Single Click User Request Submission via the ILLiad API

Krista Higham, Millersville University
Dustin Stokes, Atlas Systems, Inc.

March 20, 2019
OCLC Resource Sharing Conference
Millersville University

- Mid-size masters level public university
  - Plus three doctorate programs
- Approximately 7,000 students
- Library houses 280,000 physical volumes with 335,000 records in the catalog
- Use ILLiad, Rapid, and PALCI’s E-ZBorrow
- Use EDS – EBSCO Discovery Service
The thoughts...

- Too many logins for library services
- Implemented OpenAthens in 2018
- Users now log into databases/discovery
- Should be able to use OA to pass information
- Users can't edit their personal data in ILLiad
  - We should pass information for new users
- Why don't we automate requesting
OpenAthens, aka "OA"

- OA is a form of Single Sign On (SSO)
  - Uses SAML (security assertion markup language)
    - industry standard markup for security/authentication

- Similar to "Shibboleth" in higher education
  - BUT, comes with catalog of connections to vendors
    - Local IT makes ONE connection to OA
    - OA then connects to vendors – far less work for local IT
    - Get functionality not common with Shibboleth

- Enter library with "standard university account"
  - Users login to library regardless of physical location
OpenAthens, Functionality

- IP / Proxy "Reporting"
  - IP and Proxy didn't IDENTIFY user in any way
  - Not even aggregate level of Faculty, Student, Staff
  - Pretty bland narrative for reports / analytics

- Reporting is a big add for OA
  - Local IT have NO reporting in place for Univ Shibboleth
    - More precisely …. "not gonna do it"
  - Anything released about users could be reported on
    - Faculty, Staff, Student, major, minor, course enrollment, etc.
    - More attributes released; more granularity possible
    - Entirely local decision about what attributes to release
OpenAthens, Functionality

- Generate OA accounts for "special cases"
  - Avoid "institution domain accounts" for just library
    - Local AP students, research visitors, etc.
  - Address specific "public" computers
    - Walk-in / public users
    - Meets federal depository compliance
  - Far less complicated (politically & technically)
  - Library is more responsive to users
OpenAthens, Functionality

- Applying Logic
  - Now that we know something about a specific user

- We can tailor access to type of user
  - Do or do not get access to resources
  - Do or do not get access to services, such as ILL

- Pass along salient pieces of user information
  - Without custom programming
  - To say … an ILL service … who knew !?!?
OpenAthens @ Millersville

- Soft launch, Summer-ish 2018
- Nearly everything by, Fall 2018
  - Spring 2019, ILLiad login OA enabled … it really works.

- Reporting
  - Currently have Faculty, Students, Staff,
    - With individual attributes – username, basic directory info
  - Working on Major, Minor, Courses

- Example:
  - For any given time period …
  - Who / how many were redirected to an ILL FORM ? (OA)
    - Could infer "loss rate" from actual ILL requests placed (ILLiad)
Nearly one click now ...

- User Experience
  - Why keep forcing logins?
  - Can accurately associate request with an actual user

- Currently with OA deployed
  - Have item, press request link, ILL form populated without user logging in again ...
  - If required fields present, user presses "submit"
  - If required fields not all present, user completes, then "submit"
With OA and EBSCO logic deployed  
  ◦ Have item, press request, done.

  ▪ If required metadata present (EBSCO determines)  
    ▪ submit the request directly via the ILLiad API  
    ▪ No user review  
      ▪ we know the user + have data we need

EBSCO checks for "required data" that was previously in the ILL form, if present, submit request along with OA supplied user information. Done.
The "operational logic"

- **Sufficient request metadata present**
  - Previously what we called "required" on the form
    - Plus other stuff if present
  - Request passed through via API
    - “thank you for your request”
  - No user review

- **Insufficient request metadata**
  - ILLiad form presented
  - User provides missing "required" elements
    - Plus other stuff if they want
  - Submits request as in the olden days
Potential Issues

- Not needed after date
  - Users rarely change this date
  - Was considering removing it from the form

- Notes
  - 2.2% of requests contain a note
  - Removing the “Need ASAP” and “Thank you” notes
    - 1.7% of requests contain a somewhat useful note
    - Mostly DOIs
  - About 0.8% of requests contain “helpful” notes
    - “Only need chapter 2”, “found citation at this link”
Logic for the Single Click

- We can rebuild this logic
- We have the technology
- We have the capability to make it ...
- Better than it was before
- Stronger than it was before
- Faster than it was before

- So Steve Austin doesn't need to sign into six million different systems !!!!!!
ILLiad API

- The ILLiad Web Platform provides programmatic access to your ILLiad installation.
- Designed for use by developers for creating robust, easy to maintain integrations between ILLiad and other external systems.
- Introduced in 8.4 with additional functionality added in each new release.
Existing API Scenarios

- Transparent request and user creation from discovery systems (EDS and Primo)
- Unmediated ILL (Relais D2D)
- Consolidated request lists in ILSes
- Coordination between multiple fulfillment systems (Aeon)
Transparent Requesting From Discovery

User → Discovery
- Browse to Discovery
- Search for item
- Search Results
- Request Item

alt
- [If Authenticated]
  - Read Username from SSO Token

[If Not Authentication]
- Redirect to SSO

Discovery → ILLiad
- User Exists?
  - Yes or No
    - opt
      - [If No]
        - Create User
          - User Created
        - Create Request for User
          - Request Created

User → Discovery
- Request Placed

www.websequence diagrams.com
Available API Functions

- Retrieve transactions by transaction number
- Retrieve transactions by username
- Create a new transaction
- Route transactions
- Updated a transaction as filled or unfilled
- Add and retrieve transaction history and notes.
- Create new users
- Retrieve user
- Send notifications to users and borrowing libraries
ILLiad API for first-time Users

- Review the documentation for instructions and warnings.

- Generate an API key using Customization Manager.
  - Each application accessing the Web Platform should have its own unique API Key
  - External systems use the API key for authentication.

- Work with developers of the external system to build the integration.
Upcoming API Changes

- Additional transaction processes
  - Mark Found, Document Uploading, Clearing Users, Merging Users, Sending Requests, Etc.

- API Key specific permissions

- Other API scenarios as we encounter them

- Slow transition of most business logic out of the client
User Experience after the API

- Will we change the wording for the Request button?
- What messaging will appear for the “request submitted” process?
- Hover text over the Requesting button to explain the process.
- Advertising, needed or not needed?
Conclusion

- Fewer clicks
- More like e-commerce
- Better reporting
- Easier customization
- More responsive to user needs
Questions?

- Krista Higham
  - Access Services Librarian
  - Millersville University
  - Krista.higham@millersville.edu

- Dustin Stokes
  - Chief Technology Officer
  - Atlas Systems, Inc.
  - dstokes@atlas-sys.com