

Script workshop “swissbib for the short distance runner”

slide 1 – introduction:

Hello – a short introduction of myself: My name is Günter Hipler. I'm working for the swissbib project since 5 years in the role as a system architect.

3 out of 5 years the service is running in productive mode answering between 20.000 to 50.000 user requests daily.

The last 8 month we worked on a further development of the whole platform. This may have been the reason why the productive service will have appeared to be unchanged to the 'external user'

What was done – in short?

- Within our Data Hub the main principles in matching and merging of duplicate records have been significantly changed
- We developed a new presentation component based on the latest VuFind software. In the last 2 years VuFind has been adopted by a lot of institutions and networks around the world. Remarkably often by institutions in Europe – especially Finland and German speaking countries. Having done this shift we think this is a good investment into the future of the service. Last but not least: Our new presentation component is Open Source. Freely available for everyone who wants to use it – either by it's own or as part of the swissbib infrastructure.

How easily this could be done is the main topic in part II of this workshop
“swissbib for the short distance runner”

slide 2 - schedule/outline “swissbib for the short distance runner”:

The whole workshop is divided into three parts:

1) During the first 30 minutes I'm going to introduce the general architecture of the solution

Here I want to address the topics:

- Which components is part of the solution

- What is the reason that I speak about Swissbib as a layered solution or service and not Swissbib as a product? Sometimes I use the expression “Swissbib as a (temporary) working bench for other people with their ideas. What's the sense behind these catchphrases?
- What are the advantages of such a layered architecture compared with other (even commercial) discovery solutions?
- Why are we, even after 5 years, (3 in daily production) still convinced that the architecture of the Swissbib platform is a good choice for the Swiss library community landscape and offers a great potential for the future?

I think this architectural overview is necessary to provide a solid understanding for the hands on session of the second part of this workshop.

2) In the second part of this workshop I will introduce you to the process of building a presentation component by your own

I will do this on my own laptop – which should give you the possibility to follow the hands on. I think there won't be enough time to retrace it immediately – this could be done later, when you have more time. Anyway – if you want to retrace immediately, I made some hardcopies of the detailed instructions which might be helpful in doing the work by yourself.

3) The third part of the workshop is reserved for an open discussion

Slide 3: swissbib architecture – A layered system with open interfaces

This picture is a rough overview of the infrastructure and components part of swissbib.

The main parts of this picture are - and please try to keep these in mind

- the swissbib solution as a whole is symbolized by the green area
- the yellow circles (like Easter eggs) are the symbols for interfaces used by the components for communication.
- the three yellow sticks divide the solution in dedicated parts (often called layers).
- the components part of each layer are symbolized by the grey rectangles.
- content flows from the left to the right. **We recognize the functionality (or components)** within each

layer

- a) First: a contentCollection component
- b) secondly: a data hub
- c) thirdly: the Search Server layer
- d) and last but not least – fourthly: our presentation component mostly used by our customers.

- **the interfaces** (the yellow Easter eggs) **are used internally** by Swissbib components – but, and this is important – they are **also used by humans and services outside of the system** because they are open for everyone.

- as you can see: the **components** we enumerated before are **Open Source as well as commercial**. They can easily **talk to each** other because both **use the open interfaces** between them.

- **the functionality within each layer is closed in itself**. It does what it should do – no more but no less. It get what is necessary by using an interface and the result of the functionality is provided again by interfaces to other components (internally or externally).

This concept makes the solution extremely flexible even for the future.

If only one component of the whole solution no longer meets the requirements it could be exchanged by another one. We have done this currently with our presentation component (where we replaced the former commercial product TouchPoint from OCLC with VuFind) and we have done this 2 years ago where we replaced the commercial Search engine FAST with the Lucene / SOLR solution.

But – although we exchanged an OCLC product (TouchPoint) we are still running with the DataHub from OCLC Leiden, NL– which is a very good choice for us.

And – this might be the most valuable and important aspect – keep this in mind, very very important!

With such a layered architecture using open interfaces **“You are not tied to a fixed product or commercial vendor”**. You can use a commercial component of a special vendor as a part of the solution if it's sensible – but you are free to change it if you come across a better one. And today – the solutions in the digital world are changing rapidly so you have to be flexible at any time.

Slide 4 - First stop for a dive: the content collection component.

Imagine we start a short boat trip on the swissbib “four lake district”. At each of the four lakes we will

make a little dive to get a better understanding about what is going on in this part.

The Lake – picture is a synonym for software layer (providing specialized functionality). The layers (lakes) are connected to each other via channels (our Easter eggs or interfaces as we called it formerly). This makes it possible to put them in a row from the left to the right through the whole “swissbib four lake district”. One can compare this journey with the flow of content or information in the service.

OK – the boat trip starts...! I hope you can swim.....

Slide 5: the contentCollector

Purpose of this component is to fetch content **from all the repositories** part of Swissbib. (The 5 Aleph IDS library systems, IDS partners, Rero, SNL, document repositories like Zora and retroseals, Archive material and more)

We can fetch the content via different channels as you can see here.

Additionally we have fetched and stored the complete GND repository because we want to use the GND variants (later more).

The fetched content is preprocessed, validated, partially transformed and the latest version of every record is stored in a data store.

OK – rowing further, using the directory API. This API is used to exchange content between the content Collector and the next lake or layer – our data hub.

Slide 6 Dive Deeper – Data Hub:

again – be prepared for a next swim

Slide 7: More detailed view on Data Hub

Within the data hub the **content collected** in the layer before **will be refined**. **What do I mean by refined?**

(Refinement is summarized in the first part of the slide)

Why do we call it Data Hub?

1) “The result of the processing is used internally by swissbib and provided to external services”

We can see examples of the external services on the slide

→ e-lib.ch (around 90% of their content comes from swissbib) / MapPortal and Worldcat. Currently we send only data to WorldCat in the future there will be a bi-directional exchange so we can enrich our data with content from WorldCat.

2) “We connect a multitude of single content resources on a national and international level”

→ *show the MapPortal example* <http://suche.kartenportal.ch>

→ → swissbib is collecting map related bibliographic meta data

→ → only this map related bibliographic meta data as a whole is fetched via OAI from Data Hub

→ → users can search within the Map Portal

→ → there is a backlink to swissbib for the detailed MetaData

→ → back in swissbib the user will be connected to the original source of the data as well as to World Cat (if useful)

possible example for duplicates and clustering:

search for: Das Strafrecht in der Krise der Industriegesellschaft

<https://test.swissbib.ch/Search/Results?>

[lookfor=Das+Strafrecht+in+der+Krise+der+Industriegesellschaft](https://test.swissbib.ch/Search/Results?lookfor=Das+Strafrecht+in+der+Krise+der+Industriegesellschaft)

to show how duplicates are brought together

slide 8 Dive deeper to the heart of Search

slide 9: The Search server as the heart of every discovery service for end users

We have reached the layer of the swissbib Search services.

The content for the Search engines is coming from the DataHub we have seen before over a so called SRU catcher (<https://github.com/swissbib/srwMessageCatcher>)

(every single new, updated or deleted record is sent via SRU to this layer.) The record “is caught” and out into a process called SDP (for sure it's not a new Internet pirate party). It means SearchDocument processing. **The already refined content from DataHub is again specialized** prepared to meet the search needs. What happens here is:

- full-text enrichment with TOCs and Abstracts
- enrichment with GND variants (remember we are hosting the complete GND content from DNB)
- VIAF and MACS is coming soon

This process is tailored for every single index.

In fact we are running two indexes at the moment

- a) a bigger one known in swissbib green with 22 million records
- b) a smaller one specially tailored for the Basel / Bern discovery solution. This is a subset with around 7 million records.

The reason why we use two specialized indexes:

This gives us the possibility to prepare the content in a way which **fits better the needs of a single institution** (for example related to facet preparation or special boosting desires)

In swissbib green which is a more general service sometimes it doesn't make sense to implement specialties of single institutions.

Another reason for such a tailored index: Currently Basel/Bern is **loading their eBooks** into the Search Engine. This is something we have in mind for swissbib green too but not just at the moment. So there are reasons why it could make sense for an institution to manage their business on the Search engine level by themselves.

The Search server itself (currently SOLR – could be enhanced or replaced by ElasticSearch) is based on Lucene 4.x – the actual standard in the Information Retrieval world.

By the way: All known discovery solutions of commercial vendors in the library world are based on Lucene.

Again: look at the interface circle <http://search.swissbib.ch/solr>:

[It's the interface used by swissbib internally and available for external services.](http://search.swissbib.ch/solr) We are going to **use it in our hands on** section to build our own presentation component.

Additionally this interface could be used for educational purposes teaching students principals of IR on real systems.

OK – now to the last stop:

Slide 10: Dive deeper – join the user and their devices

Slide 11: detailed view on presentation layer

The major task of a presentation component or layer – as the name says - is to provide hopefully relevant information to users requests.

This providing of information could be done in two ways:

- firstly:

the classic channel via web browsers. Users are interacting in a traditional way with either well known OPACS or more modern discovery interfaces like swissbib. Nowadays not only with larger desktops but with small devices.

- secondly

provide the relevant information in formats which enables other services to understand the information without any human interaction.

This enables these external services to do and process the content in a way you itself don't know and cannot influence. It's effectively out of your control. But this is the main idea of the Web of data or “Semantic Web)

Swissbib is at least on the way to fulfill both challenges.

- Related to user – interaction via web browser:

Four years ago we created a modern surface which was really new at this time. It has some elements formerly often known as “Web 2.0 elements”.

- Related to “provide machine understandable information”:

The SRU interface is a standardized protocol often used in the library community. Using this API as part of the presentation component layer Lionel Walter was able to implement his own mobile solution which in turn we thankfully could use and provide users for their modern devices. Another example for an external service using our SRU interface is the KVK (Karlsruher Virtueller Katalog)

Time goes on - Next steps we will do:

- During the first part of the upcoming year we want to **change the current user interface** into a so called **Responsive Design** which fits better to the needs of small devices.

- We want to **transform the whole swissbib content into RDF** and provide it using a more convenient API (probably SPARQL) which would be the first step to connect swissbib closer to the often called “Web of Data”

How can we transform the content into RDF

As you now should already have learned: Because of our layered and open system we can do this just by using (and extending) an already available component (DataHub or document processing for the Search Engine we have seen in the layer before) or by extending the infrastructure with an additional component. The second one is more probable.

Back to the more traditional way user interacts with the information we are providing – **our new VuFind based** presentation component.

Where does the content come from?

As you can see the main source of information is the **Search interface search.swissbib.ch** (the 22 million documents refined in the DataHub).

But beside this information we are able to include **additional information** e.g. all the article targets currently well known (Summon, PrimoCentral) These are not part of swissbib green because of licensing reasons but part of the swissbib Basel / Bern project.

Authentication and Opac

For authentication we **trust on Shibboleth (Switch AAI)** as we did just at the beginning of project.

New with the upcoming system will be the inclusion of a Switch AAI guest account for non-universal users and the general infrastructure to include OPAC functionality for every of the library systems part of swissbib.

Demo on testvf.swissbib.ch

This OPAC functionality we have just seen won't be part of the new presentation component in swissbib green (in contrary to Basel/Bern) just at the beginning. (We have in mind to deploy VuFind version of swissbib green in Beta mode (parallel to the current one) at the end of this month -October-).

The reason why it isn't included just at the beginning: I want to test it more seriously because it should be mature. But it's on our roadmap to be included as soon as possible

We are now at the end of part I - overview of the swissbib architecture.

I hope you now have a **better feeling for the principles swissbib is built** upon which are (to summarize again):

- layered system with dedicated components
- open interfaces used internally and by external services
- side by side cooperation of commercial and Open Source components.

I think **we are now well equipped for the second part** where I want to demonstrate the process you have to do to build a presentation component which could be used by yourself or could be part of the swissbib infrastructure.

OK – perhaps you can make a few minutes gymnastics or aerobic after such dusty theory although we have already done some swim exercises.

Part II – Welcome to the hands-on session

Slide 12 “Hands on – swissbib for the short distance runner”

What should **be the result of this hands on:**

Aims of the workshop:

First aim:) at the end of the next 30 to 40 minutes there should be a running presentation component here on my local laptop comparable you are using while accessing a domain like www.swissbib.ch

Second aim) this local component should be tailored in two aspects

first aspect to tailor the view:) you do not want to use the organization of existing library networks like Nebis, rero etc.

but you want to define a virtual logical view for your project. Within this logical view you want to assemble library institutions related to your purpose.

Logical views could be created following e.g. functional aspects or regional aspects.

As an example I want to show you the already available so called “Jus portal” - an information portal.. Within this portal you can find all the libraries part of swissbib especially dedicated to legal literature or libraries which possess a considerable asset of legal literature.

Demonstration

jus.swissbib.ch

Search for **Günter Stratenwerth** and open **“Die Straftat”** with 6 institutions

As you can see:

The tailored view consists of 2 groups.

The first one “Juristische Bibliotheken” with libraries dedicated to legal literature

The second one with institutions having a reasonable asset of legal literature.

How can we do this? By using a tool we implemented called libadmin – again freely available on GitHub.

Introduction to libadmin

For our workshop I created a special view facing more (but not exclusively) regional aspects. Therefore I used our libadmin tool. You can access and play around with a test installation

<http://admin.swissbib.ch/libadmintest>

→ show the tool and the workshopview

What I'm going to say as part of the demonstration

- You can see institutions, groups and views
- views contain groups and groups contain institutions
- you can combine these parts as you want
- For our workshop I defined a view called epflworkshopview
- the groups northwest and epflworkshop are part of this view
- now we take a look into the groups to see which institutions are assigned
 - epflworkshop (Group) contains institutions from the western part of Switzerland (lake Geneva, Neuchatel plus ETH Zurich)
 - northwest: some institutions around Basel

OK: This was the first aim of our tailored local installation to use a logical view!

b) The second goal of our tailored local installation:

We want to use a dedicated layout (CSS)

And remember: We are going to use the interface <http://search.swissbib.ch/solr/sb-biblio/select> provided by our Search service layer (Part I)

The interface we have seen during the architecture part I on the Search Server layer.

slide 13: Prerequisites

This slide lists the necessary basic Software components to be installed on your device that enables you to run the Presentation component locally.

As you can see – the requirements are not high Apache as a webserver, a MySQL database and some PHP extensions. To fetch the source code you have to install the Git tool.

I don't expect you to retrace the activities within this hands on immediately. But feel free over there you can find some printouts with detailed instructions what has to be done.

OK let's start (on the laptop)

slide 14: install the general functionality on your local machine (not tailored to institutional requirements or desires)

```
mkdir /usr/local/vufind/epfl [APP_DIR]
```

```
cd $APP_DIR
```

Once we got the source code from GitHub we have to change the Git branch to feature/epflworkshop

Then we can start the script as user root.

I made a script to automate the process – within the cookbook it's described in more detail.

Ok hopefully the script finishes successfully so we should be able to start the local application with <http://localhost/epfl>

slide 15: meet the institutional requirements for your local installation

Ok – now where a general installation of a local presentation component is up and running on your local machine, we want to tailor it to the institution's needs.

a) first we want to use the definitions of the epflworkshopview.

At the moment **we are working with the complete list of institutions** part of swissbib (more than 800 hundred).

→ **show the definition files**

But we **only** want to have the groups and institutions **part of the epflworkshopview**. **How to reach this?**

Again interfaces! We can use one provided by the **libadmintool**

- **change the configuration** and **start the script** to fetch them
- clear the cache
- reload the application

Ok in the full view we can see our groups but there is still an issue. The requests on search.swissbib.ch are using the complete repository of 22 Million documents. But we want only the documents related to our defined institutions.

Ok, configure a search restriction, clear the cache and reload the application.

Now we get only documents related to the institutions.

Slide 16: Change the design

b) next institutional requirement is a special layout

For the purposes of this workshop I made it very simple and defined already two so called themes part of the feature/epflworkshop branch
Only with an adapted CSS color

Now you can switch between two colors.

Slide 17: Some basic principals of the View concept in swissbib / VuFind:

A few words to the view concept in VuFind / swissbib

- views (or themes) are hierarchical organized
- VuFind provides a general view called root which is used by the VuFinds themes called blueprint and jquerymobil. They are working on an additional view based on the bootstrap framework (for Responsive Design)
- for the upcoming version we wanted to present our users the same interface they were used in the last

4 years (green swissbib).

Therefore we had to re-implement /refactor quite a lot of code in the basic swissbib theme (which uses parts of blueprint)

- this basic theme is the parent for swissbibmulti (Basel / Bern) with two targets and swissbibsingl (one target)

- swissbibsingl itself is used for the two play-around workshop views

→ so only very few configurations had to be done.

We are at the end of Part II - our hands on session

slide 18: A compilation of available resources

and now part III: Open discussion.

Slide 19 Part III Open Discussion.

We have reached the third part of our workshop – Discussion!

Some suggestions I would like to talk about with you from my point of view are already on the slide.

But now it's your turn!

Additional stuff from my point of view (**not in introduction of discussion**)

Something like could be in my mind for further discussion (if it turns out in this way)

Sometimes I call swissbib a “**working bench**”.

During our journey through the 4 layers of the solution in the first part you have seen that every single component provides one can say landing stages (places) which could be part of your idea or enhance your service or solution.

This leads to the first question from my side for the discussion about:

Your ideas and desires to use the swissbib infrastructure?

Secondly – If you are motivated – how could you being **involved in the further swissbib development?**

Slide 20: Thank you!