unknown IP addresses
- Give the authenticated users' browser a cookie that is accessible to all OhioLINK's servers
- Keep all of the cookies in a database and with every request, validate the user's cookie against this database

Process: User Experience

1. User requests URL from OhioLINK Server
2. If IP address invalid, user sent to cookie authentication form
3. User submits credentials and is authenticated
4. User's browser given cookie and is redirected to originally requested page

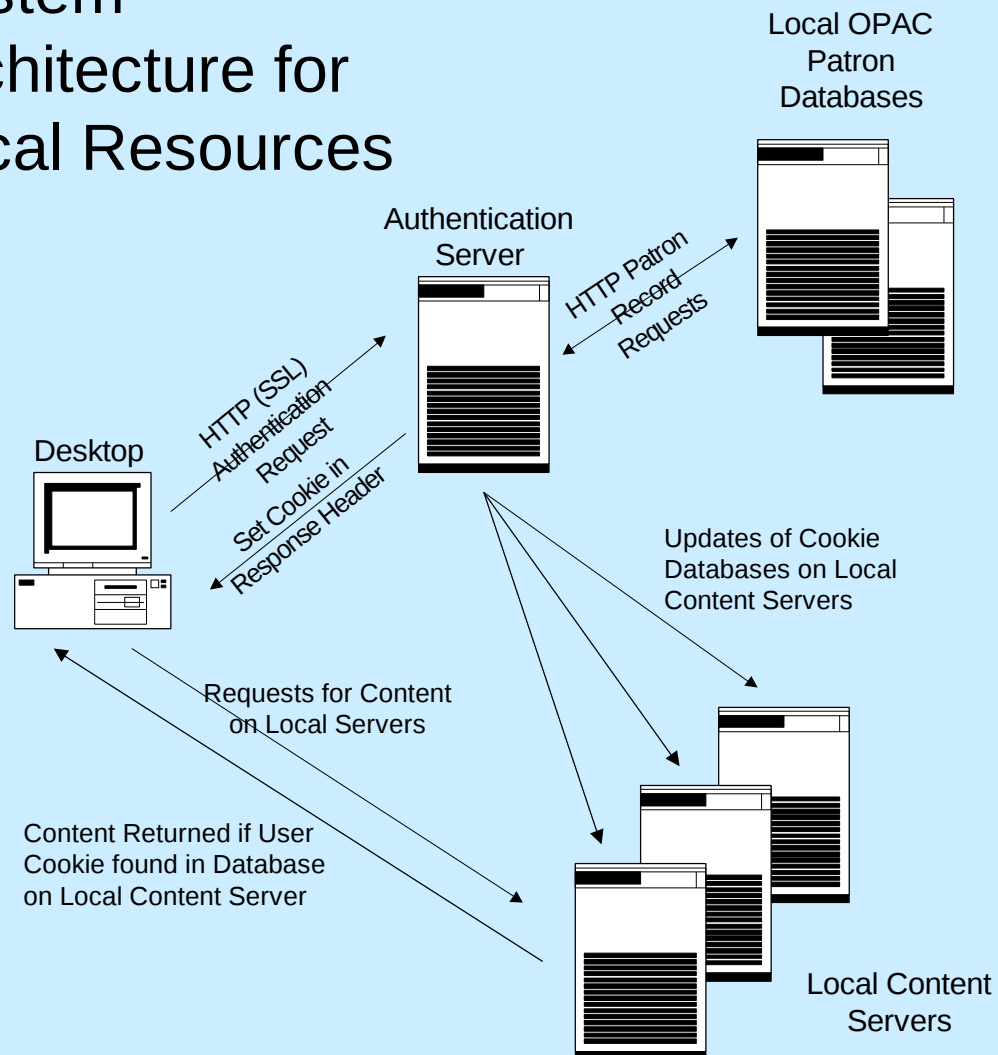
Process: Technical View

1. User submits credentials to PERL CGI script
2. Script checks if patron in local patron record cache, if not patron record requested from patron's campus OPAC
3. After patron authenticated, patron added to patron database on authentication server if not already cached.

Technical View (cont.)

4. Secure one-way hash used to create cookie
5. Cookie added to cookie database on authentication server and on local content servers.
6. Response header given back to user with cookie and Location redirect to originally requested URL

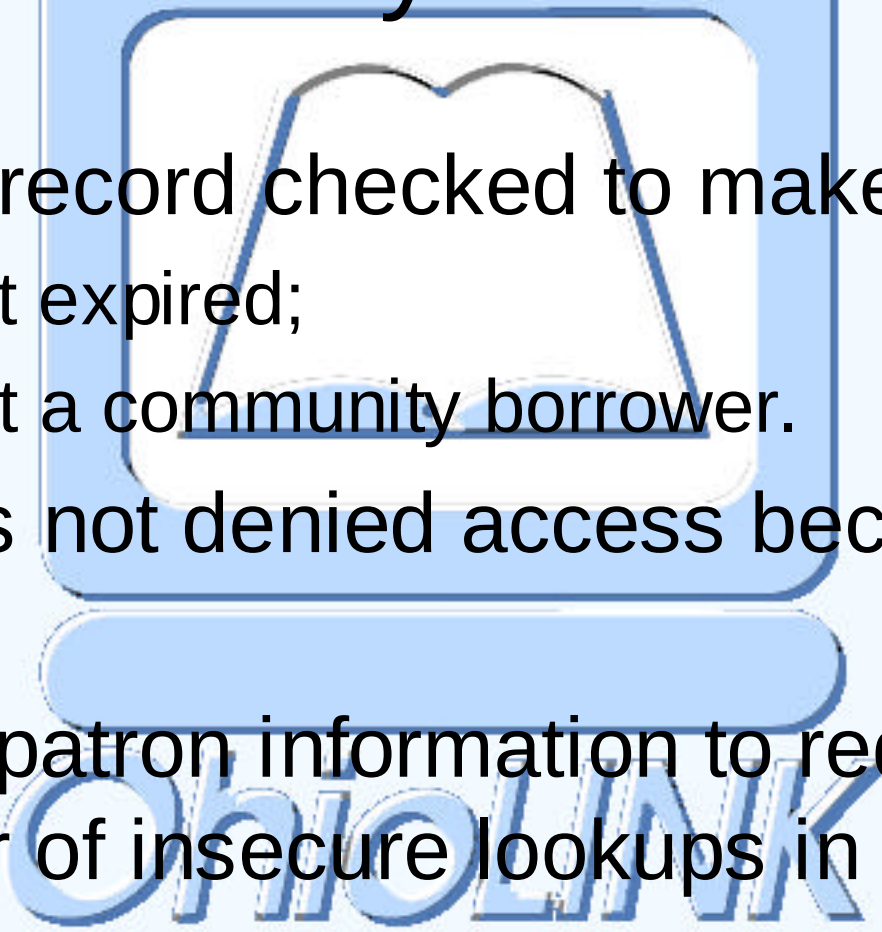
System Architecture for Local Resources



System Architecture WWW Server Configuration

- Apache WWW Server
 - Two virtual hosts listening on two ports: one port for cookie users, one for IP authentication
 - Mod_rewrite used to automatically forward users with cookies to cookie port
 - Custom module used for cookie validation

Security Policies

The logo for OhioLINK is centered in the background. It features a blue square with rounded corners containing a white stylized open book. Below the square is a blue horizontal bar. At the bottom, the word "OHIO" is written in a large, blue, serif font, and "LINK" is written in a smaller, blue, sans-serif font to its right.

- Patron record checked to make sure:
 - 1.) not expired;
 - 2.) not a community borrower.
- Patrons not denied access because of fines
- Cache patron information to reduce number of insecure lookups in patron database

More Policies

- Track authentication failures
- Each time cookie used, it is checked to make sure the cookie continues to come from the same address
- Restrict OPAC patron database access to OhioLINK machines

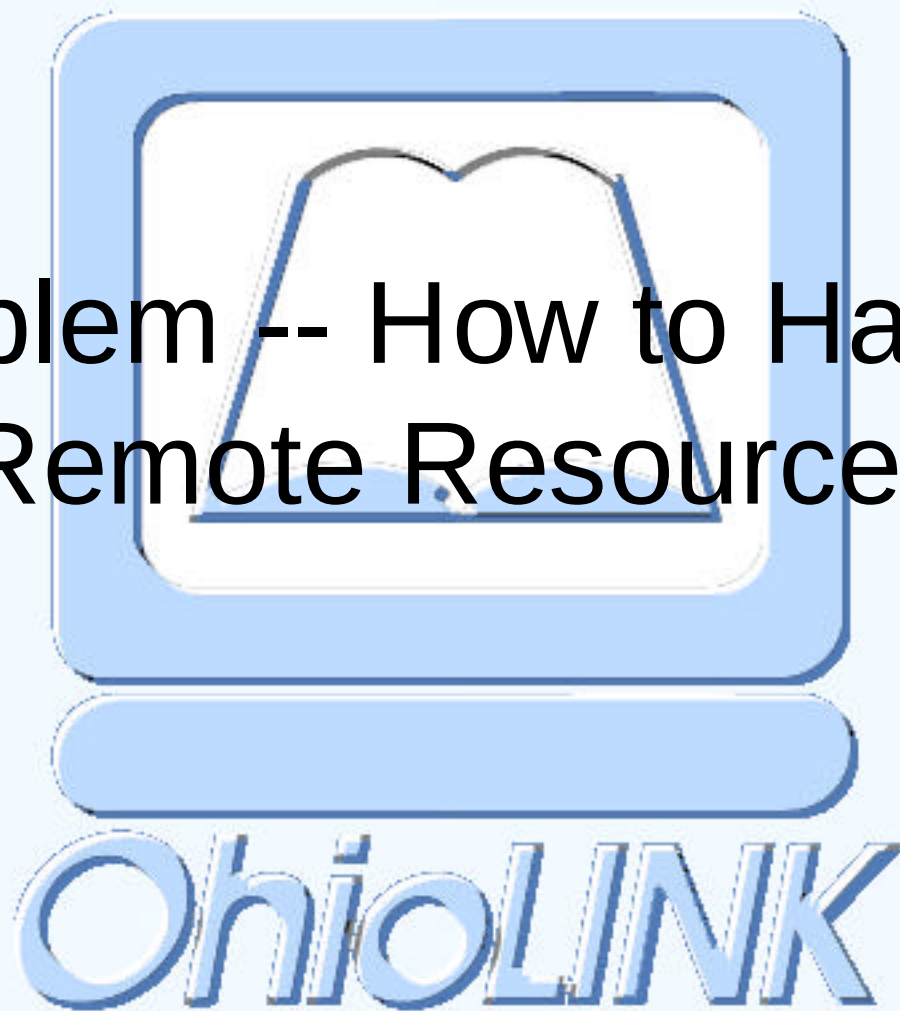
OhioLINK

More Policies

- Use Post requests and password HTML fields
- Privacy protection
 - Cookie not added to records in patron database
 - User can not be identified from cookie

OhioLINK

Problem -- How to Handle Remote Resources



Solution (In Development)

- Send users through a local WWW server gateway (filtering proxy) to access remote resources
- Authenticate gateway users with cookie

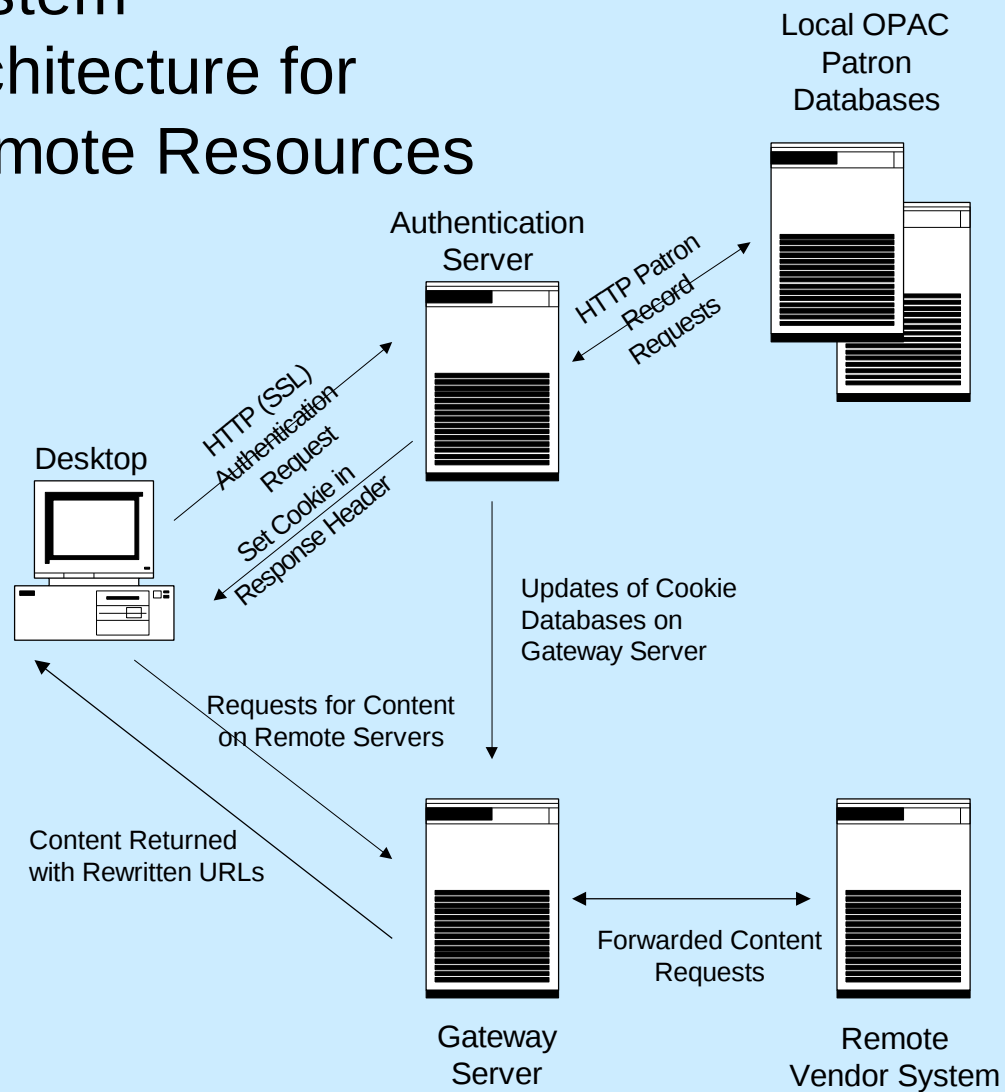
The logo for OhioLINK, featuring the word "OhioLINK" in a stylized, blue, 3D font. The "O" is large and circular, and the "LINK" is in a more compact, blocky font. The entire logo is set against a light blue background.

OhioLINK

Gateways: Rationale

- Privacy: Limits user information given to remote vendor
- Selected remote vendors might provide an alternative solution, but many will not
- By manipulating access to remote sites, value can be added to remote content.
 - a frame could be added with local information e.g. links to local holdings

System Architecture for Remote Resources



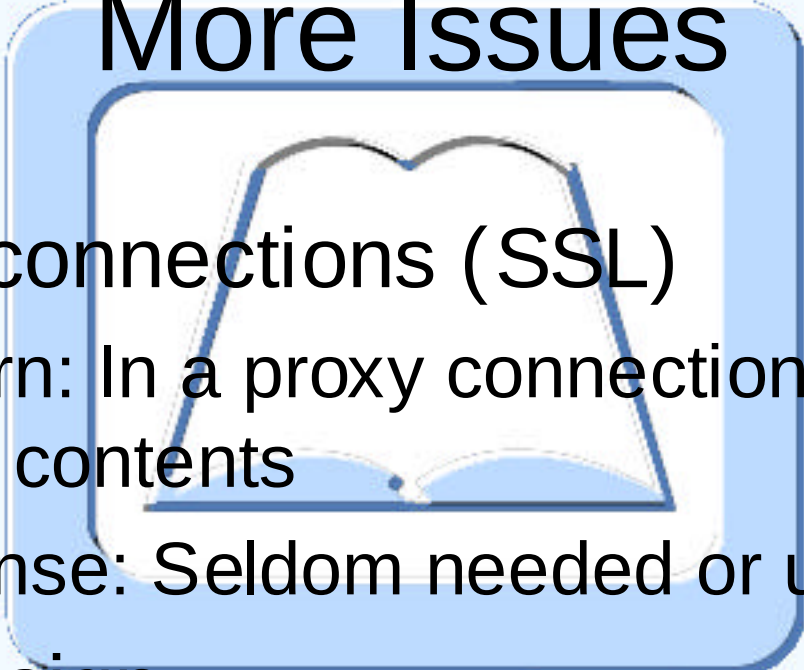
Gateway Software

- Using Apache with mod_perl and LWP for initial test
- Any gateway needs two things
 - API to insert new process into flow of WWW server
 - Regular expression library to modify document contents

Implementation Issues

- Java applets
 - Concern: socket connections to the remote host
 - Response: firewalls make this unlikely
- Performance
 - Concern: proxy/gateway servers are notoriously slow
 - Response: only operating gateways for selected resources

More Issues



- Secure connections (SSL)
 - Concern: In a proxy connection, can't modify contents
 - Response: Seldom needed or used
 - Page design
 - Concern: page might get redesigned
 - Response: HREFs are always formed the same way
- 

Summary

- Cookie given to users based on authentication in local patron databases
- Local Resources -- cookie used by local Apache server to provide access to unrecognized IP addresses
- Remote Resources -- cookie used by Apache gateway that proxies user access to remote vendor systems