



# Welcome

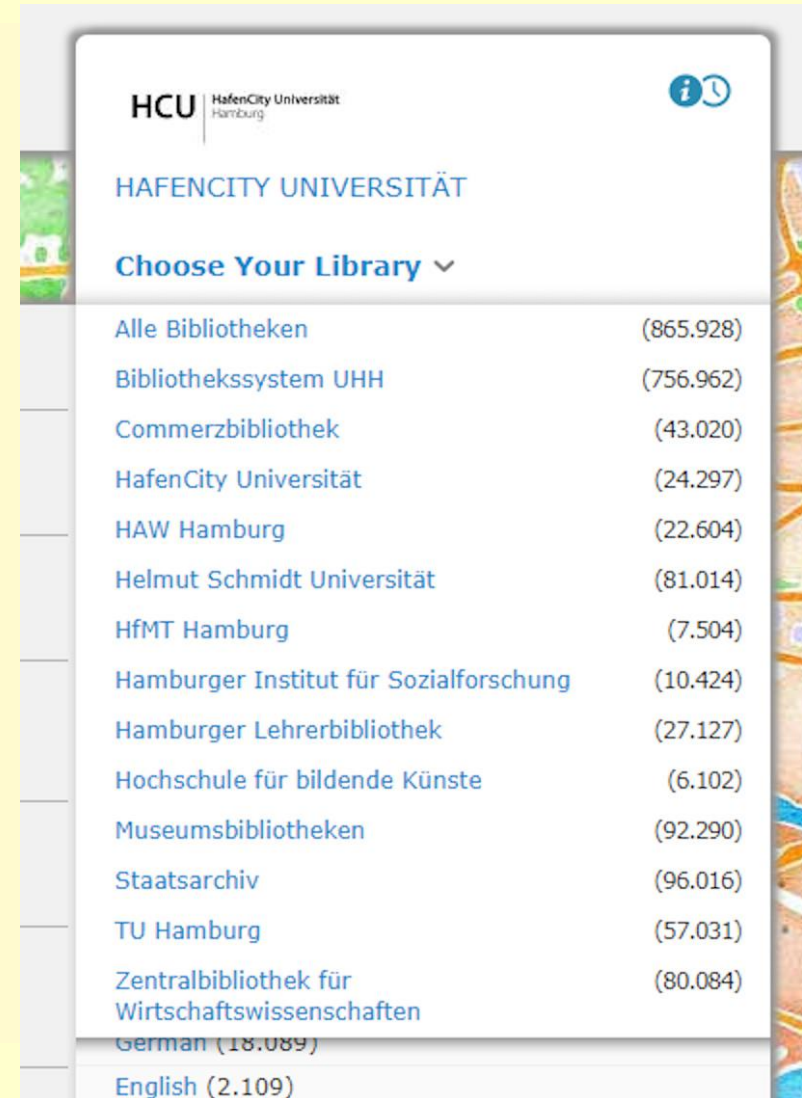
Extension and Modularization of VuFind:  
Ideas – and Questions

Hajo Seng, SUB Hamburg  
Hamburg, 1.10.2020

# Why extending VuFind? Isn't it enough?

## specific usecases

e.g. a „regions“ catalogue



The screenshot shows a dropdown menu for library selection. At the top, it displays 'HCU Hafencity Universität Hamburg' with an information icon. Below this, the text 'HAFENCITY UNIVERSITÄT' is shown. The main heading is 'Choose Your Library' with a downward arrow. The menu lists various libraries with their corresponding item counts in parentheses.

Library	Count
Alle Bibliotheken	(865.928)
Bibliothekssystem UHH	(756.962)
Commerzbibliothek	(43.020)
HafenCity Universität	(24.297)
HAW Hamburg	(22.604)
Helmut Schmidt Universität	(81.014)
HfMT Hamburg	(7.504)
Hamburger Institut für Sozialforschung	(10.424)
Hamburger Lehrerbibliothek	(27.127)
Hochschule für bildende Künste	(6.102)
Museumsbibliotheken	(92.290)
Staatsarchiv	(96.016)
TU Hamburg	(57.031)
Zentralbibliothek für Wirtschaftswissenschaften	(80.084)
German	(18.089)
English	(2.109)

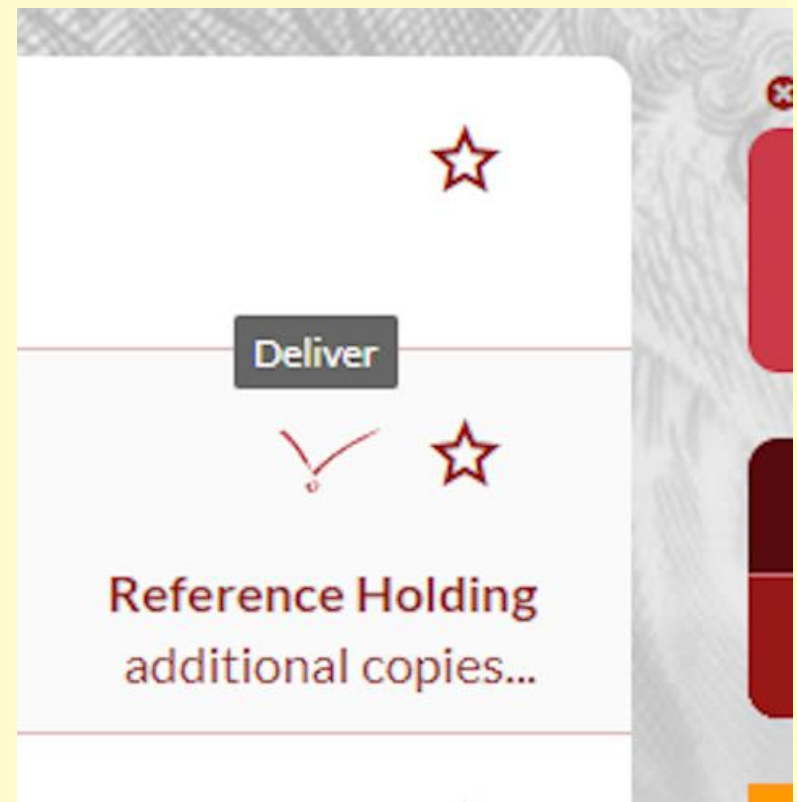
# Why extending VuFind? Isn't it enough?

specific usecases

e.g. a „regions“ catalogue

real extensions of functionality

e.g. connected delivery services



# Why extending VuFind? Isn't it enough?

## specific usecases

e.g. a „regions“ catalogue

## real extensions of functionality

e.g. connected delivery services

## particular requirements

e.g. a special year facet



Year of Publication	
1900-1999	✓
1990-1999	166,925
<u>1980-1989</u>	116,759
1970-1979	90,493
1960-1969	71,060
1950-1959	55,217
1940-1949	33,923
1930-1939	34,024
1920-1929	32,136
1910-1919	27,913
1900-1909	24,434

# Why using a modular design?

## E.g. beluga-core consortium:

consortium of libraries with and without programming powers

yet: 7 libraries in northern Germany

aim: adjust catalogue frontend to specific needs

shared development of VuFind-extensions

sharing features with other libraries

<b>AIM:</b>	<b>local functionality:</b>	creating a module
	<b>global functionality:</b>	preparing a pullrequest at VuFind (and creating a temporary module)

# Extending modules: an approach

## modules: extending namespaces

```
],
'mixins' => [
  'relevancepicker',
  'notifications',
  'belugaconfig',
  'delivery',
  'daiaplus',
  'extendedfacets',
  'libraries',
  'searchkeys',
  'dependentworks',
  'recorddriver',
  'beluga-core-base',
],
```

mixins-theme with empty templates

```
*/
namespace Delivery;

use Delivery\Driver\PluginManager;

class DataHandler {
```

## themes: using mixins

naming: <module>-<template>.phtml

basic mixin with empty templates

## languages, configs: using parents

```
@parent_ini = "../../../languages/en.ini"
```

others: database, composer, ...

# Interfaces for module extensions

**namespace**                      **convention: themes & modules have the same name**

**module.config.php**   **native inheritance**

**ViewHelper**                      **naming: <module>\View\Helper<theme>**

**AjaxHandler**                      **inheritance possible, often not necessary**

**Controller**                      **inheritance possible, often not necessary**

**classes / objects**                      **discussion**

**PluginManagers**                      **discussion**

# Stacking modules

```
namespace Libraries\Search\Factory;
...
use RecordDriver\Search\Factory\SolrDefaultBackendFactory as BackendFactory;

class SolrDefaultBackendFactory extends BackendFactory
{
```

```
namespace RecordDriver\Search\Factory;
...
use LuceneHelper\Search\Factory\SolrDefaultBackendFactory as BackendFactory;
...
class SolrDefaultBackendFactory extends BackendFactory
{
```

There are some popular objects which are overridden by several modules  
the order of modules is important  
use-declaration has to be changed if modules are plugged in or out  
extension-chains can be avoided in AjaxHandlers, Controllers, ViewHelpers  
but not everywhere



# Stacking modules: PluginManagers

```
namespace VuFind\Search\Params;
...
/**
 * Default plugin aliases.
 *
 * @var array
 */
protected $aliases = [
    'browzine' => 'VuFind\Search\BrowZine\Params',
    'combined' => 'VuFind\Search\Combined\Params',
    'eds' => 'VuFind\Search\EDS\Params',
    'eit' => 'VuFind\Search\EIT\Params',
    'emptyset' => 'VuFind\Search\EmptySet\Params',
    'favorites' => 'VuFind\Search\Favorites\Params',
    'libguides' => 'VuFind\Search\LibGuides\Params',
];
```

PluginManagers can't be  
overridden

doubled aliases & factories

```
namespace SearchKeys\Search\Params;
...
class PluginManager extends \VuFind\Search\Params\PluginManager
{
    /**
     * Default plugin aliases.
     *
     * @var array
     */
    protected $aliases = [
        'browzine' => 'VuFind\Search\BrowZine\Params',
        'combined' => 'VuFind\Search\Combined\Params',
        'eds' => 'VuFind\Search\EDS\Params',
        'eit' => 'VuFind\Search\EIT\Params',
        'emptyset' => 'VuFind\Search\EmptySet\Params',
        'favorites' => 'VuFind\Search\Favorites\Params',
        'libguides' => 'VuFind\Search\LibGuides\Params',
        'mixedlist' => 'VuFind\Search\MixedList\Params',
        'pazpar2' => 'VuFind\Search\Pazpar2\Params',
    ];
}
```

# Stacking modules: PluginManagers

```
namespace VuFind\Search\Params;
...
class PluginManager extends \VuFind\ServiceManager\AbstractPluginManager
{
    /**
     * Default plugin aliases.
     *
     * @var array
     */
    protected $aliases = [
        'browzine' => 'VuFind\Search\BrowZine\Params',
        'combined' => 'VuFind\Search\Combined\Params',
        'eds' => 'VuFind\Search\EDS\Params',
        'eit' => 'VuFind\Search\EIT\Params',
        'emptyset' => 'VuFind\Search\EmptySet\Params',
        'favorites' => 'VuFind\Search\Favorites\Params',
        'libguides' => 'VuFind\Search\LibGuides\Params',
    ];
}
```

PluginManagers can't be overridden

Is there a better solution with VuFind 7?

```
namespace VuFind\Search\Params;
...
class PluginManager extends \VuFind\ServiceManager\AbstractPluginManager
{
    /**
     * Default plugin aliases.
     *
     * @var array
     */
    protected $aliases = [
        'browzine' => \VuFind\Search\BrowZine\Params::class,
        'combined' => \VuFind\Search\Combined\Params::class,
        'eds' => \VuFind\Search\EDS\Params::class,
        'eit' => \VuFind\Search\EIT\Params::class,
        'emptyset' => \VuFind\Search\EmptySet\Params::class,
        'favorites' => \VuFind\Search\Favorites\Params::class,
        'libguides' => \VuFind\Search\LibGuides\Params::class,
        'mixedlist' => \VuFind\Search\MixedList\Params::class,
        'pazpar2' => \VuFind\Search\Pazpar2\Params::class,
    ];
}
```

# Solutions and questions

themes

**mixins, naming-conventions**

language- & configurationfiles

**inclusion of parents**

modules

module.config.php

**native inheritance**

ViewHelpers

**naming-conventions**

AjaxHandlers, Controllers

**avoid inheritance**

classes / objects

**manipulate use-declarations**

PluginManagers

**override (and double) properties**

others (database, composer)

**additional configuration snippets**

# Solutions and questions

## AIM: pluggable extensions

modulestack: dependencies of stacked objects:

look after the order of modules

change use-declarations

**not pluggable** (but almost)

translation- and configfiles: change parent-declarations

**not pluggable** (no source code affected)

themes: **completely pluggable**

others: **another topic ...** (particularly composer-extensions)


# Thank you very much


## Hajo Seng

Von-Melle-Park 3  
20146 Hamburg

040 / 4 28 38-8336  
hajo.seng@sub.uni-hamburg.de

[www.sub.uni-hamburg.de](http://www.sub.uni-hamburg.de)

 [facebook.com/stabihh](https://facebook.com/stabihh)

 [twitter.com/stabihh](https://twitter.com/stabihh)

[www.beluga-core.de](http://www.beluga-core.de)  
[github.com/beluga-core/core](https://github.com/beluga-core/core)