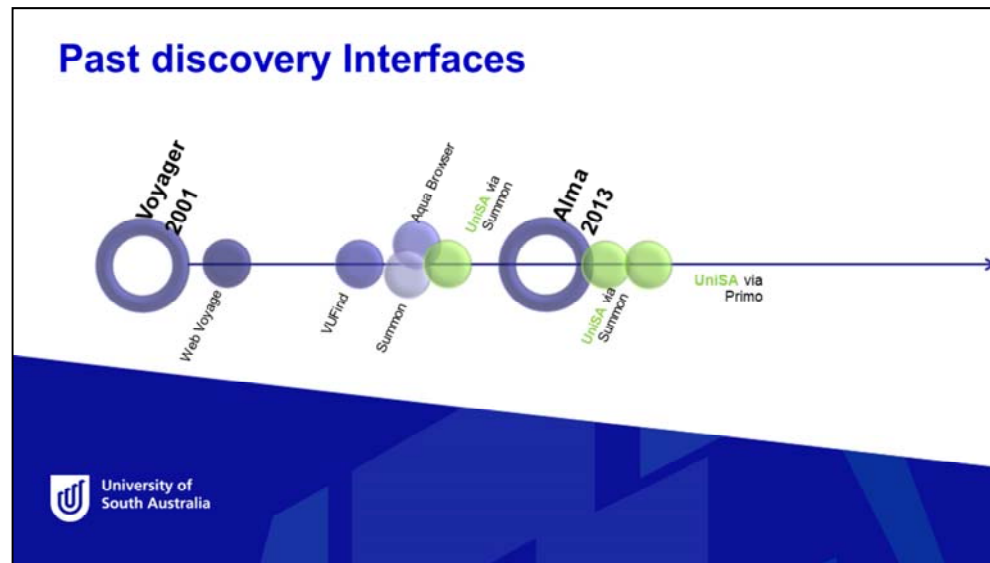




Hello, thanks for joining me!

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



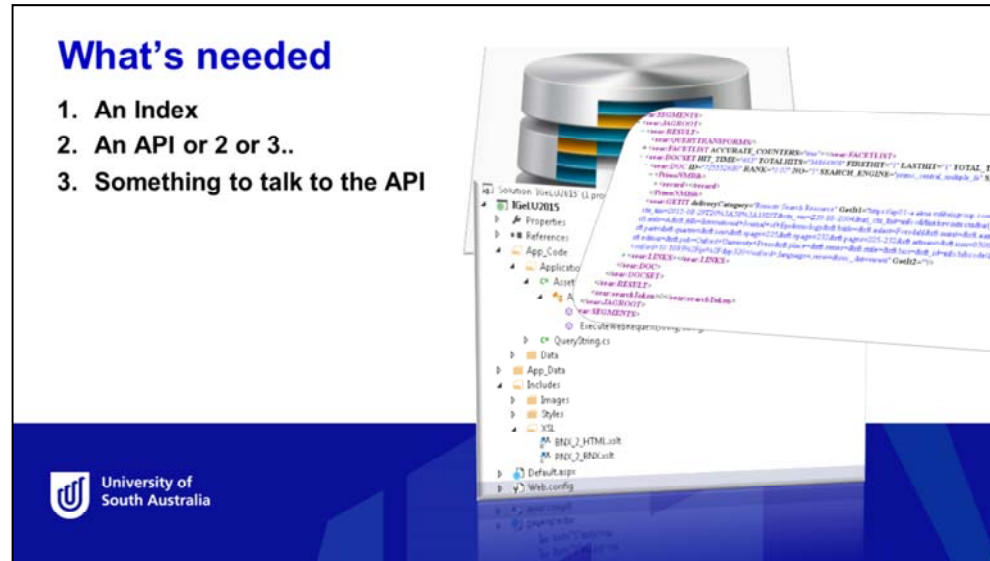
Here's a bit of history of our discovery interfaces... We've gone through a few interfaces over the past 12 years. When I began at unisa we had only just started on voyager, and that meant web voyage. [pop-up web voyage]

After many years, and several customisation short-comings, we took the plunge into trying out something different. This is where VUFind came in [pop up pic] much more customisable, but was lacking the more powerful indexing abilities, and as such, our management decided to give summon a go... our first hosted solution, this being somewhat a challenge when it came to patron services and customisations. It's interface had a few more bells and whistles, but managing the metadata and availability aspects was cumbersome at best. We then trialled Aqua Browser, and had a brief stint in that domain, which was once again a challenge, having a hosted solution where the metadata and indexing was outside of our domain, it remained a ongoing challenge.. We'd still had our subscription to summon, and had been looking into their APIs during this phase, and we came up with an idea... create our own interface, using the APIs of Summon for discovery, and the APIs of Voyager for patron services... a plan was born... we focus-grouped students and staff to detail all services and interfaces that were used by the end user. Then produced a prototype. [screen shot]. We had done all this knowing we'd be moving to Alma/Primo, in fact, it was one of the incentives to get ahead of it and create an interface that would be vendor agnostic, one that would allow us to (relatively)

simply “plug in” primo APIs and continue with a seamless change of vendor without users needing to learn yet another interface...

What's needed

1. An Index
2. An API or 2 or 3..
3. Something to talk to the API



- So, basically, there only 3 things we need to get to a point where our index can be turned into a discovery interface. Providing we have these, we can have something up and running in no time.
- I was going to bring my laptop and do this interactively, but lugging a laptop all over europe for 6 weeks didn't sound fun... So unfortunately, I've only really got screen shots. I'll try and access the basic interface later on, to see what I can access.

The Index

- Availability of the metadata
- How customisable is it
- Searchable fields
- Filtering capabilities
- Speed

```
</SEGMENTS>
<sear:JAGROOT>
<sear:RESULT>
<sear:QUERYTRANSFORMS>
+ <sear:FACETLIST ACCURATE_COUNTERS="true"></sear:FACETLIST>
- <sear:DOCSET HIT_TIME="463" TOTALHITS="34864908" FIRSHIT="1"
- <sear:DOC ID="725552680" RANK="0.07" NO="1" SEARCH_ENGINE="p
+ <sear:NMBB>
+ <sear:NMBB>
</sear:NMBB>
<sear:GETIT deliveryCategory="Remote Search Resource" GetIt="http://ap
ctc_tm=2015-09-29T20%3A58%3A19IST&ctc_ver=239.88-2004&url_ctc_fmat
rft_atitle=&rft_id=International+Journal+of+Epidemiology&rft_btitle=&rft_aulat=F
rft_part=&rft_quarter=&rft_ssn=&rft_spage=225&rft_epage=232&rft_page=225-232&
rft_edose=&rft_pub=Oxford+University+Press&rft_place=&rft_serie=&rft_state=&rft
</oafid=10.1093%2Fq%2F-dp.320</oafid>_language=&sew=&svc_dat=vieww" G
+ <sear:LINKS></sear:LINKS>
</sear:DOC>
</sear:DOCSET>
</sear:RESULT>
</sear:searchToken></sear:searchToken>
</sear:JAGROOT>
</SEGMENTS>
```



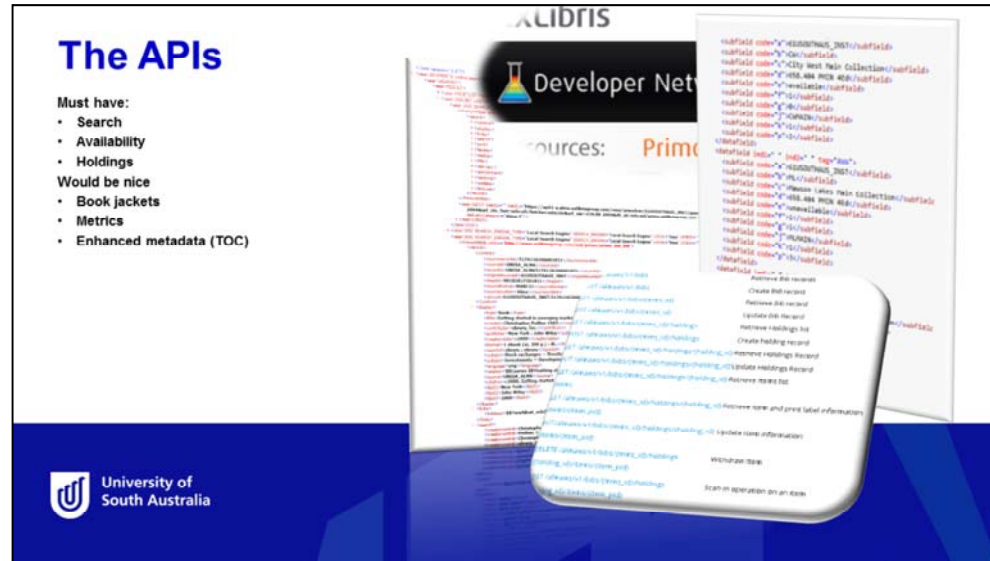
University of
South Australia

Obviously, without the index, we have nothing...

When we began the adventure into doing our own interface from scratch, we were using Serials Solutions, Summon as our indexing vendor. We had previously been using a local index using VUFind and Solr, but our management decided to go with Summon at the time, and as such we had it's more powerful index to leverage, and we knew we'd eventually be in a fully hosted Primo/Alma environment, so we would soon have Primo's Index to get our hands on...

These are 5 things we needed to consider when beginning.

- Do we have access to enough of the metadata
- Can we customise the index, fields, mappings, relevancy, boosting, etc.
- Can we create additional searchable fields
- Does it support filtering / facets / customising
- And finally, How fast can we get all that back?! S



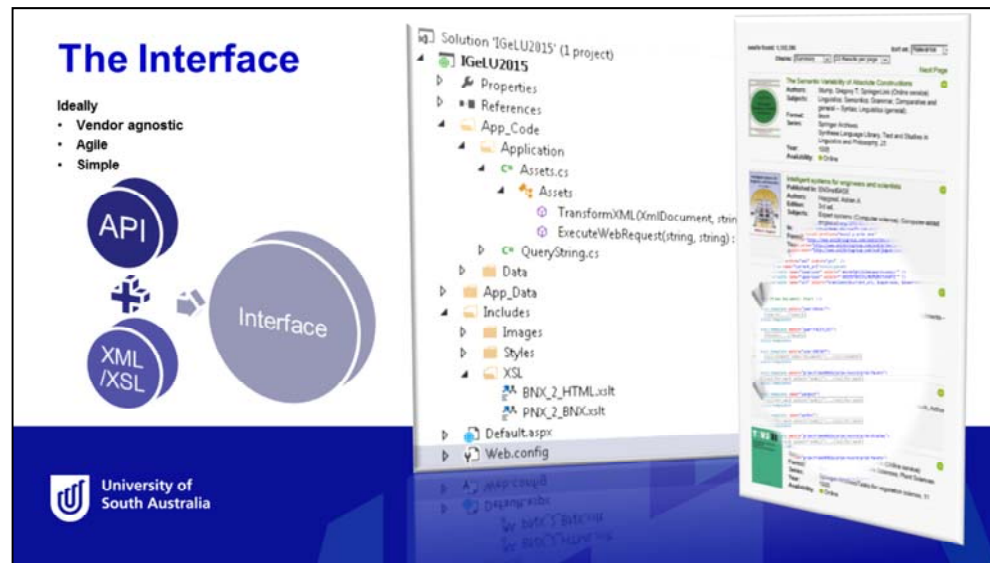
The only way we could achieve any kind of interface, especially when our index is externally hosted, is with APIs.

As I mentioned previously, we'd began this journey with Summon, and the Summon search API. There were a few challenges, a few issues, a few annoyances, but it wasn't anything we couldn't handle with a bit of persistence.

- I was pleasantly surprised when I got my hands on the Primo search API. To begin with, it was so much more strait forward to deal with. We were missing some functionality to begin with, but once we had some back office access, we could add in some of the missing bits we needed for our interface to cooperate.
- As far as discovery goes, we need to obviously a search API, [check, primo], some real time availability of physical items was also necessary for us [check, alma] though it needed some work when we first started,

but it's been much more reliable since we first began. Holdings data was a bit issue for us for a while, we wanted to be able to show our holdings for physical items, so a request could be actioned at the item level. [volume] this wasn't readily available when we began, so we had to use the [alma analytics] API, [slow and 24 hours old] to get our holdings lists [no RTA, not always available] we were happy once the [alma bib] APIs were enhanced, and we could finally get some live holding out of alma. [although, it can take anywhere from 4 to 10 API calls to achieve a full holdings list]

- The not so necessary, but nice ones were all dependant on other vendors, which we have, so we integrated syndetics and altmetrics for some additional enhancements.



Now, none of these are of any use without a way to talk to the API so it can talk back to us...

As far as our first iteration into this space, where summon was our indexer, we initially built the interface to a summon centric spec. Which wasn't a great idea. A lot of the functions were built to deal with summon specifically, rather than abstracting the presentation from the data. But at that stage, it was more about getting something out, rather than best practices. However, we knew we'd be tweaking endlessly to get it how we want, and we had our Primo/Alma install 12 months away, so we had time to release plenty of updates, which allowed us to abstract the data from presentation.

I'll try and show the 3 simple steps which gets us from api to interface. [
<http://search.library.unisa.edu.au/anzreg2016/>]

Using Primo API

Back office flexibility

- Custom search scopes (7)
- Custom facets (16)
- Custom metadata fields (51)

Interfaces

- Host based config / multi hosted
- We use 5 different addresses (URLs) to dictate different interfaces

The image shows two overlapping screenshots. The top one is a search result for '5-Hydroxyvitamin D, APOE ε4 genotype and cognitive function: findings from the 1958 British birth cohort'. It shows the publication title, journal name (European Journal of Clinical Nutrition), and authors (Mason, J., Casadino, A., Power, C.). The bottom screenshot shows the 'myLibrary' interface with a 'Course Readings' section listing various articles and books.

University of South Australia

Primo has given us the ability to use the search scopes to create sub interfaces which target a specific collection, such as our digitool replacement, research outputs repository and our alumni specific interface. Though, the scoping and the index are far from perfect, with a few workarounds, we can get the desired results.

We've also utilised the various customisable PNX local fields, this has given us the ability to tailor certain records to our needs, such as open access, copyright, streaming videos, citations, PIDs from our handle server, the list goes on.

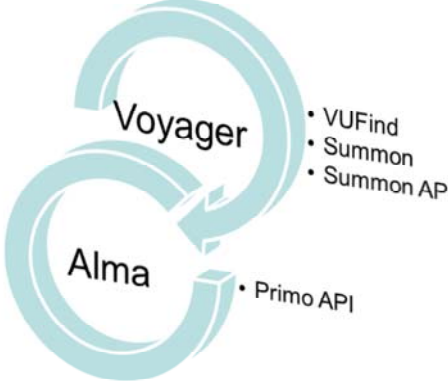
Another nice feature we use quite frequently, are the custom search fields. This gives us the ability to customise such searches as our course readings lists, which we not only provide our LMS with an API into that, so we can provide students with their readings in our moodle environment, but we can use that same logic to customise our myLibrary account pages to list off all the students courses, which we've only just released last week..

This flexibility gives us the freedom to work with our data, the way we need, to get it to our clients, and having this interface of our own, opens up this to endless possibilities..

[requests at item level [holdings], media streaming, document repository, SSO to ezproxy, SSO from moodle, some basic.]

Migration: Discovery

- **Identify the capabilities (APIs)**
 - Availability of existing APIs
 - Losing direct access to data (voyager)
 - Limitations of exiting APIs
 - Availability of data
 - Adapt to Missing Functionality
 - Future APIs
 - Performance of APIs
 - On-site vs Cloud



University of South Australia

Now all of this doesn't come without it's risks... we had to work through the lack of APIs and data in the first instance, which was quite a challenge, we had to adapt a bit, but this was all part of the deal when moving into the fully hosted cloud environment. And as it stands now, we've got far more flexibility with our interface than we'd ever have with one of the hosted solutions. We can be far more agile with any changes or enhancements that come along. (so long as the functions and features are available of course..) our biggest issues to date have really been performance.

Questions?

Thank You



University of
South Australia

Ben.Dalwood@unisa.edu.au