Outline

- Origin & goals
- Demo of DIBS
- System architecture
- Conclusion
## DIBS origin story

Pandemic → libraries close, classes go online  

Spring 2020: Password protected PDFs in Box  

Decision in late 2020: do CDL – and a.s.a.p.!  

Investigated options, spoke w/ other groups  

Found no suitable off-the-shelf solution  

Jan. 2021: started development  

Mar. 2021: deployed for spring term
Position Statement on CDL

https://controlleddigitallending.org/statement

“... a good faith interpretation of U.S. copyright law for American libraries considering how to perform traditional lending functions using digital technology while preserving an appropriate balance between the public benefit of such lending and the protected interests of private rights holders.”

“CDL is about replicating with digital lending the legal and economically significant aspects of physical lending.”
CDL Requirements

1. Ensure original works are acquired lawfully
2. Apply CDL only to works that are owned & not licensed
3. Maintain an “owned to loaned” ratio: Limit total number of copies in any format in circulation at any time to the number of physical copies the library lawfully owns
4. Lend each digital version only to a single user at a time
5. Limit the loan time period analogously to physical lending
6. Use access controls to prevent copying and redistribution
Adjunct objectives

Single Sign On authentication capabilities for local projects
https://www.shibboleth.net/about-us/the-shibboleth-project/

IIIF (International Image Interoperability Framework)
knowledge and capabilities
https://iiif.io/community/faq/

Standardize on the Universal Viewer
https://universalviewer.io/
Origin as single viewer across British Library websites
IIIF enabled
Extensible (3D, audio, video, PDF)
DIBS design goals

- Keep it simple (concept, code, interface)
- Preserve patron privacy
- Serve scanned books using IIIF
- Use institutional single sign-on for auth.
- Limit interactions w/ other systems

“Everything should be made as simple as possible, but not simpler.”
Loan policy

- Remove physical items from circulation
- Limit people to 1 loan at a time
- Limit loan durations
- Impose wait between repeated loans
- **No queue or reservation** (currently)
Demo of DIBS
System architecture
Software elements

Python (server-side)
JavaScript
Apache web server
Ubuntu 20
Software elements

Python (server-side)

JavaScript

Apache web server

Ubuntu 20

- Python scripts for image conversion & generation of manifests for IIIF
- Python-based DIBS server
  - Bottle for WSGI web framework
  - SQLite database for items & loans
  - Peewee ORM for database interface
  - Other Python packages
Software elements

- Python (server-side)
- JavaScript
- Apache web server
- Ubuntu 20

- IIIF server back end (on AWS)
- Universal Viewer (in browser)
  - Downloads are disabled
  - Interface is slightly modified
- Custom AJAX-based calls to DIBS and misc. UI controls in web pages
Conclusion
Feature summary

DIBS is a basic, standalone, CDL implementation
Software developed in-house by Caltech DLD
Content scanned by Library collection services
Written in Python, JavaScript, HTML, CSS
Uses IIIF and the *Universal Viewer*
Uses institutional single sign-on
DIBS server runs as WSGI application in Apache
Future plans

- Link from Library’s ILS records to DIBS items
- Explore other scenarios w/ Caltech Archives
- Switch to MySQL for DIBS database
- Improve accessibility
- Improve customizability
Caltech DIBS (Digital Borrowing System)

Source Code  https://github.com/caltechlibrary/dibs
License       BSD 3-clause
Online docs   https://caltechlibrary.github.io/dibs

Mike  Robert  Tommy  Stephen

Thank you!