Lightweight applications by librarians for librarians

Asbjørn Risan, Unit, BIBSYS
Developers Day, IGELU 2018, 23rd of August 2018

Agenda

- About me and Unit
- Why librarians should code
- Things to be aware of!
- The BIBSYS environment
- Live! examples

Goal of the day: Inspire to explore the possibilities

Disclaimer

- I will show (almost) no code in this presentation, but I’m more than happy to share and/or discuss code afterwards

- Contact information are included in this presentation

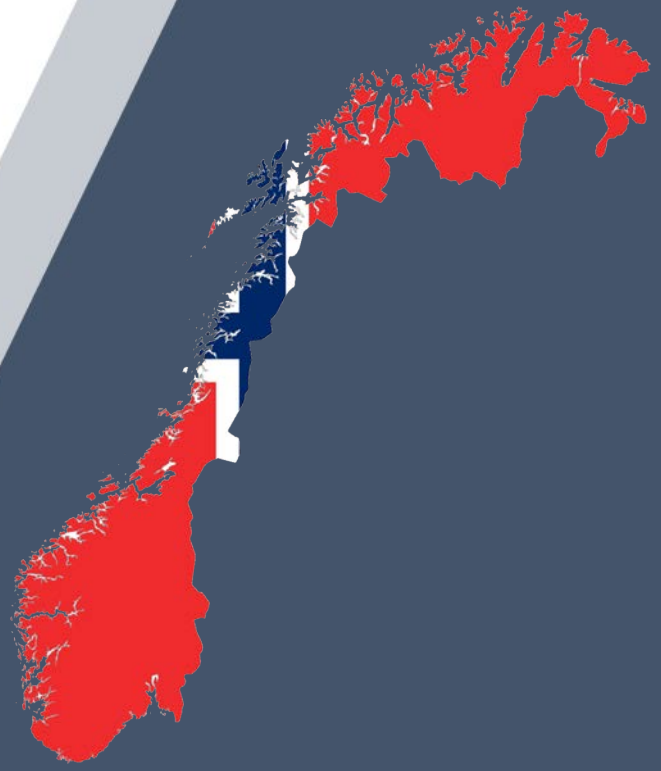
About me

- Librarian with a masters degree from OsloMet - Oslo Metropolitan University
- Started in BIBSYS* in 2008
- Product owner of Primo, Leganto and BIBSYS Authority files
- No formal degree in programming, except what was part of the library education
- Like to code on the side

*Now Unit

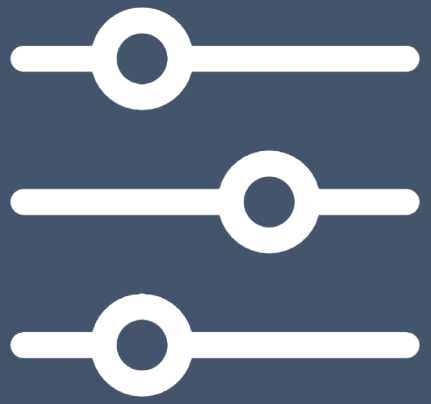
About Unit

- Unit – The Norwegian Directorate for ICT and Joint Services in Higher Education and Research
- Merger between three governmental institutions (BIBSYS, CERES and parts of UNINETT) as of January 1st 2018
- Established as a directorate owned by the Ministry of Education and Research
- Unit offers a wide range of services within the areas of education and research management for the higher education sector and academic/research libraries



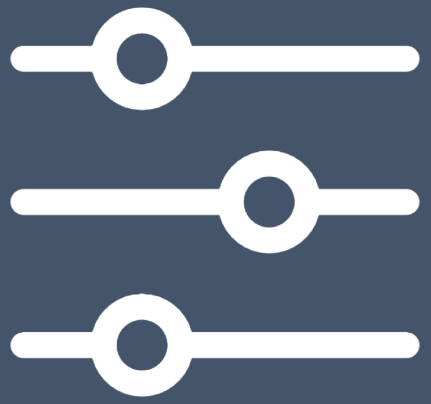
Why should librarians code? (or why do I code!)

- Educational – learn something new
- Challenging – develop skills in problem solving
- Be creative – find solutions
- Save time – automate instead of repeating tasks
- Efficient – I know my needs
- Knowing the basics makes it easier to talk to developers and understand challenges and possibilities
- Instant reward and feedback on screen
- It’s fun



To be aware of!

- Security – are the service secure, can it be misused, do it affect data?
  - Peer-review of the code
- Privacy – are there any personal data involved, can anyone access/change personal data?
  - Peer-review of the code
- Maintenance – who will do the maintenance? If you develop something you are probably stuck with it!
  - Stuff do break!
- Bugs – have you covered all of the corner-cases?
  - Testing, testing, testing (preferably by someone that's not you)
- Changes – as soon as you launch something you will get suggestions for changes!



The BIBSYS consortia

- ≈ 80 institutions
- Products:
  - Alma
  - Oria (Primo)*
  - Leganto (reading list consortia)
- ≈ 12 persons handling configuration, set up, administration and support for Ex Libris products to the consortia
- A need for efficiency and structure

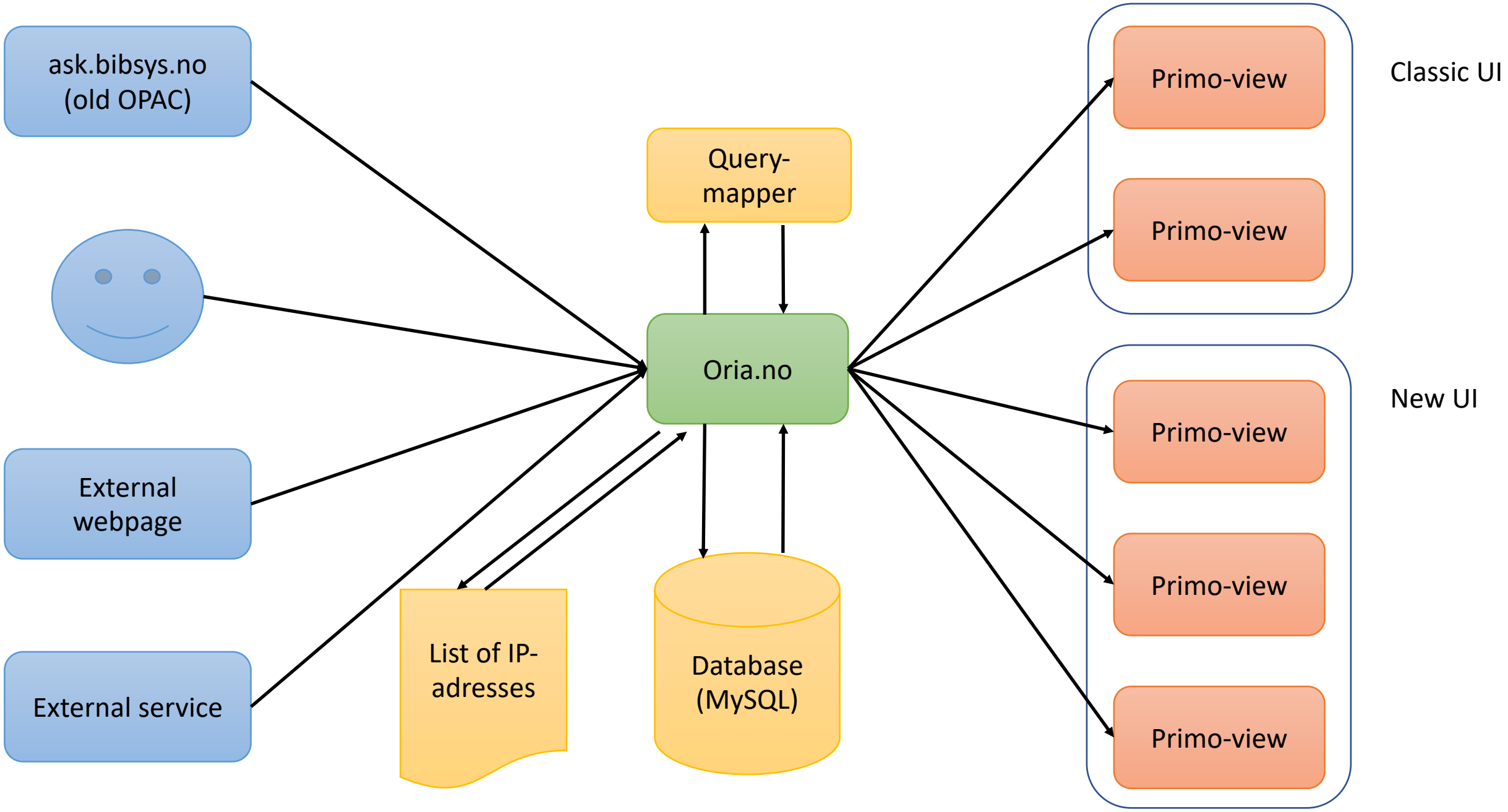
*Oria is the Norwegian flavour of Primo

The BIBSYS environment – key elements

- Database (MySQL) with basic information about each institution
  - Display name
  - Primo code
  - Alma code
  - Base-urls
  - Live with new UI-indicator
  - …

- Re-direct-functionality behind oria.no*
  - Redirects users based on IP
  - Redirects based on parameters (view-code)
  - Re-directs from old OPAC to Primo
  - Basic query-mapping

*Oria is the Norwegian flavour of Primo


ask.bibsys.no
(old OPAC)
External
webpage
External service
Query-
mapper
Oria.no
List of IP-
adresses
Database
(MySQL)
Primo-view
Primo-view
Primo-view
Primo-view
Primo-view
Classic UI
New UI

Examples

Demonstrate some examples of services created

- Search box generator
- New items report
- Permalinks for courses and reading lists
- uResolver feedback

Example 1: Search box generator

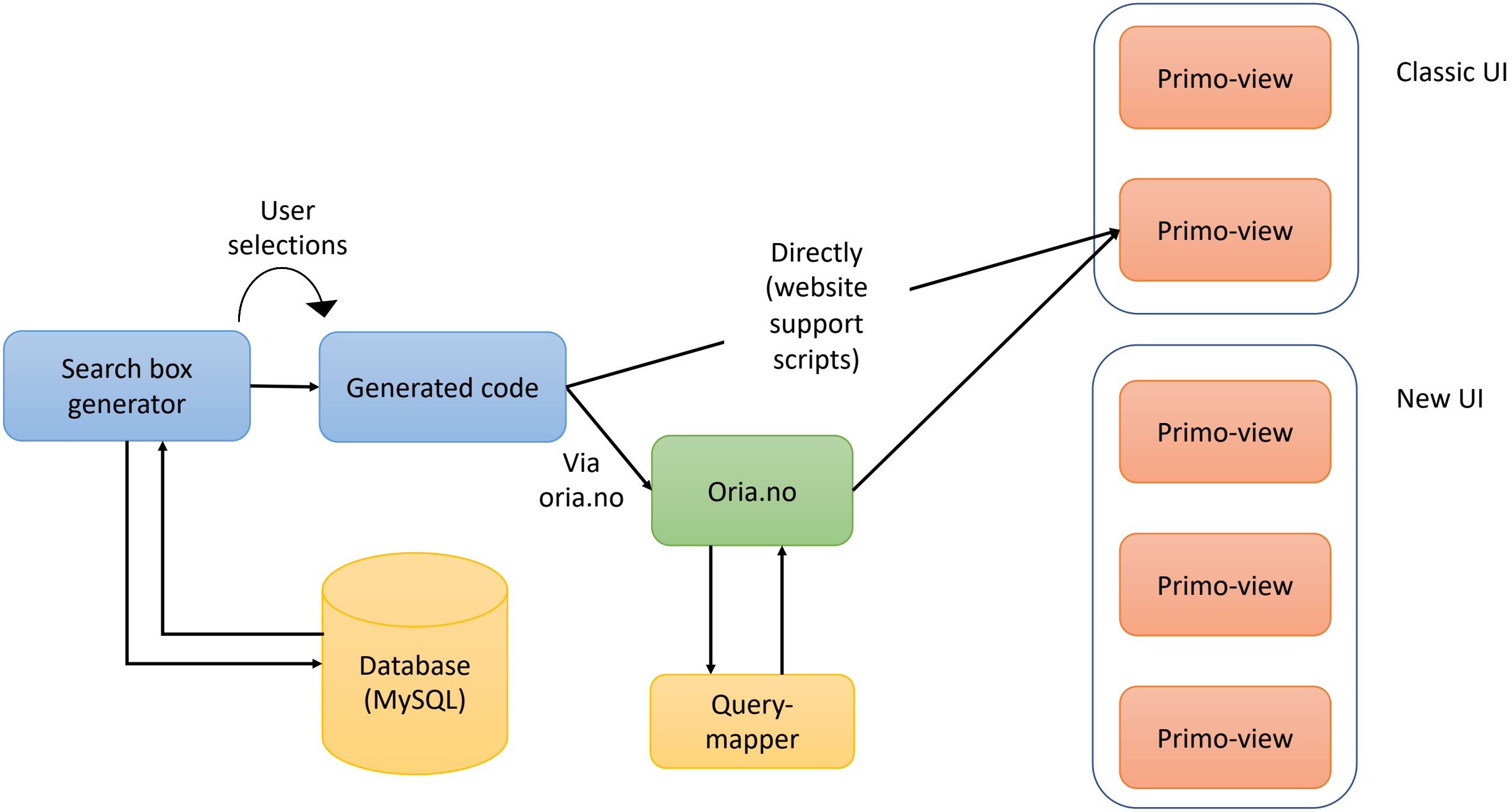
Usecase: 80 institutions needs code for a search box for Primo that they can include on their own website

How to ensure that they get the correct code without the need to contact BIBSYS Support?

Example 1: Search box generator

Solution: We buildt a search box generator that generates the code needed based on a few selections by the user

Uses the database to display list of institutions and to fetch view code etc


User selections
Search box generator
Generated code
Directly (website support scripts)
Via oria.no
Oria.no
Database (MySQL)
Query-mapper
Primo-view
Primo-view
Classic UI
Primo-view
Primo-view
Primo-view
New UI

Example 1: Search box generator

Solution: We buildt a search box generator that generates the code needed based on a few selections by the user

Uses the database to display list of institutions and to fetch view code etc

https://www.oria.no/sokeboks/sokeboks_nui/sokeboks.php

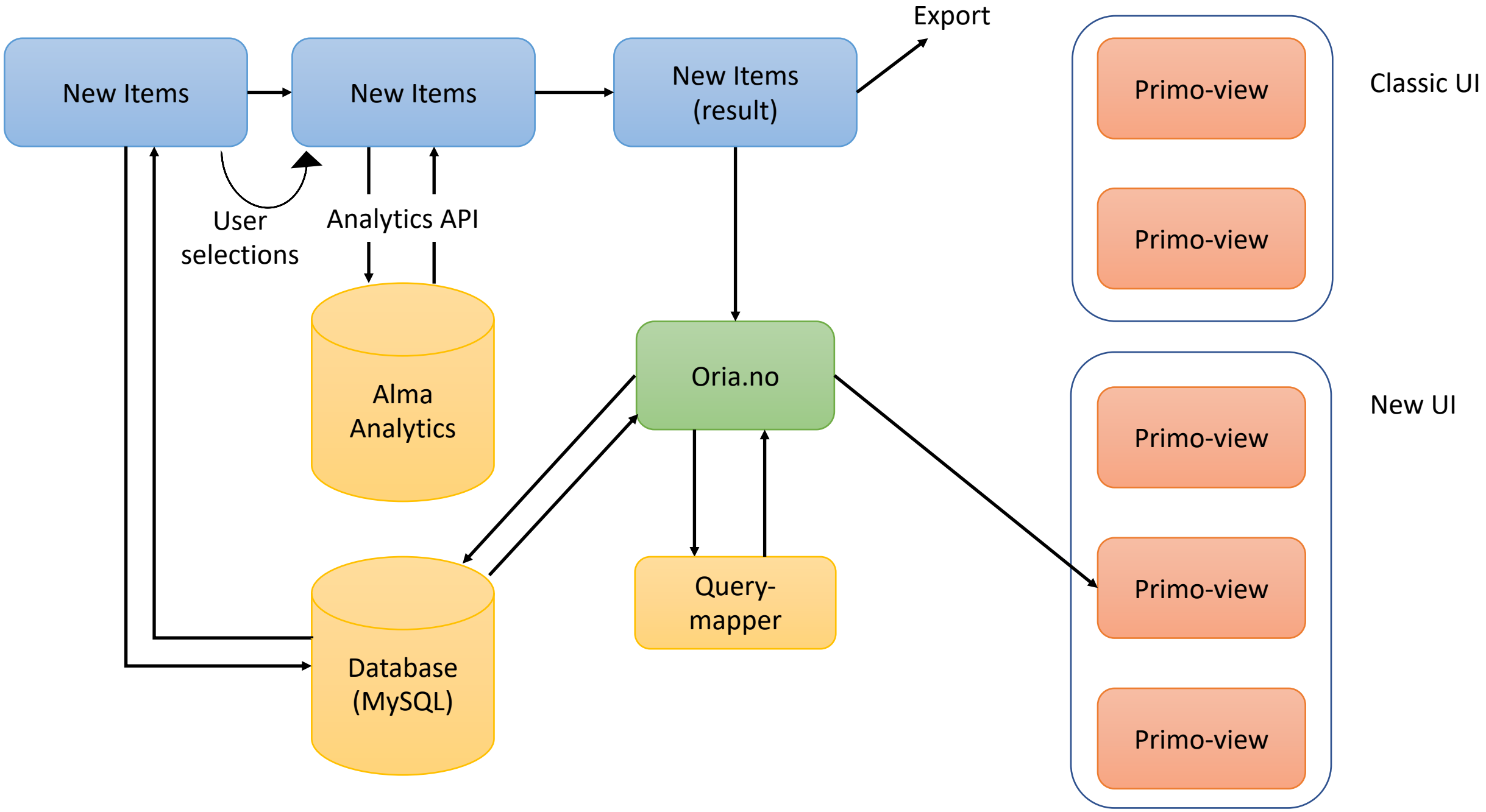
Example 2: New items list

Usecase: The institutions wants a sorted list of new documents for a given period with a link to the OPAC

Example 2: New items list

Solution: A service that uses the analytics report-api filtered by user input for institution and time period

Format to HTML and include link to Oria with the possibillity to download the file


New Items
New Items
New Items (result)
Export
User selections
Analytics API
Alma Analytics
Oria.no
Query-mapper
Database (MySQL)
Primo-view
Primo-view
Classic UI
Primo-view
Primo-view
Primo-view
New UI

Example 2: New items list

Solution: A service that uses analytics report filtered by user input for institution and time period

Format to HTML and include link to Oria with the possibillity to download the file

https://www.oria.no/nyhetslister/index.php

Example 2: New items list

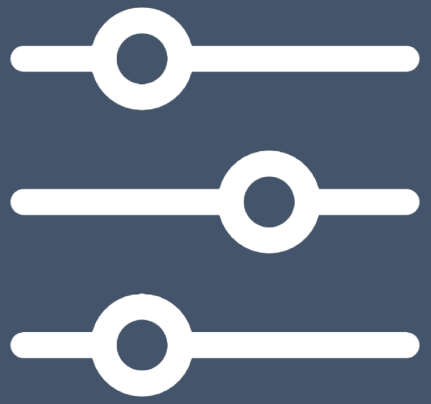
Experiences:

- May be slow especially for longer time spans – need to add a lot of retries due to time out
  - Should use API-key for each institution instead of Network-API
- Response in the API not to be trusted – stated finished without delivery of data

Example 3: Permalinks for courses and reading lists

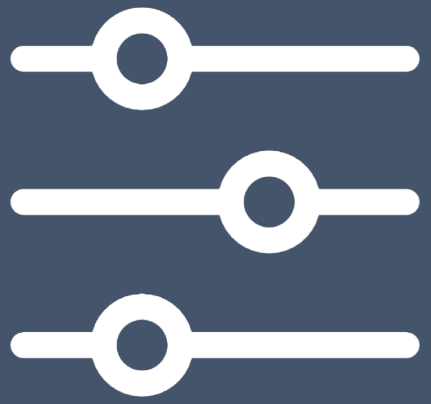
Usecase: There was a need to get permalinks for all (or selected) courses and permalinks to a specific reading-list. Analytics did not provide the necessary ID’s

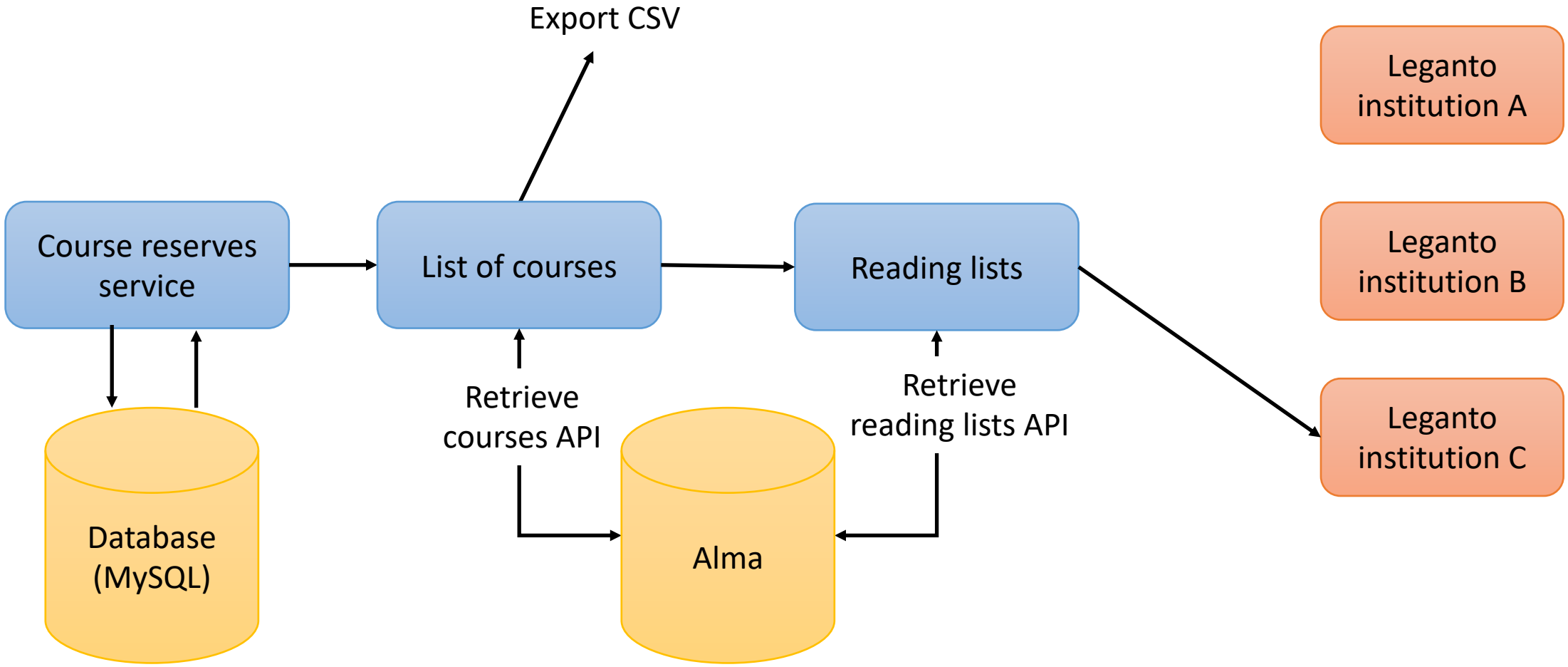
Bonus: Showcase the possibilities with the Leganto-API’s for the reading list consortia



Example 3: Permalinks for courses and reading lists

Solution: Build a service that use the Leganto-API’s to provide a list of all courses with the possibillity to export to CSV or drill down to reading list level

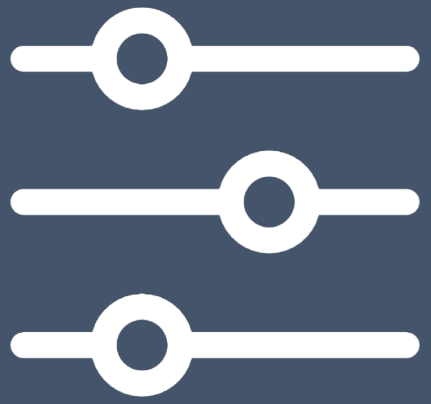



Export CSV
Leganto institution A
Course reserves service
List of courses
Reading lists
Leganto institution B
Retrieve courses API
Retrieve reading lists API
Leganto institution C
Database (MySQL)
Alma

Example 3: Permalinks for courses and reading lists

Solution: Build a service that use the Leganto-API's to provide a list of all courses with the possibillity to export to CSV or drill down to reading list level

https://pensum.bibsys.no/lister/index.php



Example 4: uResolver feedback

Usecase: There is a need for a feedback form in the link-resolver for the end user to report broken links or other feedback to the library

Example 4: uResolver feedback

Solution: Add a general electronic service in the uResolver that links to a webform that enable the end user to send a feedback to the library including all necessary metadata

Service Details

Feedback

Service Code feedback
Service Description Tilbakemeldingsskjema i lenketjeneren

Service Details | Service Availability Rules

Active: **Active**

Service Code *: **feedback**

Service Name *: **Feedback**

Service Description: **Tilbakemeldingsskjema i lenketjeneren**

Public Name: **Tilbakemelding/Feedback**

Public Note:

Display Location *: **Getit & viewit**

URL Template *: **http://www.oria.no/feedback?atitle={rft.atitle}&au={rft.au}&aufirst={rft.aufirst}&aulast={rft.aulast}&btitle={rft.btitle}&date={rft.date}&dcidentifier={rft.dcIdentifier}&doi={**

Enable without login: (●) Yes ( ) No

Disable service: (●) Never

( ) When resource is owned by the campus

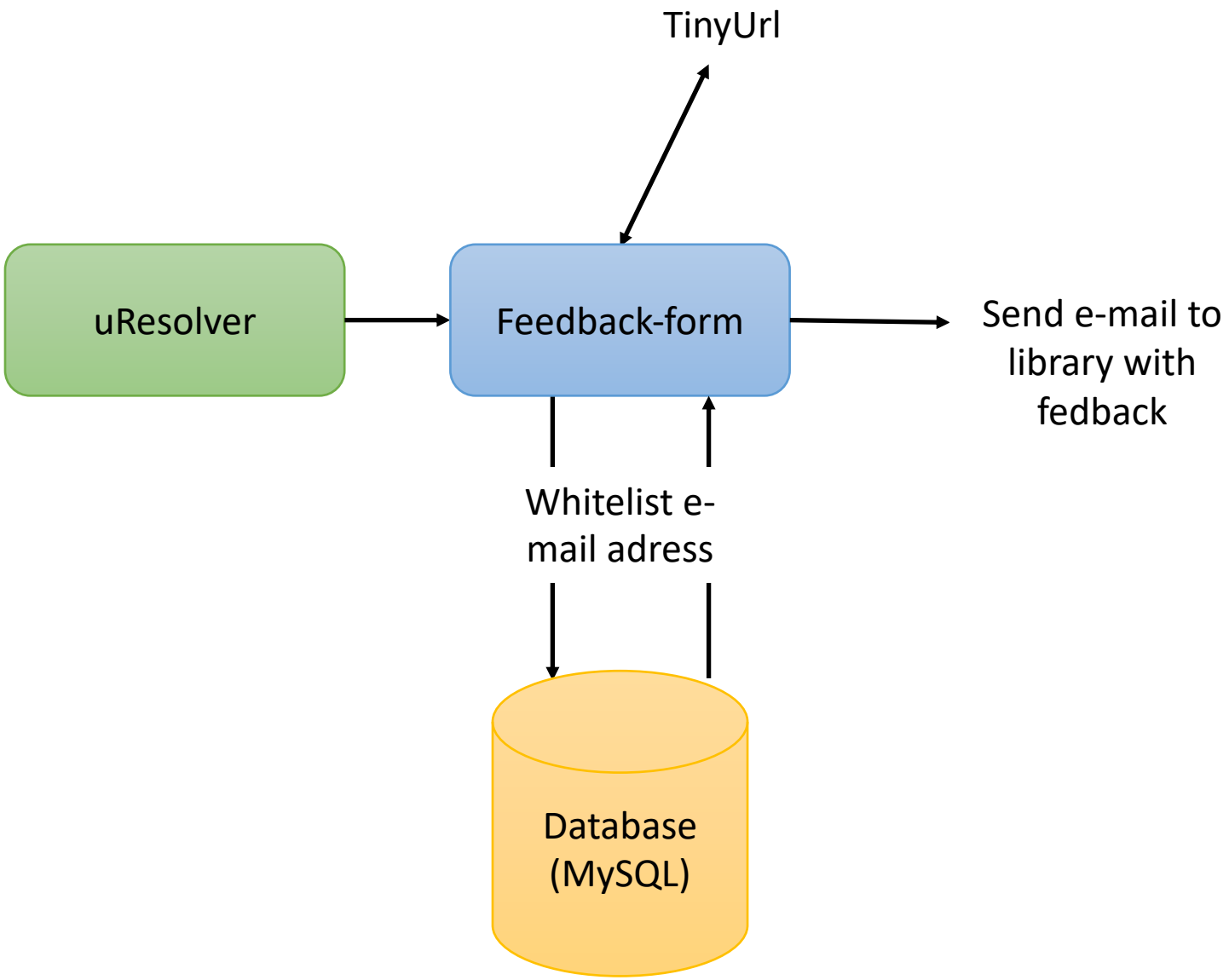
( ) When resource is owned by the campus and available

( ) When resource is owned by the institution

( ) When resource is owned by the institution and available

Example 4: uResolver feedback

http://www.oria.no/feedback?atitle={rft.atitle}
&au={rft.au}&aufirst={rft.aufirst}&aulast={rft.au
last}&btitle={rft.btitle}&date={rft.date}&dcident
ifier={rft.dcIdentifier}&doi={rft.doi}&issn={rft.iss
n}&eissn={rft.eissn}&isbn={rft.isbn}&eisbn={rft.
eisbn}&epage={rft.epage}&spage={rft.spage}&p
ages={rft.pages}&genre={rft.genre}&issue={rft.i
ssue}&jtitle={rft.jtitle}&pub={rft.publisher}&pub
lisher={rft.publisher}&title={rft.title}&volume={r
ft.volume}&source={rfr_id}
&nz_id={rft_dat}&epost=library@institution.no
&oria_inst=diakon


TinyUrl
uResolver
Feedback-form
Send e-mail to library with fedback
Whitelist e-mail adress
Database (MySQL)

Example 4: uResolver feedback

Solution: Add a general electronic service in the uResolver that links to a webform that enable the end user to send a feedback to the library including all necessary metadata

http://uib.oria.no

Thanks!

Questions?

Contact information:

- E-mail: asbjorn.risan@unit.no
- Phone: (+47) 95 88 41 31

Links

Search box:
https://www.oria.no/sokeboks/sokeboks_nui/sokeboks.php

New items report:
https://www.oria.no/nyhetslister/index.php

Courses/lists:

https://pensum.bibsys.no/lister/index.php

Feedback: http://uib.oria.no