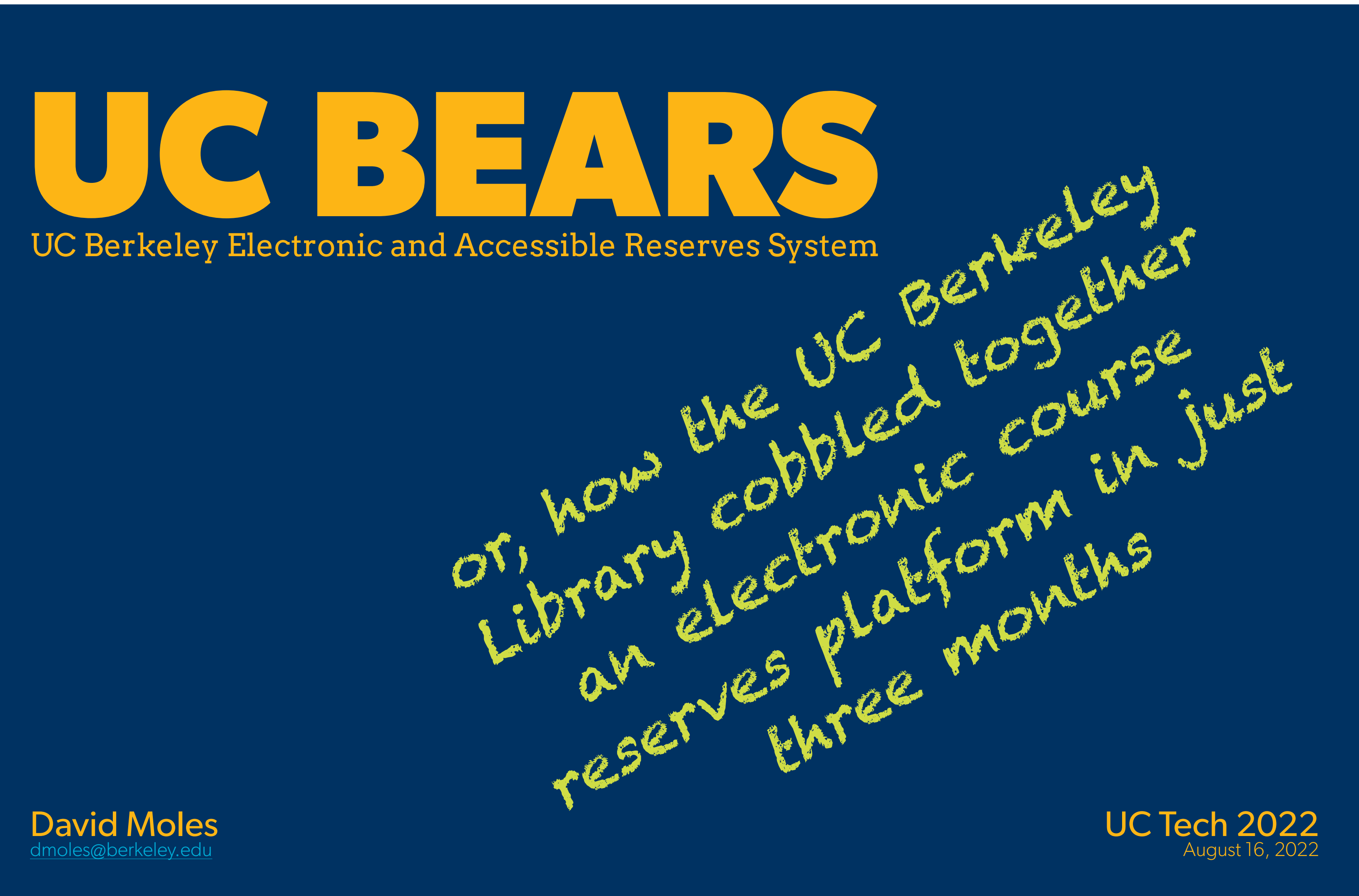
Hi, folks. My name is David Moles. I'm head of applications at the UC Berkeley Library, and I'm here to talk to you about our in-house electronic course reserves platform, which we call UC BEARS — the UC Berkeley Electronic and Accessible Reserves System — and which we built over the course of about three months in the summer of 2021.

My alternate title for this talk was "How I spent your summer vacation", but I'm pretty sure all of you were working just as hard that summer as we were.



UC BEARS

UC Berkeley Electronic and Accessible Reserves System

or, how the UC Berkeley Library cobbled together an electronic course reserves platform in just three months

David Moles
dmoles@berkeley.edu

UC Tech 2022
August 16, 2022

So, I’m going to start off with some background.

background

We closed down in March 2020, just like all of you did.

And here in the library, we were scrambling to figure out how to get books and other resources to students, when we couldn't even get into the stacks ourselves.

Timeline

March 2020 Berkeley campus (& libraries) closed

Thankfully our friends at HathiTrust stepped up with their Emergency Temporary Access Service.

Timeline

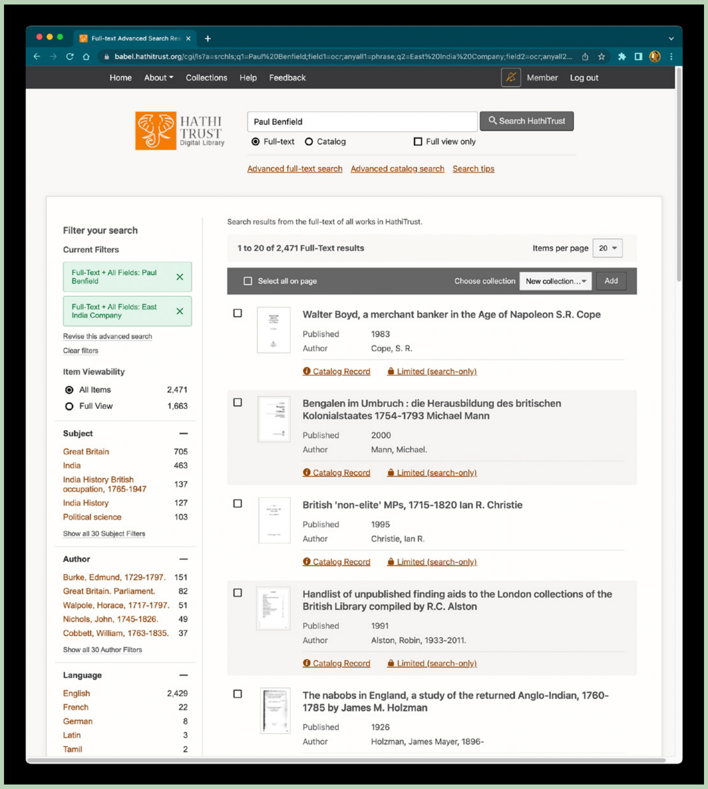
March 2020	Berkeley campus (& libraries) closed
April 2020	HathiTrust opens ETAS

For those of you who aren’t familiar with HathiTrust, it’s a collaborative of academic and research libraries that works to digitize library collections and make them available to researchers.

You know about Google Books? This is the other side of that.

HathiTrust

collaborative of academic and research libraries


Full-text Advanced Search Res
babel.hathitrust.org/cgi/ls?a=srchls;q1=Paul%20Benfield;field1=ocr;anyall1=phrase;q2=East%20India%20Company;field2=ocr;anyall2...
Home
About
Collections
Help
Feedback
Member
Log out
HATHI TRUST Digital Library
Search HathiTrust
Advanced full-text search
Advanced catalog search
Search tips
Select all on page
Choose collection
New collection...
Add

Paul Benfield

[✓ Full-text] Catalog Full view only

Filter your search

Current Filters

Full-Text + All Fields: Paul Benfield

Full-Text + All Fields: East India Company

Revise this advanced search

Clear filters

Item Viewability

[✓ All Items]	2,471
Full View	1,663

Subject

Great Britain	705
India	463
India History British occupation, 1765-1947	137
India History	127
Political science	103

Show all 30 Subject Filters

Author

Burke, Edmund, 1729-1797.	151
Great Britain. Parliament.	82
Walpole, Horace, 1717-1797.	51
Nichols, John, 1745-1826.	49
Cobbett, William, 1763-1835.	37

Show all 30 Author Filters

Language

English	2,429
French	22
German	8
Latin	3
Tamil	2

Search results from the full-text of all works in HathiTrust.

1 to 20 of 2,471 Full-Text results

Items per page 20

Walter Boyd, a merchant banker in the Age of Napoleon S.R. Cope
Published 1983
Author Cope, S. R.
Catalog Record Limited (search-only)

Bengalen im Umbruch : die Herausbildung des britischen Kolonialstaates 1754-1793 Michael Mann
Published 2000
Author Mann, Michael.
Catalog Record Limited (search-only)

British 'non-elite' MPs, 1715-1820 Ian R. Christie
Published 1995
Author Christie, Ian R.
Catalog Record Limited (search-only)

Handlist of unpublished finding aids to the London collections of the British Library compiled by R.C. Alston
Published 1991
Author Alston, Robin, 1933-2011.
Catalog Record Limited (search-only)

The nabobs in England, a study of the returned Anglo-Indian, 1760-1785 by James M. Holzman
Published 1926
Author Holzman, James Mayer, 1896-

Hathi has more than 17 million digitized items, about a quarter of which come from UC libraries.

We're the second largest contributor, after University of Michigan, they're a couple of hundred thousand books ahead of us.

Harvard has about a million, UIUC has almost that many, there's a dozen other libraries that have all scanned hundreds of thousands.

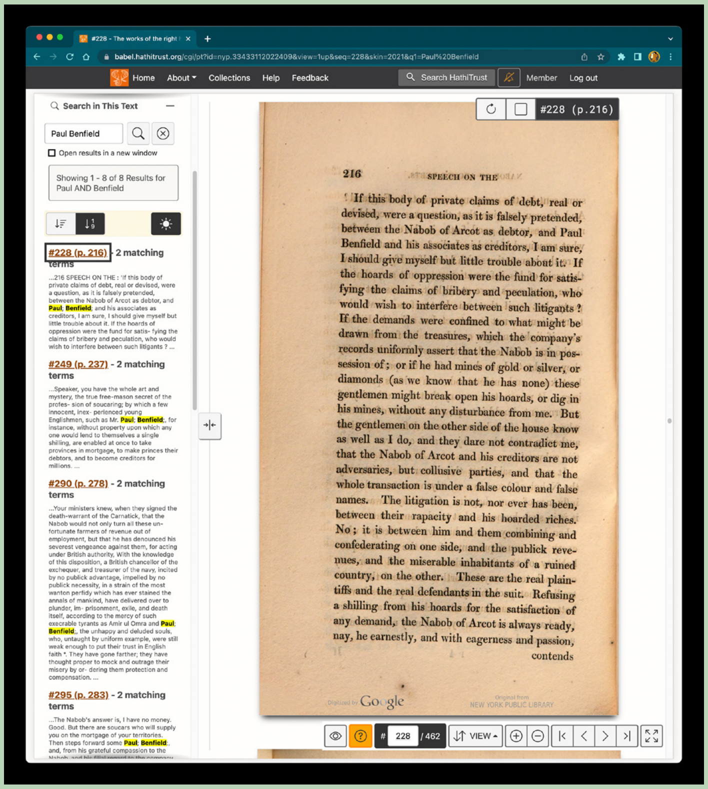
Obviously there's some duplicates, but it's a pretty big collection.

Hathi indexes all of them, they provide full-text search, they provide this viewer you can see here.

HathiTrust

collaborative of academic and research libraries

17+ million digitized items (4.6 million from UC)


#228 - The works of the right
babel.hathitrust.org/cgi/pt?id=nyp.33433112022409&view=1up&seq=228&skin=2021&q1=Paul%20Benfield
Home
About
Collections
Help
Feedback
Search HathiTrust
Member
Log out
Search in This Text
Paul Benfield
Open results in a new window
#228 (p.216)
# 228 / 462
VIEW
Digitized by Google
Original from NEW YORK PUBLIC LIBRARY

Showing 1 - 8 of 8 Results for Paul AND Benfield

#228 (p. 216) - 2 matching terms

...216 SPEECH ON THE : 'If this body of private claims of debt, real or devised, were a question, as it is falsely pretended, between the Nabob of Arcot as debtor, and Paul; Benfield; and his associates as creditors, I am sure, I should give myself but little trouble about it. If the hoards of oppression were the fund for satis- fying the claims of bribery and peculation, who would wish to interfere between such litigants ? ...

#249 (p. 237) - 2 matching terms

...Speaker, you have the whole art and mystery, the true free-mason secret of the profes- sion of soucaring; by which a few innocent, inex- perienced young Englishmen, such as Mr. Paul; Benfield;, for instance, without property upon which any one would lend to themselves a single shilling, are enabled at once to take provinces in mortgage, to make princes their debtors, and to become creditors for millions. ...

#290 (p. 278) - 2 matching terms

...Your ministers knew, when they signed the death-warrant of the Carnatick, that the Nabob would not only turn all these un- fortunate farmers of revenue out of employment, but that he has denounced his severest vengeance against them, for acting under British authority, With the knowledge of this disposition, a British chancellor of the exchequer, and treasurer of the navy, incited by no publick advantage, impelled by no publick necessity, in a strain of the most wanton perfidy which has ever stained the annals of mankind, have delivered over to plunder, im- prisonment, exile, and death itself, according to the mercy of such execrable tyrants as Amir ul Omra and Paul; Benfield;, the unhappy and deluded souls, who, untaught by uniform example, were still weak enough to put their trust in English faith *. They have gone farther; they have thought proper to mock and outrage their misery by or- dering them protection and compensation. ...

#295 (p. 283) - 2 matching terms

...The Nabob's answer is, I have no money. Good. But there are soucars who will supply you on the mortgage of your territories. Then steps forward some Paul; Benfield;, and, from his grateful compassion to the

216 SPEECH ON THE

'If this body of private claims of debt, real or devised, were a question, as it is falsely pretended, between the Nabob of Arcot as debtor, and Paul Benfield and his associates as creditors, I am sure, I should give myself but little trouble about it. If the hoards of oppression were the fund for satisfying the claims of bribery and peculation, who would wish to interfere between such litigants? If the demands were confined to what might be drawn from the treasures, which the company's records uniformly assert that the Nabob is in possession of; or if he had mines of gold or silver, or diamonds (as we know that he has none) these gentlemen might break open his hoards, or dig in his mines, without any disturbance from me. But the gentlemen on the other side of the house know as well as I do, and they dare not contradict me, that the Nabob of Arcot and his creditors are not adversaries, but collusive parties, and that the whole transaction is under a false colour and false names. The litigation is not, nor ever has been, between their rapacity and his hoarded riches. No; it is between him and them combining and confederating on one side, and the publick revenues, and the miserable inhabitants of a ruined country, on the other. These are the real plaintiffs and the real defendants in the suit. Refusing a shilling from his hoards for the satisfaction of any demand, the Nabob of Arcot is always ready, nay, he earnestly, and with eagerness and passion, contends

If something's in-copyright, you can still do a search, and you can find out where your search terms appear and on what pages;

and you can do computational text analysis, there are some nice rich datasets Hathi makes available for that;

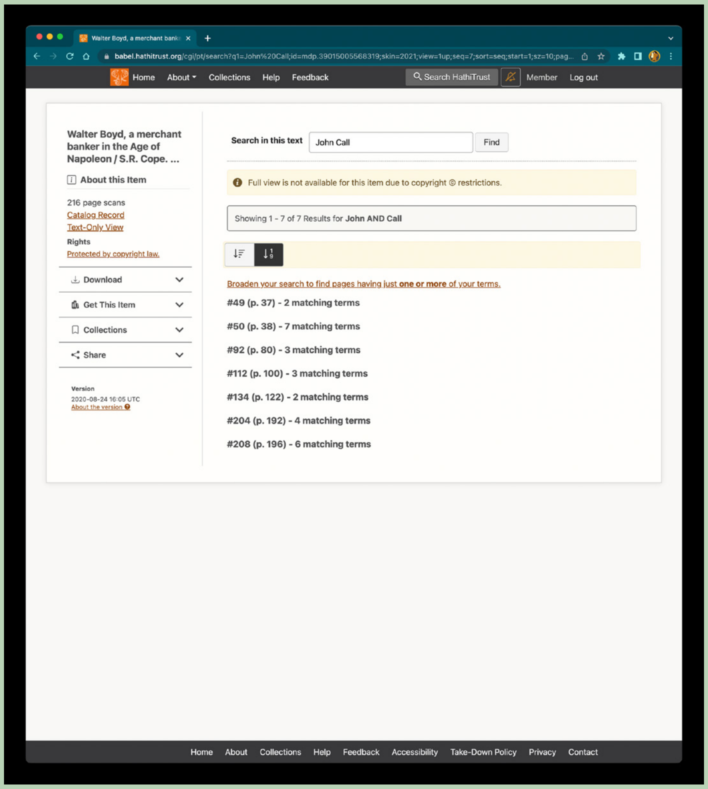
but you can't just read the book.

HathiTrust

collaborative of academic and research libraries

17+ million digitized items (4.6 million from UC)

reading access limited by US copyright law


Walter Boyd, a merchant banke

babel.hathitrust.org/cgi/pt/search?q1=John%20Call;id=mdp.39015005568319;skin=2021;view=1up;seq=7;sort=seq;start=1;sz=10;pag...

Home About Collections Help Feedback Search HathiTrust Member Log out

Walter Boyd, a merchant banker in the Age of Napoleon / S.R. Cope. ...

About this Item

216 page scans
Catalog Record
Text-Only View
Rights
Protected by copyright law.

Download
Get This Item
Collections
Share

Version
2020-08-24 16:05 UTC
About the version

Search in this text: John Call [Find]

Full view is not available for this item due to copyright © restrictions.

Showing 1 - 7 of 7 Results for **John AND Call**

Broaden your search to find pages having just **one or more** of your terms.

#49 (p. 37) - 2 matching terms
#50 (p. 38) - 7 matching terms
#92 (p. 80) - 3 matching terms
#112 (p. 100) - 3 matching terms
#134 (p. 122) - 2 matching terms
#204 (p. 192) - 4 matching terms
#208 (p. 196) - 6 matching terms

Home About Collections Help Feedback Accessibility Take-Down Policy Privacy Contact

But, with the pandemic, Hathi opened up what they call their Emergency Temporary Access Service, or ETAS.

This is an implementation of what's called "Controlled Digital Lending."

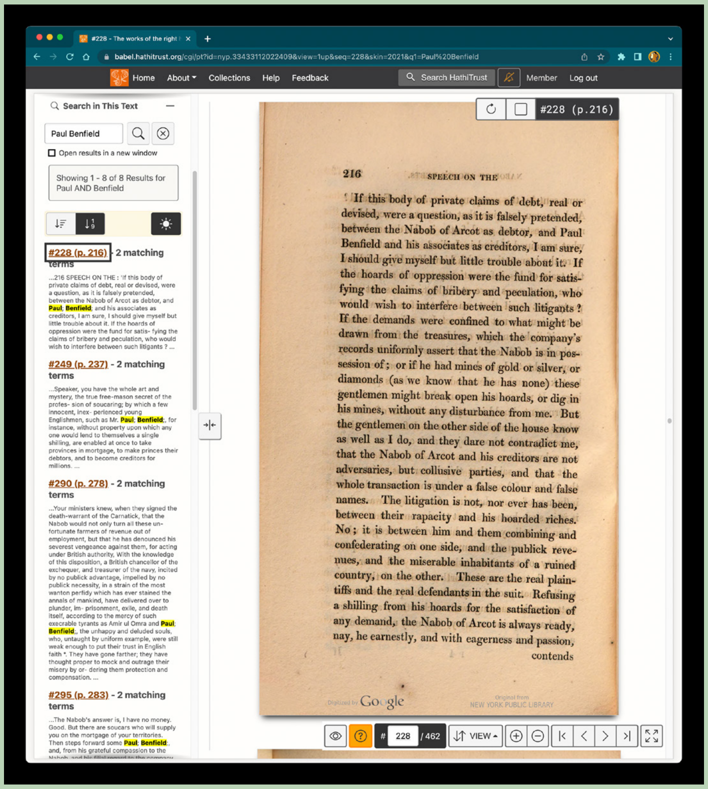
HathiTrust

collaborative of academic and research libraries

17+ million digitized items (4.6 million from UC)

reading access limited by US copyright law

pandemic → ETAS



#228 - The works of the right

babel.hathitrust.org/cgi/pt?id=nyp.33433112022409&view=1up&seq=228&skin=2021&q1=Paul%20Benfield

Home About Collections Help Feedback Search HathiTrust Member Log out

Search in This Text

Paul Benfield

Open results in a new window

Showing 1 - 8 of 8 Results for Paul AND Benfield

#228 (p. 216) - 2 matching terms

...216 SPEECH ON THE : 'If this body of private claims of debt, real or devised, were a question, as it is falsely pretended, between the Nabob of Arcot as debtor, and Paul; Benfield; and his associates as creditors, I am sure, I should give myself but little trouble about it. If the hoards of oppression were the fund for satis- fying the claims of bribery and peculation, who would wish to interfere between such litigants ? ...

#249 (p. 237) - 2 matching terms

...Speaker, you have the whole art and mystery, the true free-mason secret of the profes- sion of soucaring; by which a few innocent, inex- perienced young Englishmen, such as Mr. Paul; Benfield;, for instance, without property upon which any one would lend to themselves a single shilling, are enabled at once to take provinces in mortgage, to make princes their debtors, and to become creditors for millions. ...

#290 (p. 278) - 2 matching terms

...Your ministers knew, when they signed the death-warrant of the Carnatick, that the Nabob would not only turn all these un- fortunate farmers of revenue out of employment, but that he has denounced his severest vengeance against them, for acting under British authority, With the knowledge of this disposition, a British chancellor of the exchequer, and treasurer of the navy, incited by no publick advantage, impelled by no publick necessity, in a strain of the most wanton perfidy which has ever stained the annals of mankind, have delivered over to plunder, im- prisonment, exile, and death itself, according to the mercy of such execrable tyrants as Amir ul Omra and Paul; Benfield;, the unhappy and deluded souls, who, untaught by uniform example, were still weak enough to put their trust in English faith *. They have gone farther; they have thought proper to mock and outrage their misery by or- dering them protection and compensation. ...

#295 (p. 283) - 2 matching terms

...The Nabob's answer is, I have no money. Good. But there are soucars who will supply you on the mortgage of your territories. Then steps forward some Paul; Benfield;, and, from his grateful compassion to the

#228 (p.216)

216 SPEECH ON THE

'If this body of private claims of debt, real or devised, were a question, as it is falsely pretended, between the Nabob of Arcot as debtor, and Paul Benfield and his associates as creditors, I am sure, I should give myself but little trouble about it. If the hoards of oppression were the fund for satisfying the claims of bribery and peculation, who would wish to interfere between such litigants? If the demands were confined to what might be drawn from the treasures, which the company's records uniformly assert that the Nabob is in possession of; or if he had mines of gold or silver, or diamonds (as we know that he has none) these gentlemen might break open his hoards, or dig in his mines, without any disturbance from me. But the gentlemen on the other side of the house know as well as I do, and they dare not contradict me, that the Nabob of Arcot and his creditors are not adversaries, but collusive parties, and that the whole transaction is under a false colour and false names. The litigation is not, nor ever has been, between their rapacity and his hoarded riches. No; it is between him and them combining and confederating on one side, and the publick revenues, and the miserable inhabitants of a ruined country, on the other. These are the real plaintiffs and the real defendants in the suit. Refusing a shilling from his hoards for the satisfaction of any demand, the Nabob of Arcot is always ready, nay, he earnestly, and with eagerness and passion,

contends

Digitized by Google

Original from NEW YORK PUBLIC LIBRARY

The idea behind controlled digital lending is pretty simple. A library buys a paper book, it's a physical object, we can only lend it to one person at a time. They borrow it for a limited period, they bring it back, we can lend it to someone else. Under controlled digital lending, we scan that paper book, we take it out of circulation — this is important — and we provide that scan, again just to one person at a time, for a limited period. If we have take one physical copy of the book out of circulation, we can let one person at a time see the digital copy; if we take two copies out, two people, and so on. This is the "controlled" part. It's not a free-for-all, we're not just giving PDFs away to anyone who wants one.

But, OK, it's a legal gray area. Publishers, obviously, they'd like to collect a nickel every time you look at a book, or better yet a dollar. Maybe ten dollars, if it's a textbook. We think, I mean a lot of librarians think, and UC's lawyers and the lawyers at a lot of other libraries and universities think, this is a pretty straightforward extension of what libraries do already, what they've been doing for hundreds of years. It's morally defensible, it's legally defensible. But it's never been litigated.

So this is where we are, right now. April 2020, Berkeley joins the Hathi Emergency Temporary Access Service, or ETAS. And now if I'm affiliated with Berkeley, I can see the full text of all those in-copyright Berkeley books we've uploaded to Hathi — again on a one-to-one basis, again for a limited time. And of course everything's out of circulation, because the library's closed.

Controlled Digital Lending

- digital equivalent of traditional lending
- one physical copy = one digital reader
- digitized files secure against copying
- uncharted legal territory

So summer of 2020, we start coming back to campus in a limited capacity, masks, social distancing, one person in a room at a time, you all remember, that kind of thing.

And our Imaging Services team is able to come back and start scanning books for fall course reserves.

Timeline

March 2020	Berkeley campus (& libraries) closed
April 2020	HathiTrust opens ETAS
July 2020	Imaging Services returns to campus

We upload those to Hathi ETAS, and by the time classes start in the fall of 2020—which, remember, as far as students are concerned, we're still almost entirely remote at this point—we have all our course reserves in there.

We'd already been scanning stuff and sending it to Hathi for years, of course, bit by bit, but this was a whole 'nother level of effort.

Timeline

March 2020	Berkeley campus (& libraries) closed
April 2020	HathiTrust opens ETAS
July 2020	Imaging Services returns to campus
September 2020	Fall course reserves in HathiTrust

And that keeps going into spring 2021.

But now it's spring. COVID cases are way down from the winter peak, we've barely heard of beta and gamma, let alone delta, omicron hasn't even happened yet, and we've decided we're opening campus back up in the fall.

Now the thing with Hathi ETAS, it's only available as long as the library's closed to the public, as long as nobody can get at the books.

Timeline

March 2020	Berkeley campus (& libraries) closed
April 2020	HathiTrust opens ETAS
July 2020	Imaging Services returns to campus
September 2020	Fall course reserves in HathiTrust
March 2021	Berkeley plans fall re-opening

So in spring 2021 we can see that we're going to lose access to it.

But we're still going to have a lot of remote classes and a lot of remote students.

Hathi wasn't built as a controlled digital lending system. They're not set up for the complexity of, OK, it's just these specific titles, this many copies of each one, that's what we've pulled from circulation, that's what we want you to make available.

So if the UC Berkeley Library wants a controlled digital lending system, we're going to have to run it ourselves.

Timeline

March 2020	Berkeley campus (& libraries) closed
April 2020	HathiTrust opens ETAS
July 2020	Imaging Services returns to campus
September 2020	Fall course reserves in HathiTrust
March 2021	Berkeley plans fall re-opening
September 2021 (predicted)	Berkeley to lose Hathi ETAS access; alternative system needed

Now, as I said, this is a legal gray area, and before the pandemic, it's something libraries were talking about, but it's not something people were actually doing at scale.

So at that time, spring/summer 2021, it's not like we could just call up our library services vendor and say, hey, we want to pay for your controlled digital lending add-on.

(That's starting to change now, by the way, but it's still early days.)

So there wasn't something we could literally buy. But there were a few schools out there that were facing the same issues we were, that weren't part of HathiTrust, and were a bit ahead of us on the implementation.

So we thought, there might be something we could use there.

build or buy?

The constraints we were under, though, we only had a few months to do this, we didn't have many people available to work on it, and it absolutely had to be up and working by fall.

Constraints

short timeline	just 4-5 months till launch
minimal resources	only 1-2 developers available
mission critical	no course reserves = no classes

The first system we looked at was one called G-CDL, from Fordham University.

They'd done a presentation to a controlled digital lending interest group.

They had it up and running.

There was one guy working on it, and he was interested in collaborating.

Fordham G-CDL

advantages	working example
	developer interested in collaborating

So we worked with him to get a test version of it up and running in our infrastructure, and we quickly ran into a lot of issues.

It wasn't easy to Dockerize — it hadn't been designed to run outside Fordham's infrastructure.

It was PHP, and in Berkeley Library IT we're mostly a Ruby shop, historically a lot of Perl but lately mostly Ruby, Python mostly on the DevOps / infrastructure side. A bit of PHP for the Drupal CMS that runs the library website, but not really a deep bench there.

It depended on Google Drive to store the files, which meant some file size limitations, there were some unresolved issues around personal and institutional accounts, Google terms of service, SSO was going to need a lot of work, we were just going to need to do a lot of work in general.

Again, on an unfamiliar platform, and Fordham only had one developer working on it — who's since left, I hear, by the way. So that's been tricky for a couple of libraries that did decide to go with G-CDL.

Fordham G-CDL

disadvantages

unfamiliar tech stack (PHP/Angular)

PDF-based workflow

scalability limited by Google Drive

unresolved SSO & security issues

still in early alpha, extensive development needed

But we didn’t.

The next system we looked at was one called DIBS, from CalTech. Compared to what we’d been looking at with G-CDL, this was a pretty mature system.

They’d done a pretty good job of thinking through the requirements, it looked pretty slick, and they were using IIIF — that’s the International Image Interoperability Framework, for those who don’t know, it’s a set of open standards for using web services to deliver high-quality digital images in a scalable way, and I’ll be talking more about how it works for us later — which seemed like it would be a better fit for our existing scanning workflow than Fordham’s PDF, Google Drive setup.

Compared to G-CDL it was pretty easy to get up and running.

CalTech DIBS

advantages

solid requirements analysis

relatively mature code

IIIF-based design fits Berkeley ecosystem

But there were still some issues.

The core web app was solid, but a lot of the integration points — like the way they got metadata into the system, the SSO setup, stuff like that — were CalTech-specific — I mean, not surprising, and they'd done a pretty good job making it modular, so it wouldn't be that hard to replace them, but we'd need to write our own. And again, not our tech stack — less of a barrier because we do have, I mean I personally had and the team has, some Python experience, and also because it's Python and not PHP, but still a speed bump.

And another speed bump, this sounds like a small thing, but they weren't following the PEP-8 style guide. Not that we're dogmatic about that sort of thing, but it meant that anybody who opened a source file, their editor was going to want to reformat it, it meant we couldn't use off-the-shelf code quality tools without extra configuration, and given the short timeline it was just something that was going to be a pain to take on.

And then, one thing we are pretty dogmatic about is tests. All our Rails apps, all the ones we've written in the last 3-4 years anyway, have 100% test coverage. (Line coverage, anyway, path coverage or even branch coverage would be nice, but we're not there yet.)

And again, not only were we on a short timeline, but this was mission-critical stuff So step one, before we started hacking on it, would have been to get the whole thing under test. And with the timeline we were on, it just didn't seem feasible.

CalTech DIBS

disadvantages	import workflow tied to TIND ILS
	CalTech-specific Shibboleth SSO
	unfamiliar tech stack (Python/Bottle)
	nonstandard coding style
	no automated tests

On the other hand, if we built our own in-house system, we had a lot of code we could leverage,

we could tailor it to fit our existing infrastructure and make it play well with the workflow we'd already worked out for getting stuff into Hathi,

and we could do the whole thing with full test coverage and automated code quality checks from the get-go, so we could have full confidence in what we were shipping.

And, honestly, it was really helpful to have looked at Fordham's G-CDL and especially at CalTech's DIBS and the requirements analysis they'd done there, that gave us something to riff off of, really helped us get a concrete idea of what it was we wanted to build, some similarities and some differences.

In-house system

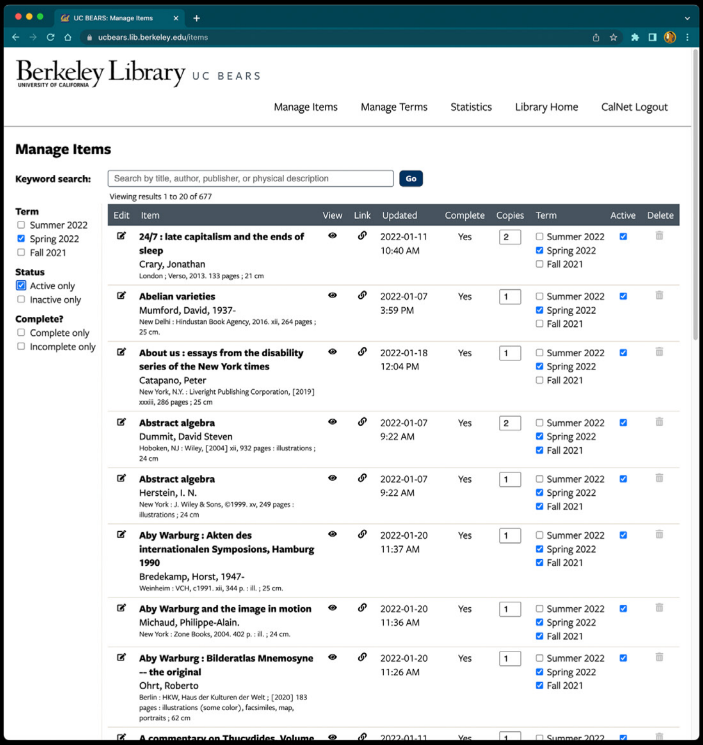
advantages	leverage existing Rails code (e.g. SSO)
	leverage existing digital asset storage
	leverage Hathi digitization workflow
	test-driven development

At this point you're probably tired of looking at slides, so let's take a little break.

demo

I'm going to skip ahead and show you some working software, so you have some context when I go back and talk about the architecture and the development process and some of the issues we ran into. there.

Demo



UC BEARS: Manage Items

ucbears.lib.berkeley.edu/items

Berkeley Library UNIVERSITY OF CALIFORNIA UC BEARS

Manage Items | Manage Terms | Statistics | Library Home | CalNet Logout

Manage Items

Keyword search: Search by title, author, publisher, or physical description **Go**

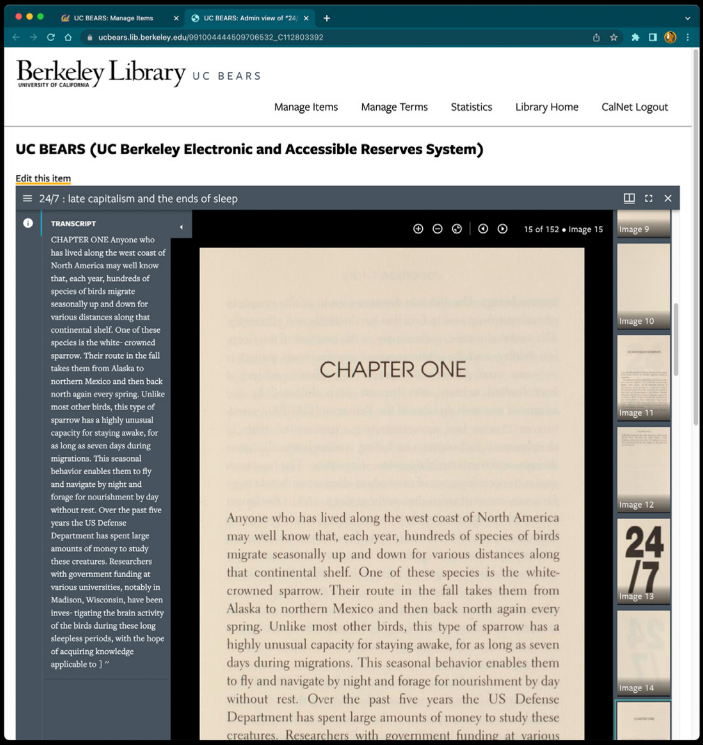
Viewing results 1 to 20 of 677

Term
- ☐ Summer 2022
- ☑ Spring 2022
- ☐ Fall 2021

Status
- ☑ Active only
- ☐ Inactive only

Complete?
- ☐ Complete only
- ☐ Incomplete only

Edit	Item	View	Link	Updated	Complete	Copies	Term	Active	Delete
	24/7 : late capitalism and the ends of sleep Crary, Jonathan London ; Verso, 2013. 133 pages ; 21 cm			2022-01-11 10:40 AM	Yes	2	☐ Summer 2022 ☑ Spring 2022 ☐ Fall 2021	☑	
	Abelian varieties Mumford, David, 1937- New Delhi : Hindustan Book Agency, 2016. xii, 264 pages ; 25 cm.			2022-01-07 3:59 PM	Yes	1	☐ Summer 2022 ☑ Spring 2022 ☐ Fall 2021	☑	
	About us : essays from the disability series of the New York times Catapano, Peter New York, N.Y. : Liveright Publishing Corporation, [2019] xxxiii, 286 pages ; 25 cm			2022-01-18 12:04 PM	Yes	1	☐ Summer 2022 ☑ Spring 2022 ☐ Fall 2021	☑	
	Abstract algebra Dummit, David Steven Hoboken, NJ : Wiley, [2004] xii, 932 pages : illustrations ; 24 cm			2022-01-07 9:22 AM	Yes	2	☐ Summer 2022 ☑ Spring 2022 ☑ Fall 2021	☑	
	Abstract algebra Herstein, I. N. New York : J. Wiley & Sons, ©1999. xv, 249 pages : illustrations ; 24 cm			2022-01-07 9:22 AM	Yes	1	☐ Summer 2022 ☑ Spring 2022 ☑ Fall 2021	☑	
	Aby Warburg : Akten des internationalen Symposions, Hamburg 1990 Bredekamp, Horst, 1947- Weinheim : VCH, c1991. xii, 344 p. : ill. ; 25 cm.			2022-01-20 11:37 AM	Yes	1	☐ Summer 2022 ☑ Spring 2022 ☑ Fall 2021	☑	
	Aby Warburg and the image in motion Michaud, Philippe-Alain. New York : Zone Books, 2004. 402 p. : ill. ; 24 cm.			2022-01-20 11:36 AM	Yes	1	☐ Summer 2022 ☑ Spring 2022 ☑ Fall 2021	☑	
	Aby Warburg : Bilderatlas Mnemosyne -- the original Ohrt, Roberto Berlin : HKW, Haus der Kulturen der Welt ; [2020] 183 pages : illustrations (some color), facsimiles, map, portraits ; 62 cm			2022-01-20 11:26 AM	Yes	1	☐ Summer 2022 ☑ Spring 2022 ☑ Fall 2021	☑	
	A commentary on Thucydides. Volume			2022-01-11	Yes	1	☐ Summer 2022	☑	



UC BEARS: Manage Items | UC BEARS: Admin view of "24/

ucbears.lib.berkeley.edu/991004444509706532_C112803392

Berkeley Library UNIVERSITY OF CALIFORNIA UC BEARS

Manage Items | Manage Terms | Statistics | Library Home | CalNet Logout

UC BEARS (UC Berkeley Electronic and Accessible Reserves System)

Edit this item

24/7 : late capitalism and the ends of sleep

15 of 152 • Image 15

TRANSCRIPT

CHAPTER ONE Anyone who has lived along the west coast of North America may well know that, each year, hundreds of species of birds migrate seasonally up and down for various distances along that continental shelf. One of these species is the white- crowned sparrow. Their route in the fall takes them from Alaska to northern Mexico and then back north again every spring. Unlike most other birds, this type of sparrow has a highly unusual capacity for staying awake, for as long as seven days during migrations. This seasonal behavior enables them to fly and navigate by night and forage for nourishment by day without rest. Over the past five years the US Defense Department has spent large amounts of money to study these creatures. Researchers with government funding at various universities, notably in Madison, Wisconsin, have been inves- tigating the brain activity of the birds during these long sleepless periods, with the hope of acquiring knowledge applicable to] "

CHAPTER ONE

Anyone who has lived along the west coast of North America may well know that, each year, hundreds of species of birds migrate seasonally up and down for various distances along that continental shelf. One of these species is the white-crowned sparrow. Their route in the fall takes them from Alaska to northern Mexico and then back north again every spring. Unlike most other birds, this type of sparrow has a highly unusual capacity for staying awake, for as long as seven days during migrations. This seasonal behavior enables them to fly and navigate by night and forage for nourishment by day without rest. Over the past five years the US Defense Department has spent large amounts of money to study these creatures. Researchers with government funding at various

So that's UC BEARS in action.

Now let's look at what's going on behind the scenes — the design of the system, the architecture and the workflow that gets those scanned books to the screen.

design

So, top center here, you have the Rails web app, UC BEARS itself, running in our Docker swarm, which serves up the pages we were looking at there. (It also serves up the JSON APIs behind that admin interface, though I'm not showing that in the diagram here.)

Over on the right, you've got that viewer web page, with the Mirador IIIF viewer embedded in it — that's a React component, developed mostly out of Stanford, but used a lot of places. CalTech's DIBS uses one called Universal Viewer, which we've used for other things, but for reasons I'll go into later we went with Mirador for UC BEARS.

The way Mirador works — the way any IIIF viewer works — you give it a IIIF manifest, which is a JSON linked data file, and it pulls whatever images it needs to show the part of the page you're looking at, at whatever zoom level. And if you're only looking at part of the page, it only asks for that part, so it's pretty efficient.

And the IIIF manifest can also include metadata about the image, annotations or whatever, and in this case we're using it for the OCR text.

A side benefit of using IIIF is it's nontrivial to stitch all the images back together. It's not impossible, we're not talking BluRay DRM and HDMI copy protection here, but it's way more trouble than it's worth. And it's not like you can't screenshot your ebook reader or PDF viewer or whatever the alternative would be. So that ticks the box for controlled digital lending.

Then at the bottom center, there's the image server — we use one called IIPImage, but anything that speaks IIIF would work. It's implemented as a web server plugin, in this case Nginx, which again we're running in Docker.

What IIPImage reads is what are called tiled TIFFs or pyramidal TIFFS, one for each scanned page — this is a special flavor of TIFF file that has multiple resolutions and is structured so the image server can pull out parts of the image as needed, again depending on where you're scrolling or zooming or whatever.

Those are served up off the filesystem, in this case our NetApp storage array, which is what we use for most of our storage.

There's a UC BEARS volume that's mounted to both the image server container and the Rails container.

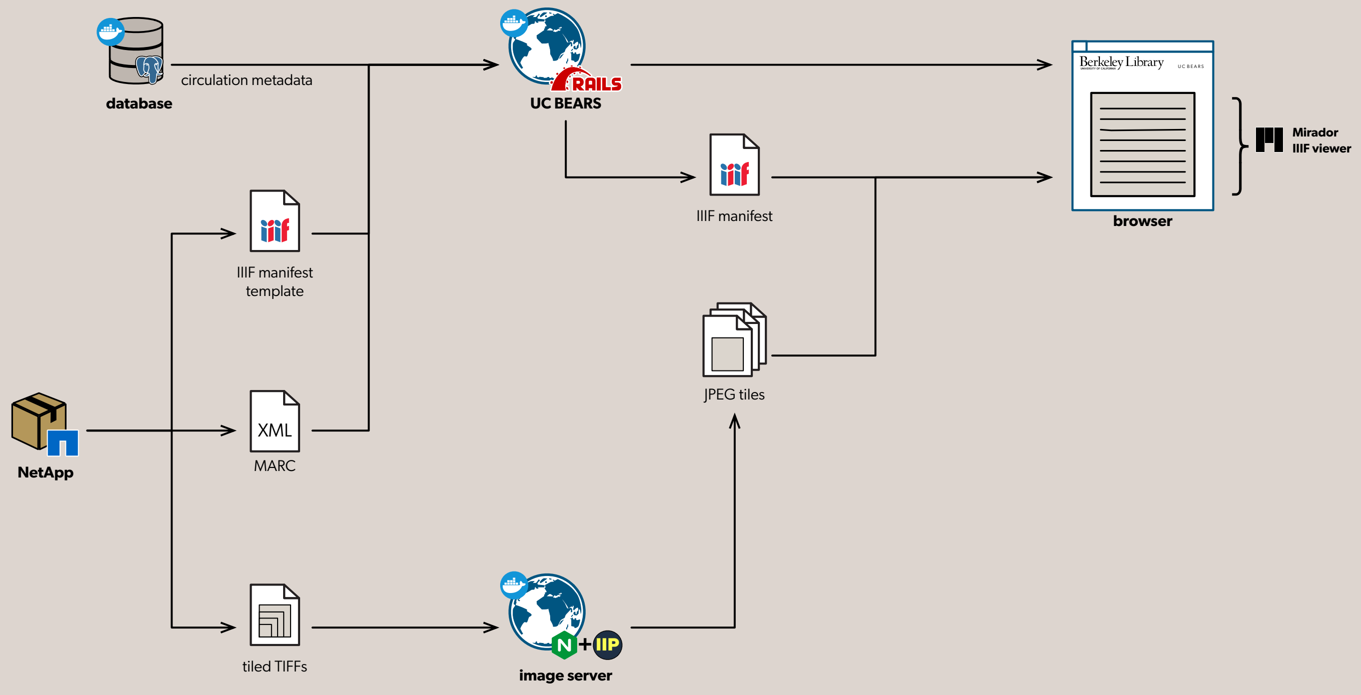
In there, for each scanned book, besides the TIFFs for all the pages, we also have a MARC file—that's machine-readable cataloging, it's a library thing—that has all the metadata for the book, the author, title, publisher and so on—and we have a template for the IIIF manifest. It's not the actual IIIF manifest because the manifest needs to have URLs in it, and we don't want to hard-code those, we want to inject them at run-time.

So the Rails app reads that MARC XML file, and it reads the image manifest.

And it also talks to this Postgres database, up ihere in the upper left, again running in our Docker swarm, and that keeps track of things like when was which book checked out, when's it due, is it active or inactive, if it's active for which term, all that stuff we saw in the admin interface.

So that's the architecture.

Architecture



Then what we've got here is the process for getting data into the system.

One of the things we wanted to avoid was having to keep track of a bunch of state as we processed stuff into the system.

So the workflow is structured around this set of magic directories.

Under the UC BEARS directory, you've got "upload", "ready", "processing", and "final". Under that, depending on the state an item is in — where by "item" I mean "book", basically — you'll have a directory for each item, named with the item's bibliographic record number and item barcode. The item barcode, that's the actual barcode number that's on a sticker that's on the book, and the bibliographic record number, that's what we use to pull the MARC record that has the author and title and so on. So these are guaranteed to be unique.

An item starts here on the left in "upload" — this is kind of a scratch space for the staff member that's collecting the scanned images, running the OCR software on them, pulling the MARC record, all the manual processing that has to be done before the item can go into UC BEARS — which is pretty much what we were already doing for Hathi.

When the folder's complete, when it has all the images and all the ancillary files — the staff member moves it from "upload" to "ready".

This is a move, not a copy, because this is all happening over NFS, and we want it to be atomic.

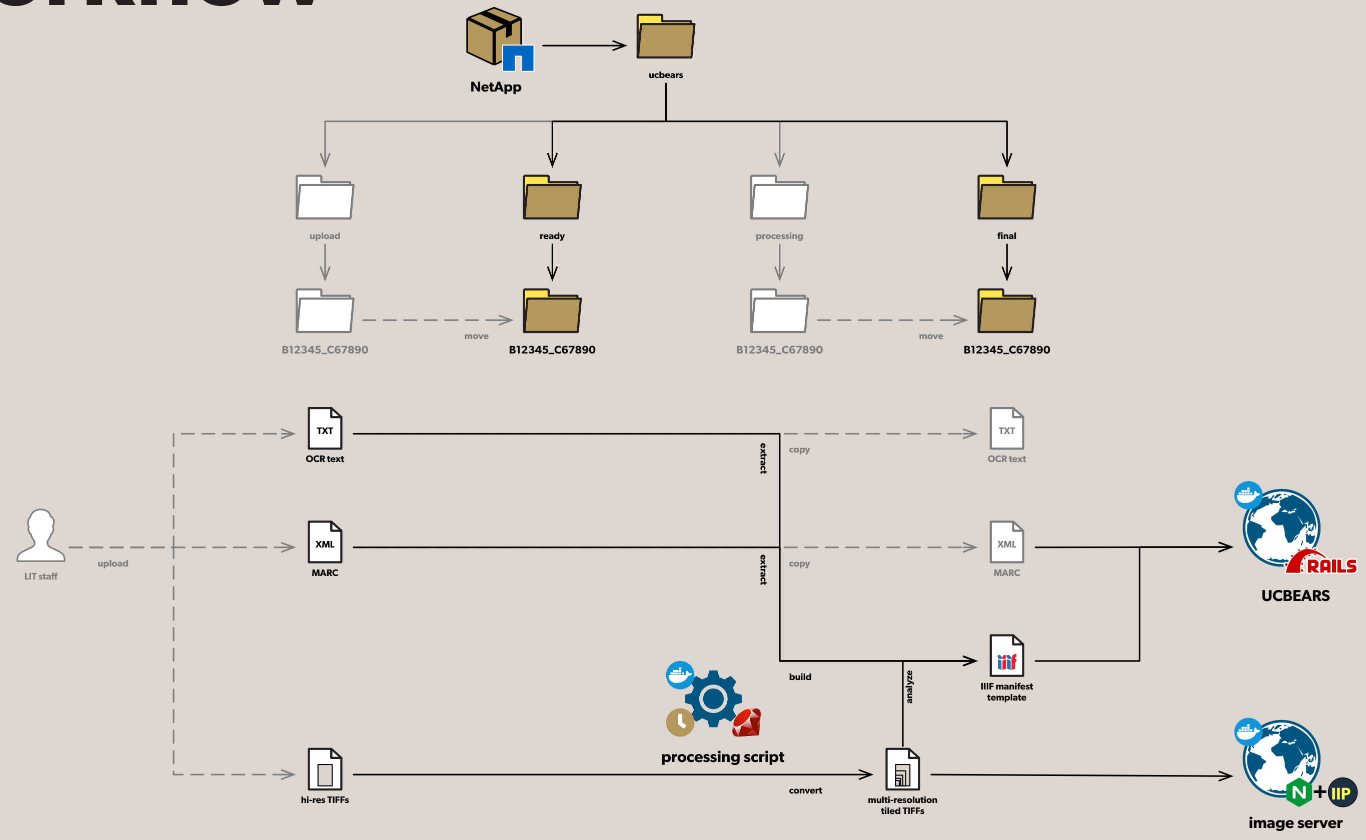
We want it to be atomic, because there's a processing script running on a timer — you can think of it as a cronjob, basically, although it's actually in the Docker stack configuration — which is polling that "ready" folder, looking for new item directories.

It knows they're "new" if there isn't a corresponding directory in "processing" or "final". So we don't do any extra bookkeeping there, the state of the filesystem is our single source of truth.

When the processing script kicks off, for each "ready" directory, it creates a corresponding directory in "processing". And that's where it puts the IIIF manifest template — which includes the OCR text — and the pyramidal tiled TIFFs. Generating those is the most time-consuming part of the process, by the way, it can be like seconds per page. We use a library called VIPS for that.

Then when the processing script's all done with an item directory, it does another atomic move, from "processing" to "final", where it's available to the UCBEARS Rails app and to the image server.

Workflow



Now I’m going to talk a little bit about the development process and some of the pitfalls we ran into.

This is my chance to complain, basically.

development

So when we started to build out our first UC BEARS prototype, we wanted to hit the areas with the most technical risk first. The really critical piece of UC BEARS is the viewer, right, without that we don't have an app. The rest of it, that's garden-variety web app stuff, we know we know how to do that.

But we're looking at a really short timeline and we don't know how hard the viewer is going to be, and we don't want to spend even a week spinning up a new app and then find out we really needed that week at the end of the summer when it's time to go live.

And we had, we have, this existing Rails app we use for miscellaneous forms, mostly terms and conditions, but also some random things like a staff interface for downloading metadata from our digital asset management system, basically any small thing that might need to send email, or that we might need to put behind a campus single-sign-on page, or whatver. And it's pretty easy to just add a new set of model-view-controller classes and just add a new independent piece of functionality. So that's what we decided to do for our first UC BEARS prototype. If we had time to spin it out into a standalone app, great, but if we could just get the viewer working in this forms app, if we had to, we could go live with that in the fall.

The only thing, though, is that the forms app was still on Rails 5 — Rails 6 had been out for about two years, but we just hadn't got around to upgrading it, partly because we knew from other apps that the 5 to 6 upgrade was kind of a pain.

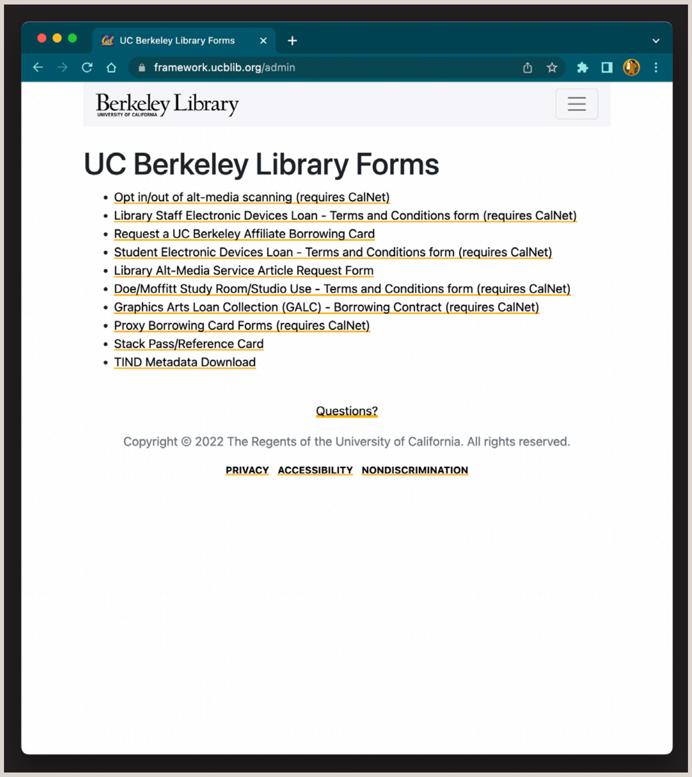
Initial Prototyping

Library Forms

existing app with CI/CD pipeline, SSO

easy to add new MVC set

Rails 5



UC Berkeley Library Forms

- Opt in/out of alt-media scanning (requires CalNet)
- Library Staff Electronic Devices Loan - Terms and Conditions form (requires CalNet)
- Request a UC Berkeley Affiliate Borrowing Card
- Student Electronic Devices Loan - Terms and Conditions form (requires CalNet)
- Library Alt-Media Service Article Request Form
- Doe/Moffitt Study Room/Studio Use - Terms and Conditions form (requires CalNet)
- Graphics Arts Loan Collection (GALC) - Borrowing Contract (requires CalNet)
- Proxy Borrowing Card Forms (requires CalNet)
- Stack Pass/Reference Card
- TIND Metadata Download

Questions?

Copyright © 2022 The Regents of the University of California. All rights reserved.

PRIVACY ACCESSIBILITY NONDISCRIMINATION

And why was it a pain? So the Rails philosophy is kind of to hide as much as possible from the developer. Okay, I'll be charitable, they're trying to make things as easy as possible for the developer, to let you focus on your business logic and so on.

So there's a lot of behind-the-scenes magic, so your Rails developer can do all this JavaScript-y stuff without having to learn any JavaScript, they just do everything in Ruby and the JavaScript gets generated behind the scenes. But that also means Rails reinvents a lot of wheels, trying to do everything the Rails way, while the rest of the world moves on.

So one of those wheels they reinvented was JavaScript and CSS minification and SASS compilation and asset fingerprinting. And I could go into a whole rant about that, but I'll spare you.

But at least for five or ten years they had a pretty consistent story for that, which was this library called Sprockets.

Then about three years ago they decided to replace Sprockets with Webpacker, which was a wrapper around the Webpack Node.js module bundler. Only Webpack's SASS compilation performance was just awful, so for a while the official recommendation was that you use Webpack for JavaScript and keep Sprockets for CSS. Even though Sprockets was dependent on LibSass, which is deprecated, and had a bunch of other issues, and wasn't really under active development.

Then in Rails 7 they tossed out Webpacker in favor of a new collection of wrapper libraries that are supposed to be agnostic about your underlying JavaScript or CSS tooling, but they didn't put any of this in the Rails release notes, just in posts on David Heinemeier Hansson's blog. I guess they thought it wasn't important enough, I don't know.

And anyway this is all probably heading in the right direction, but as far as we're concerned it's too late. At this point we've kind of given up on Rails for the front end.

And that's the direction we're going for the future, the back end API is still going to be served up by a Rails app, because we have a lot of Ruby expertise, but the front end is going to be JavaScript components. UC BEARS might be the last traditional Rails app we do.

And as you've seen it's already kind of a hybrid — there's still HTML pages being generated from ERBs, but the most important parts of the app, the interactive parts, are mostly JavaScript components, they're mostly Vue and React. Sort of a Jamstack direction, if you've heard about that.

Rails & JavaScript

Rails' ever-changing JavaScript / CSS / static asset story

Rails 3.1 through Rails 5: Sprockets

Rails 6: Node & Webpacker
(Maybe stick w/Sprockets for CSS. And LibSass. Which is deprecated.)

Rails 7: Node & ...stuff (jssbundling-rails + cssbundling-rails + *Sprockets? Propshaft?)*
(Release notes? Don't you read DHH's blog?)

But in that initial prototype, we were still in the forms app, we were still working with Rails 5 and Sprockets, and we needed to get that IIIF viewer working.

The first one we looked at was Universal Viewer, which started out at the Wellcome Library in the UK — it's what CalTech DIBS uses, as I mentioned, and it's a great viewer.

But I spent about three days banging my head against the wall trying to get it to work with Sprockets, and I gave up.

So we switched to Mirador, which is another great viewer, developed mostly at Stanford I think, and historically they do a lot of Rails over there, so they had a canned Rails integration that I was able to get working right away.

It was for Mirador 2, which was already pretty much end-of-lifed.

But Mirador 3, it's a modern JavaScript component like Universal Viewer, and we could just see we were going to run into all the same issues, and we were in a hurry.

So we decided to stick with Mirador 2 for the time being, since it was working, and in fact that's what we ended up going live with in the fall.

IIIF Viewer Integration

Universal Viewer	wouldn't play well with Sprockets
Mirador 2	easy install w/mirador_rails gem, but no longer supported
Mirador 3	not feasible w/o Node = not feasible w/o upgrade to Rails 6.1 + Webpacker

Then once we had Mirador working, we had a bunch of changes we needed to make to it. Because we're taking this image viewer that's really designed for galleries and museums, and we're trying to turn it into an e-book reader.

Mostly we just needed to turn features off. Like, you can see a typical Mirador setup here on the left, it's designed to be a full-screen or anyway full-window app, with multiple sub-windows, where you can freely add images from any URL, any IIIF URL, and look at them side by side, and we needed to turn that off. Annotation features, we didn't need those.

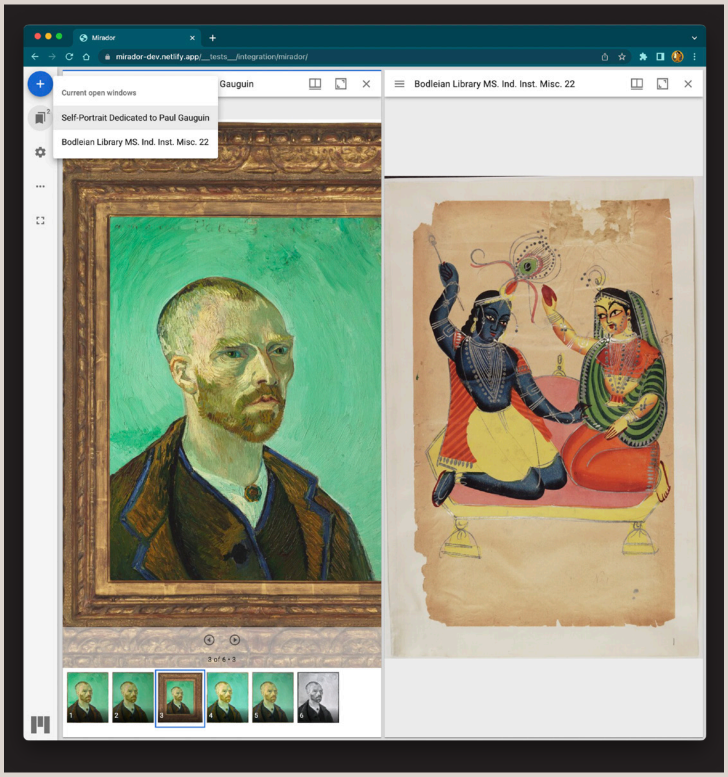
Then the navigation layout was really pretty good for an image viewer, but not ideal for our use case.

Anyway, stuff like that.

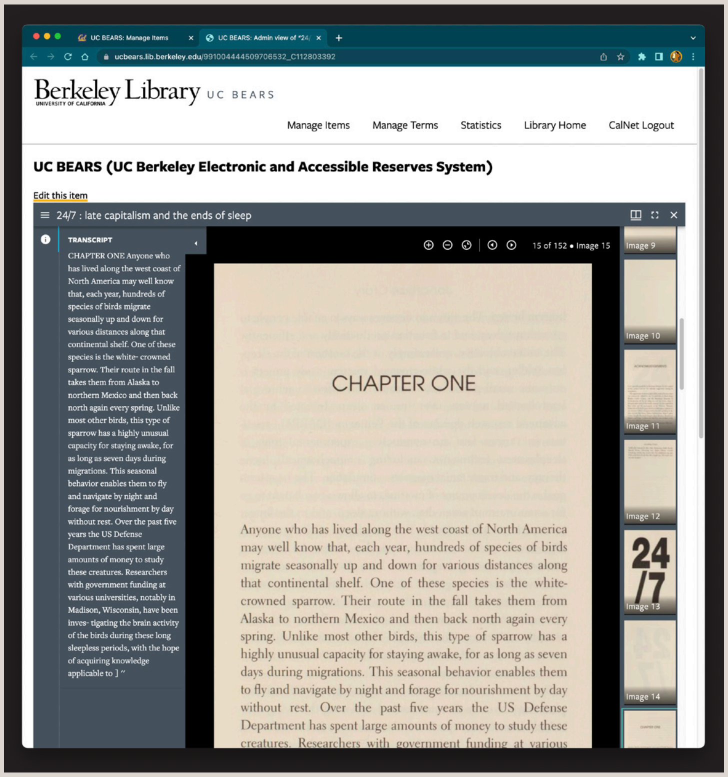
And Mirador actually has quite a few configuration options, we could turn off like 80% of the features we didn't need just with configuration, but that still left the other 20%.

Plus other customizations that they just didn't think of, or that they wouldn't think of because you only need them if you have a weird one-off use case like ours.

Customizing Mirador



Mirador
mirador-dev.netlify.app/__tests__/integration/mirador/
Current open windows
Self-Portrait Dedicated to Paul Gauguin
Bodleian Library MS. Ind. Inst. Misc. 22
Gauguin
3 of 6 • 3



UC BEARS (UC Berkeley Electronic and Accessible Reserves System)

24/7 : late capitalism and the ends of sleep

TRANSCRIPT

CHAPTER ONE Anyone who has lived along the west coast of North America may well know that, each year, hundreds of species of birds migrate seasonally up and down for various distances along that continental shelf. One of these species is the white- crowned sparrow. Their route in the fall takes them from Alaska to northern Mexico and then back north again every spring. Unlike most other birds, this type of sparrow has a highly unusual capacity for staying awake, for as long as seven days during migrations. This seasonal behavior enables them to fly and navigate by night and forage for nourishment by day without rest. Over the past five years the US Defense Department has spent large amounts of money to study these creatures. Researchers with government funding at various universities, notably in Madison, Wisconsin, have been inves- tigating the brain activity of the birds during these long sleepless periods, with the hope of acquiring knowledge applicable to] "

CHAPTER ONE

Anyone who has lived along the west coast of North America may well know that, each year, hundreds of species of birds migrate seasonally up and down for various distances along that continental shelf. One of these species is the white-crowned sparrow. Their route in the fall takes them from Alaska to northern Mexico and then back north again every spring. Unlike most other birds, this type of sparrow has a highly unusual capacity for staying awake, for as long as seven days during migrations. This seasonal behavior enables them to fly and navigate by night and forage for nourishment by day without rest. Over the past five years the US Defense Department has spent large amounts of money to study these creatures. Researchers with government funding at various

So Mirador's open source, but at that time, we didn't have a lot of contemporary JavaScript skills in-house, like ES6 and all that, and especially not React. Plus we were nervous about going too far in messing with the Mirador source code, because it's all under an Apache 2 license, and the Apache 2 license includes a patent grant, and that makes UC's IP people very nervous. I mean, we could do it, for internal use, but if we're going to put our time into open-source development, we'd really rather be working on stuff we can give back to the community.

So apart from a couple of places where we hooked into the React state lifecycle and stuff like that, we pretty much stayed out of the actual Mirador component, stayed out of JavaScript and we tried to do as much as we could with straight CSS. And that was actually pretty good for things like hiding pieces of the interface we didn't want, moving things around, changing some of the colors.

But it was a lot harder than it needed to be because Mirador was all built with Google's Material Design, and they kind of follow the Rails philosophy, we're going to hide everything from the developer that we think they don't need to know. And in this case what they're hiding is HTML and CSS, and so a lot of the CSS classes have these bonkers machine-generated names, and it was really nontrivial trying to find a semantically meaningful CSS class you could hang your changes on.

Sometimes I feel like I'm the only web developer in the world who actually likes CSS. Every framework that comes out tries to hide it under the rug, it seems like.

Except Vue, maybe. Which is why I like Vue. I want to thank the UCLA Library for turning me on to Vue, by the way.

Customizing Mirador

skills gap	strong HTML/CSS skills, some JS, very little React
OSS/IP policy	patent grant puts Apache 2 license in UC's "Moderate Risk" category
Material Design	nontrivial to customize look, feel, layout and accessibility in plain CSS

But anyway, we did it. We got it working, we got Mirador to behave pretty much the way we wanted. We got the rest of the pieces working, that content workflow with the magic directories, all that stuff.

We did end up pulling the UC BEARS code out of the library forms app into its own standalone application, as it got more and more complex — but by then we knew we had all the hard stuff figured out and we had time to do it.

In about two months from that first prototype we had something internal users could test, first in Library IT and then on the course reserves team that was actually going to be running this, and then in August we were ready to go to production.

The imaging services team's been busy scanning all this time, now the course reserves team can get them loaded and activated, with whole weeks to spare before students and faculty need to start looking at them.

Timeline

March/April 2021	G-CDL & DIBS evaluations
May 2021	prototype (add-on to forms app)
July 2021	alpha & beta testing
August 2021	first production release

And then that winter we had our first major update, we bit the bullet and updated to Rails 6 so we could upgrade to Mirador 3, we added that slick Vue.js-based admin interface—the initial production release was much more old-school, tables and forms. We made a bunch of accessibility improvements. Still some work to do there, and some of those same Mirador customization issues, but it's a lot better than it was. We added term support, so we could turn books off for Fall and on for Spring, and then an admin interface for terms so we could keep going into Summer and now Fall again.

Timeline

March/April 2021	G-CDL & DIBS evaluations
May 2021	first internal prototypes
July 2021	alpha & beta testing
August 2021	first production release
January 2022	first major update: Rails 6.1, Mirador 3, new Vue.js-based admin UI, accessibility improvements

And anyway it's pretty much been rolling, with minor tweaks, ever since.

Just a few stats—

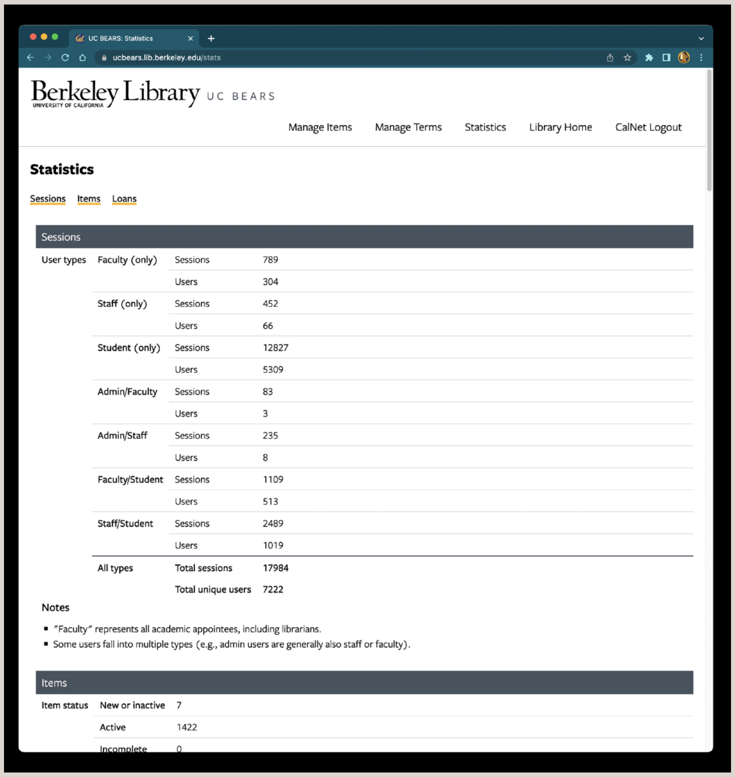
stats

More than 1400 items, almost 7000 student patrons, 22 thousand loans.

More going live tomorrow, I think, August 17th.

Stats

> 1,400	items
> 6,800	student patrons
> 22,000	loans



Statistics

Sessions Items Loans

Sessions			
User types	Faculty (only)	Sessions	789
		Users	304
	Staff (only)	Sessions	452
		Users	66
	Student (only)	Sessions	12827
		Users	5309
	Admin/Faculty	Sessions	83
		Users	3
	Admin/Staff	Sessions	235
		Users	8
	Faculty/Student	Sessions	1109
		Users	513
	Staff/Student	Sessions	2489
		Users	1019
	All types	Total sessions	17984
		Total unique users	7222

Notes

- "Faculty" represents all academic appointees, including librarians.
- Some users fall into multiple types (e.g., admin users are generally also staff or faculty).

Items		
Item status	New or inactive	7
	Active	1422
	Incomplete	0

And people have been pretty happy with it.

Stats

1 Chancellor's Outstanding Staff Award*

Chancellor's Outstanding Staff Award

presented to

David Moles

of the Library Course E-Reserves Design & Implementation Team

In recognition of exemplary contributions to the University's mission of teaching, research, and public service.

University of California, Berkeley
April 2022

Carol Christ
Carol Christ
Chancellor

* team award, Library Course E-Reserves Design and Implementation Team

That's a team award, by the way, it's the whole e-reserves team, it's not just me.

So I did most of the development, but I just want to give a shout-out to some of the other people that made this possible, we wouldn't be here without them.

Drew Facchiano, he's not with UC Berkeley any more, he took a job at UCSC Extension, but he did the first Fordham G-CDL evaluation and kind of got the ball rolling.

Chrissy Huhn and the imaging services team — they really stepped up through the whole pandemic, just amazing work, getting all this stuff scanned, and they're still at it.

My boss Lynne Grigsby, Alvin Pollock, one of our developers, Jim Lake — they're the ones who got those scans packaged up with the metadata and into the UC BEARS pipeline, just a tremendous amount of work.

Dan Schmidt and our DevOps team — the fact we were able to get this up and running so quickly is just a tribute to all the work they've done to modernize our infrastructure the last couple of years, Docker, the NetApp, the whole thing. Just outstanding work.

And finally our Associate University Librarian, Salwa Ismail, who's the one who came up with the clever acronym.

And that's just the software side, making the whole e-reserves thing work, acquiring the books, working with the faculty, all that took dozens of people. If any of you are listening, thank you.

Credits

G-CDL Evaluation	Drew Facchiano
Imaging	Chrissy Huhn & Imaging Services
Image & Metadata Processing	Lynne Grigsby, Alvin Pollock, James Lake
Infrastructure	Dan Schmidt & DevOps
Branding	Salwa Ismail

I think we have some time for questions.

questions?

Thanks, everybody, I really appreciate you giving me your time and attention, and I hope some of this has been useful, or inspirational.

thank you!

And if you want to know more, get in touch.

For more information

GitHub	BerkeleyLibrary/UCBEARS
contact	David Moles Head of Applications UC Berkeley Library dmoles@berkeley.edu