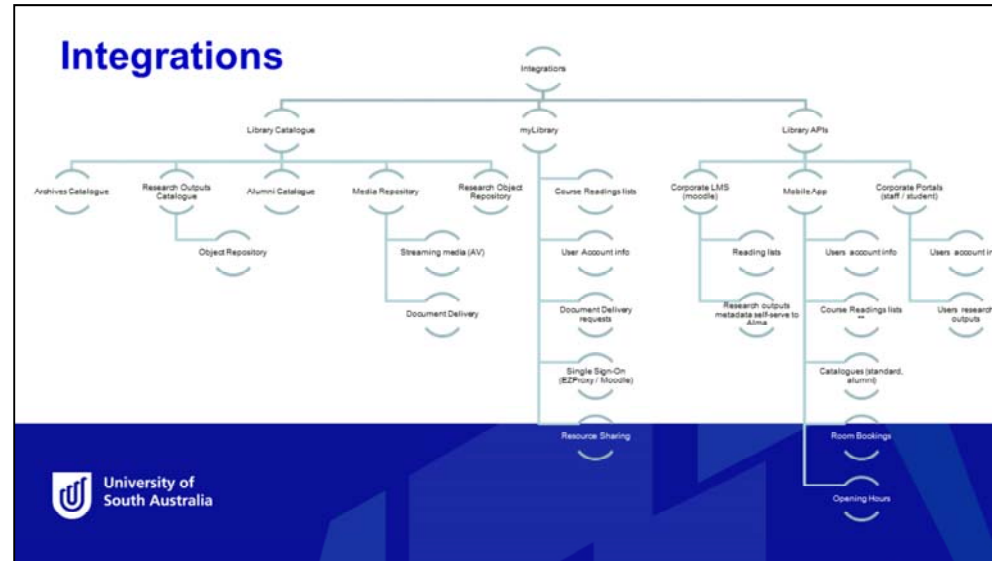


Hello, thanks for joining me!

I'm Ben Dalwood, software developer for the University of South Australia, Library.

This afternoon, I'll be talking about the various integrations we've achieved since our migration to our fully hosted Alma/Primo environment, and the lead up to that migration.



To begin, here’s a bit of an overview of the extent of the integrations.

I’ve shown it as hierarchical, as some integrations are subsequently integrations of others.

At the top, we have the two main systems in which we build our various integrations off of.

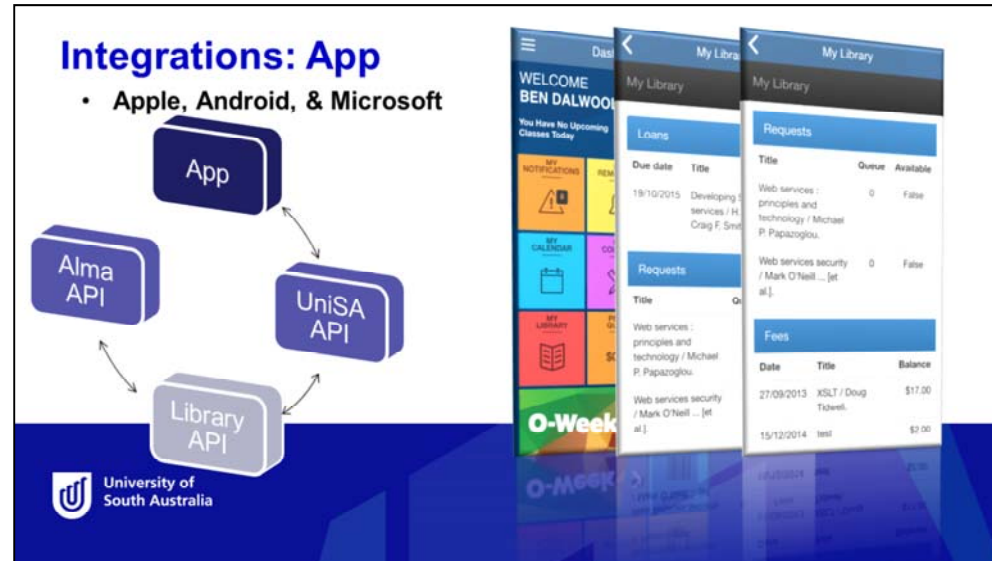
Firstly, the Libraries Catalogue, I’ll elaborate on the particulars later, but our catalogue interface has been developed from scratch using the Primo / Alma APIs. This has given us the flexibility to incorporate various integrations directly into it. Mostly, the other systems that hang off the catalogue, are other catalogues that are tailored to specific disciplines. For instance, we have an Alumni Specific catalogue which has only content in which alumni are allowed to access (under subscription licencing), we have our “digitool” replacement too, which we refer to as our Research Outputs Repository, this interface only searches on Research outputs, and interfaces with our document repository. Or Library Account service (myLibrary) also consists of several integrations within, mainly the users account info (loans, requests, fees) and just recently, we’ve added course readings and single sign-on to our ezproxy services too. We’ve begun transitions to document delivery and inter-library loans into that space too.

And lastly, our media repository, including access directly to our streaming media server and document repositories has been integrated within the discovery interfaces, providing on-page streaming of videos and audio and access to digitised materials we hold locally.

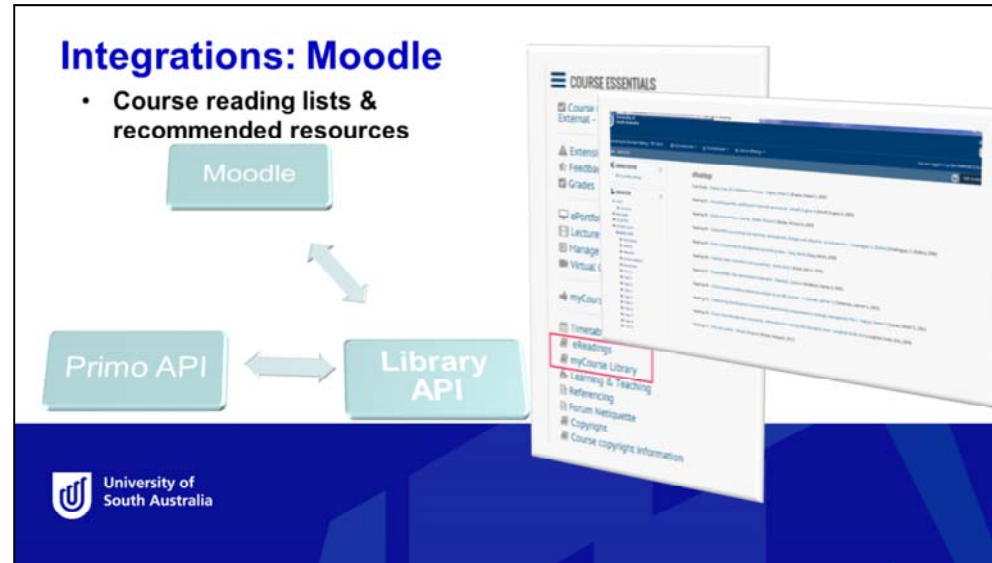
On the other side, we have our Library APIs, this is our all-in-one access to any and all data to and from our vendors, and our local data too. It's our main data layer.

Most of the integrations we've introduced go via these APIs.

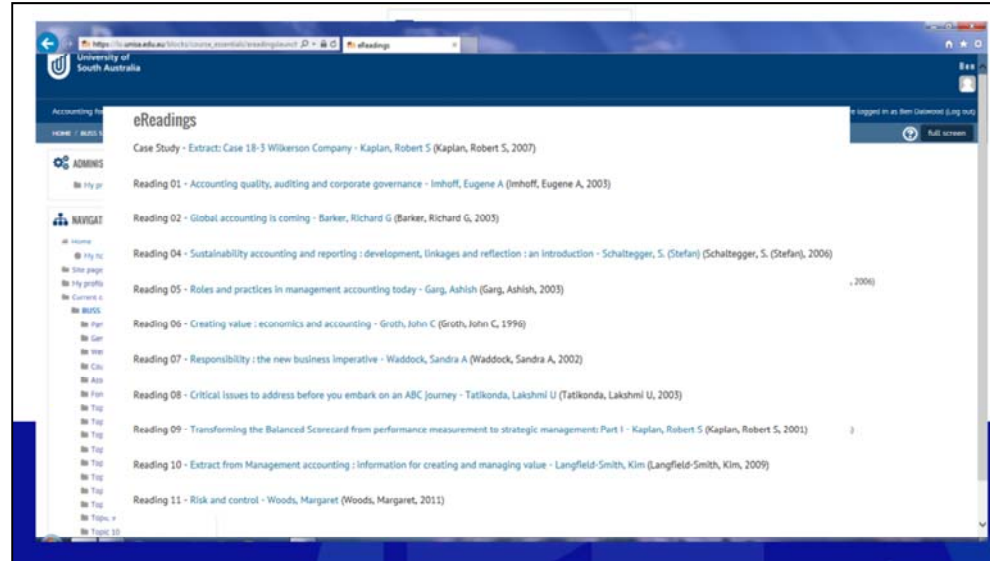
- We have the corporate learning management system (moodle), which communicates to our API for course reading lists, ereaders, streaming media, basically anything the course coordinator want's to include in their course from Alma.
 - On the corporate side, we also have the staff and student portals; within these portals, we've provided our APIs to show patron data from Alma; loans, request, fees, etc.
- Recently, the university had a mobile App created, and we've had those same integrations feeding into there too.
- Our most recent integration is a 'self service' research outputs submission form, this form talks to our API to create the BIB data directly into Alma, allowing the repository services team to review before it displays in discovery.



- Here we have the newest integration. The university commissioned some students to create a mobile App, “for students, by students”
- We were approached to integrate the library account data within the app, and since we created our data layer and APIs, it was a relatively painless process.
 - The one problem for us being, we didn’t want the APP to communicate with our API directly, we restrict our API to only specific IP / Subnets, and we did not want to open that up to the entire world.
 - So, in this case, the corporate uni team setup a web service proxy, which is the gateway for the App to all internal services, such as our API.
- We’ve only integrated a view only at this stage, so the loans, requests, and fees. We expect they’ll add the functionality to cancel requests, which we have built into our APIs already. And we’ll be offering them access to the course readings lists too.



- Our Corporate learning management system is based on Moodle. The corporate team and the course coordinators wanted a way to show all the readings and recommended resources for a course. For this one, we decided that a custom search to Primo would provide us with the most flexibility.
- What we did here, was add a custom search field in Primo index via back office that looks at the MARC field containing the course codes. This lets us execute our Primo API search using the course code, and returning the list of the relevant results.
- We provide the moodle team with a basic XML response that gives them the display, the target URL, and order in which to show. [pic on next slide]



Here's a bigger version, showing the display of the readings.

- The student would simply select the relevant link, they would then be taken to the catalogue [single sign-on] to us, and either presented with the digitised document or redirected to the subscription directly [single sign-on to ezproxy]

Integrations: Student Portal

- Loans, Request, Fees

```
graph TD; SP[Student Portal] <--> ALMA[Alma API]; SP <--> LIB[Library API]; ALMA <--> LIB;
```

The screenshot shows the University of South Australia student portal. The top navigation bar includes links for Home, my Current Studies, my Email, my Resources, my Academic Record, my Finances, and my Personal Details. Below this is a secondary navigation bar with icons for ASK IT, CAMPUS CENTRAL, LEARNING SUPPORT, LIBRARY, CAREER SERVICES, and USASA STUDENT ASSOCIATION. The main content area displays a 'Library Summary' section with the following data:

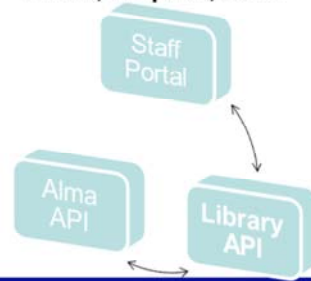
Library Summary	
Number of Loans:	0
Number of Requests:	0
Requests Available for Pickup:	0
Library Fees:	\$0.00

A link 'Login to myLibrary for more information' is visible at the bottom right of the summary section. The University of South Australia logo is in the bottom left corner of the slide.

- Our Corporate student portals have always had some library data presence, but it was always a direct database query to voyager. We created our APIs whilst still on our local voyager install, so when we had migrated off of voyager, we'd have no disruptions or changes during or after migration. They only show a summary of the account, and any alerts such as recalls, or requests ready to pickup.

Integrations: Staff Portal

- Loans, Request, Fees



A screenshot of the Library Summary page in the Staff Portal. The page displays a summary of library statistics and a list of requests.

Title	Queue Position	Location	Cancel
How services, processes and technology: Michael P. Fitzgerald	1	MTS-ADMINS	[Cancel]
How services security: Mark Chaffin - [et al.]	4	MTS-ADMINS	[Cancel]

Title	Request
How	DS
How	DS
How	DS
How	DS

- Similarly, Our Corporate staff portals, list of the same info, with the addition of the full listing of data rather than just a summary.

Integrations: Discovery

- **Streaming Media**

Primo API

Alma Uresolver

Catalogue

Streaming Server

Series 1. Ep. 10. Penguins

Online:

Series 1. Ep. 13. Cattle muster.

Series 1. Ep. 12. Arafura file snake.

Series 1. Ep. 11. Melbourne.

Series 1. Ep. 10. Penguins.

Series 1. Ep. 9. Gray Nurse shark.

Series 1. Ep. 8. Saltwater crocodiles.

Series 1. Ep. 7. Turtles.

Series 1. Ep. 6. Great white shark.

Series 1. Ep. 5. Honey ant.

Series 1. Ep. 4. Cassowary.

Series 1. Ep. 3. Tiger snake.

Series 1. Ep. 2. Funnel web spider.

Series 1. Ep. 1. Tassie devil.

VIEWER ADVICE
DO NOT DISTURB
OR TRY TO HANDLE
WILD ANIMALS AS THEY
MAY BE DANGEROUS

University of South Australia

- We've had streaming media available for a while now, and at this stage we have over 2100 videos on our streaming server. When Alma came in, we decided to further integrate this service. It was previously disconnected from the bibliographic record, and there was no correlation between the videos and the relevant courses.
- Once alma came along, we migrated each video link across to portfolios which are attached to the relevant bib record, which allows us to not only present them within discovery, but we can now also associate them with course reading lists, and present them within the other integrated systems.

Integrations: Discovery

- **Media Repository and Metrics**

```

    graph TD
      Primo[Primo API] --- Catalogue((Catalogue))
      Media[Media Server] --- Catalogue
      Alma[Alma Uresolver] --- Catalogue
      Altmetrics[Altmetrics] --- Catalogue
  
```

5-Hydroxyvitamin D, APOE ε4 genotype and cognitive function: Findings from the 1958 British birth cohort: Save Print Cite

Publication Title: European Journal of Clinical Nutrition

Online: Full text available via UnSA Research Outputs Repository @ Publisher
Full text available via UnSA Research Outputs Repository - Restricted Published Outputs

DOI: 10.1038/ejcn.2014.201

Authors / Contributors: Maccioni, J (University College London); Casadino, A (University College London); Power, C (University College London); Huoynh, E (School of Population Health)

Published in: European Journal of Clinical Nutrition v.68, no. 4, pp. 505-508

Publication Details: UK: Nature Publishing Group, 2015

Description: Both high and low vitamin D statuses have been associated with poorer memory function. Apolipoprotein E (APOE) ε4 alleles have been associated with reduced memory function, and separately with higher vitamin D concentrations... read more

Rights: Copyright 2015 Macmillan Publishers Limited

Content Type: Journal article

Funding Body / Grant No: UK Medical Research Council (09001953), Medical Research Council (09000934)

Physical Description: 4 pages

Bibliographic Identifier: 971500201101021

Usage Metrics:

Followed up by 8 news outlets
 Tweeted by 8
 On 8 Facebook pages

University of South Australia

- We have just recently migrated away from our DigiTool instance, and put all the metadata into Alma. However, we don't have/or want to move all our digital objects into the cloud, so we needed another solution. What we ended up doing was migrating all our objects off digiTool and onto simple file system storage, utilising the mms and portfolios for identification, and integrating all authentication within our discovery interface. We list off all portfolios, and link off directly to the digital object.

In this example, we also have an Altmetrics integration, we simply supply the DOI to the Altmetric API, and receive the metrics graphic, which links off for further analysis if needed.

Integrations: Back Office

- **Media Repository File Uploads**
 - Research Repository
 - Course Readings
 - Archival Content
 - Inter-Library Requests (Resource sharing)
 - *Audio / Video

- We've given our staff the ability to upload files directly to our media server, associating it with the mms / portfolio, making the digital objects accessible through discovery. This has recently been extended to include resource sharing requests, associating the digital objects to a users account / requests, making them available for download in their "myLibrary" requests page.
- Our staff either enter an MMS to lookup the portfolios for that record to assign files to, or enter a username of a patron with active resource sharing requests, which will list off the requests, allowing file uploads against each request.

Integrations: Discovery

- Library Account

The diagram illustrates the integration of the myLibrary system with Alma API and Library API. myLibrary is shown as a central hub connected to both Alma API and Library API, which are also connected to each other.

University of South Australia

Current Loans

Item Title	Due Date
Developing Semantic Web services / H. Peter Alessi, Craig F. Smith	19 Oct 2015, 06:00 PM

Outstanding Requests

Item Title	Status	Queue Position	Pickup at	Request Expiry
Web services - principles and technology / Michael P. Papadogiorgis	Pending	N/A	STADIAMRILL	01 Jan 0001

Inter-Library Requests

Item Title	Status
Development of evidence-based Australian medication-related indicators of potentially preventable hospitalisations: a modified RAND appropriateness method	Download Available
An evaluation study of sensory motor therapy	Created

Outstanding Fees

The following fees require payment.

Item Title	Fee	Paid	Balance
test	\$2	\$0	\$2
XSLT / Doug Tidwell	\$17	\$0	\$17

Course Readings


- CHE3009 (SP2 - 2015)
 - Reading: Practical non engineering - notes, even
- MATH10009 (SP2 - 2015)
 - Reading 01: Functions, graphs and models - Edwards, C. H. (Charles Henry)
 - Reading 02: Vectors and matrices - Edwards, C. H. (Charles Henry)
 - Reading 03: Algebra, Geometry, Trigonometry formulae, Table of Integrals, Table of Contents - Edwards, C. H.

- We of course integrated the patron account info into our discovery Webvity interfaces too... using the same data layer as all the other systems use.
- We just added the course readings lists too, giving students an overall view of all their course readings in the one place.


Working with the APIs

Some risks:

- Availability of relevant data
- Inconsistency of data
- Change of delivery (SOAP / REST)
- Downtime / Loss of connection
- **Speed**



The diagram shows three interlocking gears. The top gear is labeled 'Voyager Web Services', the middle-left gear is 'Summon API', and the bottom-right gear is 'Alma / Primo APIs'. Arrows indicate a clockwise flow of interaction between the gears.

 University of South Australia

We were so used to having direct, unrestricted access to our data when using Voyager. So, when it came time to move into the cloud and alma, we were facing some challenges to keep the same level of service to all our systems.

The Alma APIs have improved dramatically since we first began, initially there limitations on what they provided, and the speed was a concern too. Since we first began, we have seen improvements in both data availability and speed. There's definitely room for improvement, and we are always happy to work closer with ExLibris to make them as good as they can.

Now, ExLibris aren't the only source of data we have, and as I mentioned previously, all our data communications basically go via our APIs, which we can refer to as our data layer.

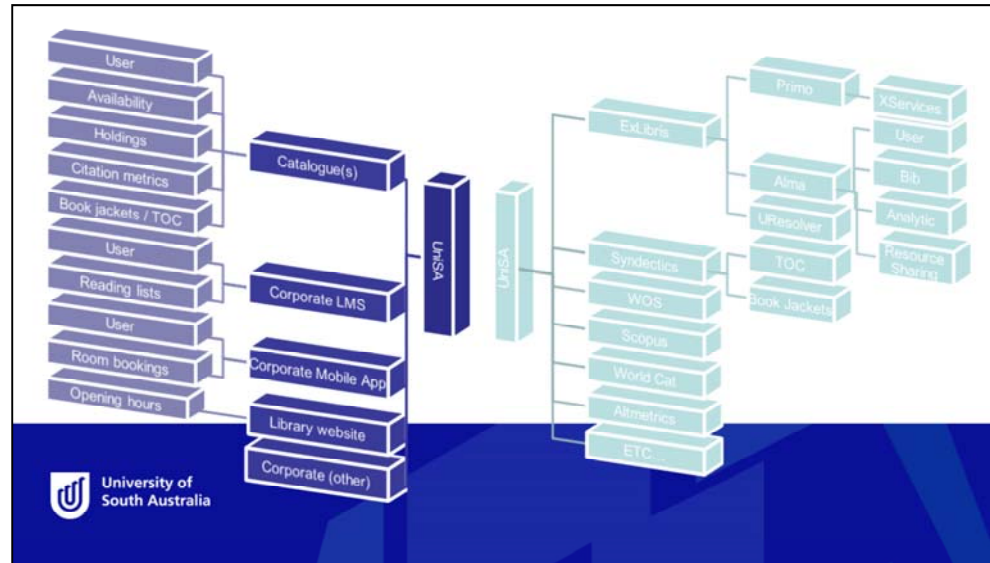
The “Data Layer”

- **Vendor / Data custodian:**
 - e.g. ExLibris, Syndetics, Serials Solutions, In-House Data stores, Altmetrics, WoS, Scopus
- **Data Layer:**
 - **Common, simplified** communication interface to end user systems
- **End user / User Interfaces:**
 - The clients (via discovery)
 - Integrated systems (via APIs)

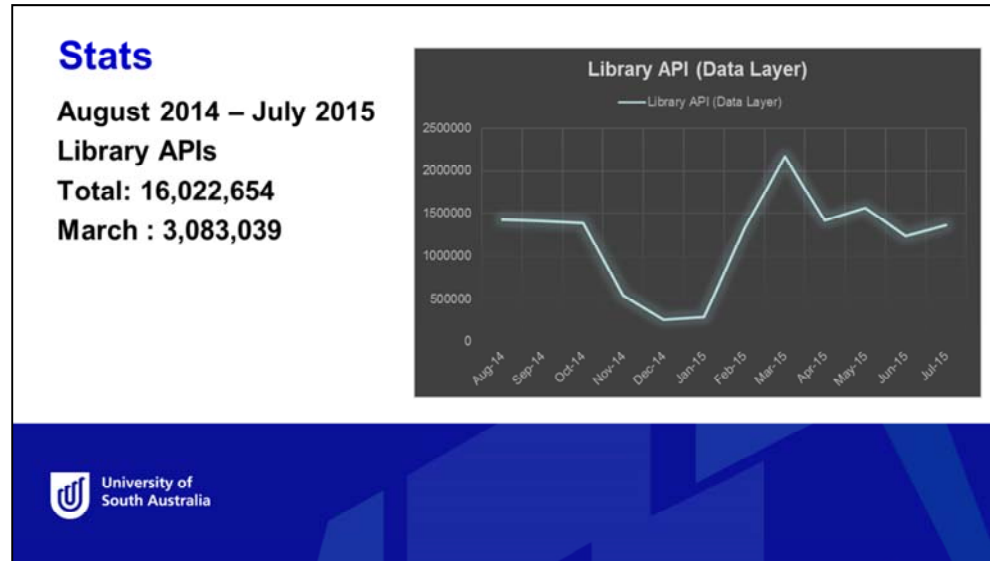
The diagram illustrates the data layer architecture. It features three main components: 'Vendors' (top), 'User Interfaces (discovery, portals, LMS, etc.)' (middle left), and 'Data Layer (our APIs)' (middle right). Arrows indicate the flow of data: from Vendors to the Data Layer, from the Data Layer to User Interfaces, and from User Interfaces to the Data Layer. The bottom of the slide features the University of South Australia logo and name.

Now, none of these interfaces exist without the “data layer” the interpreter between (in this case) the vendors and the interfaces the users will interact with.

This next slide is an overview of some of data sourced via our APIs and from our APIs to other systems.

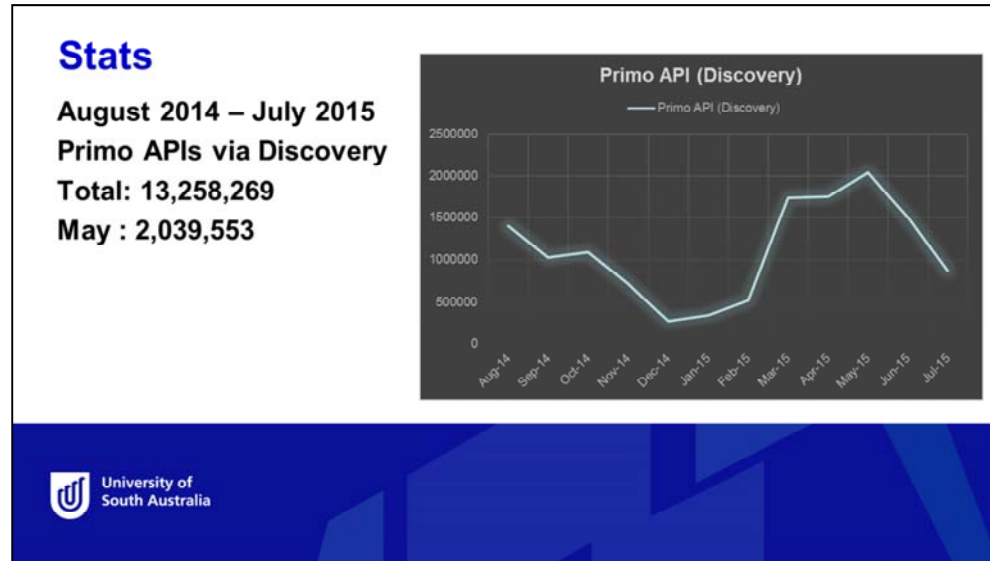


On the left (lighter blue) is the various 3rd party data sources we use to provide our services and integrations. On the other side is just some of our APIs, and the various systems that interface with them.

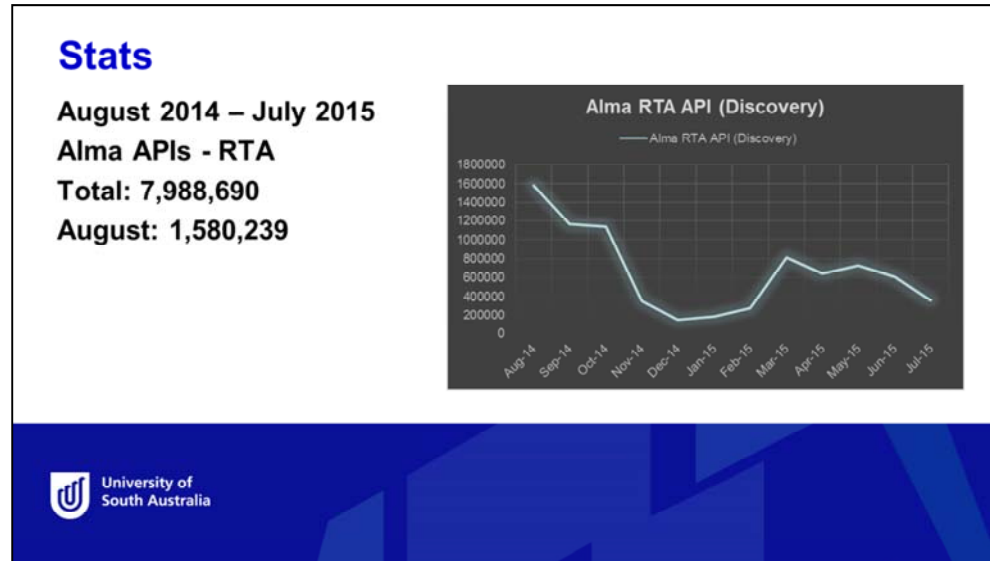


I'll just throw a few stats out now, relatively small in the big picture of APIs (I'd like to get a look at the ExLibris numbers)

But this surprised me a little, our APIs total hits for the year, just over 16 million with our start of semester in March being our peak, at just over 3 million.



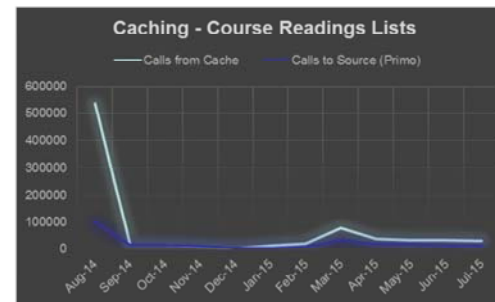
Our Discovery interface relies almost entirely on the Primo API, but yet again, I was surprised, with a total of over 13 million, and May being our peak with a little over 2 million there. I'm guessing assignment crunch time in may, and exam prepping the cause for that peak.



Now we have some Alma API stats, here's just our Real Time Availability for the period. As you can see, it dips a bit from the beginning. There's a good reason for that. We began caching some calls locally to improve speed and reduce the overall bandwidth. With caching we more than halved the calls from one month to the next.

Stats

August 2014 – July 2015
Course Readings (Cache)
Total From Cache: 829,683
Actual to Primo: 261,886



This is an example of where we cache our custom API calls to save on speed and bandwidth. We can see a massive peak in August 2014, which was due to our colleagues in the learning management systems team deciding to do some load tests, forgetting to switch off our production API connection... so we saw an extra few hundred thousand there, luckily we were caching, otherwise, primo would have got a big hit overnight, and probably caused us some usage limit issues...

Questions?

Thank You



University of
South Australia

Ben.Dalwood@unisa.edu.au