

## Kontraktbilag A1 – Teknisk dokumentation

# Technical documentation for UCC.dk

This document describes technical documentation for the site ucc.dk. The document contains:

## [Data import and the bridge](#)

[UCC Services Bridge](#)

[SIS import and signup](#)

[SIS import](#)

[SIS signups](#)

[CUBE import](#)

[UMS import](#)

[PURE import](#)

[PeopleXS import](#)

## [Drupal](#)

[Drupal logical structure](#)

[Drupal core and modules](#)

[Sections on the site](#)

[International section](#)

[Node types](#)

[Teasers](#)

[List and search pages](#)

[Peytz Mail integration](#)

[Search and indexing](#)

[Access and permission rules](#)

[Logic for Section editor](#)

[Theming](#)

## [Coding standards and documentation](#)

### [Logging](#)

[Server Logging](#)

[Logging in Drupal](#)

[Logging in UCC Services Bridge](#)

### [Infrastructure](#)

[Servers](#)

[Hosting](#)

[Service Bridge](#)

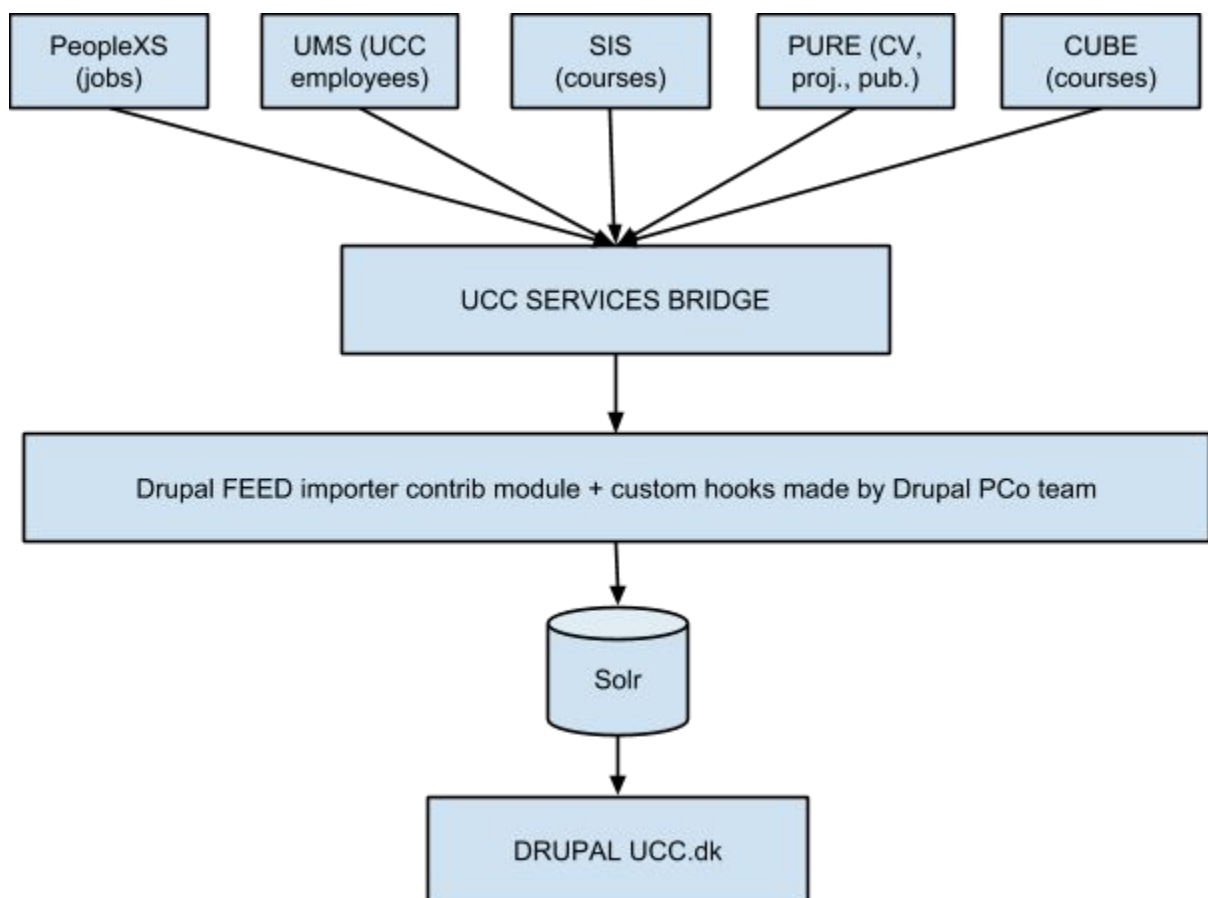
### [Links](#)

## Data import and the bridge

A large part of the content on ucc.dk is imported through a Peytz-developed data bridge, that formats the content and streamlines the values for easier integration with Drupals way of organizing content.

Agreements about import to ucc.dk is described in the document 'UCC Service Bridge import agreements'. This document describes the times of the imports, the import methods and rules for imports.

### UCC Services Bridge



The bridge is hosted by Peytz & Co and to accommodate the requested business logic there is custom logic implemented in `ucc_site` module in file `plugins/FeedsSisNodeProcessor.inc` and in `hook_feeds_presave` in `ucc_site` module.

The bridge has a logging system telling when a feed was built and when there is a request to

every particular feed, i.e. when ucc site is making request to import data from feed or someone is trying to make unauthorized access. The imported nodes have an import ID that points to a generated feed on the bridge. Thus the origin of every imported node can be backtracked to a certain import. In total there are around 10000 nodes updated every day.

## SIS import and signup

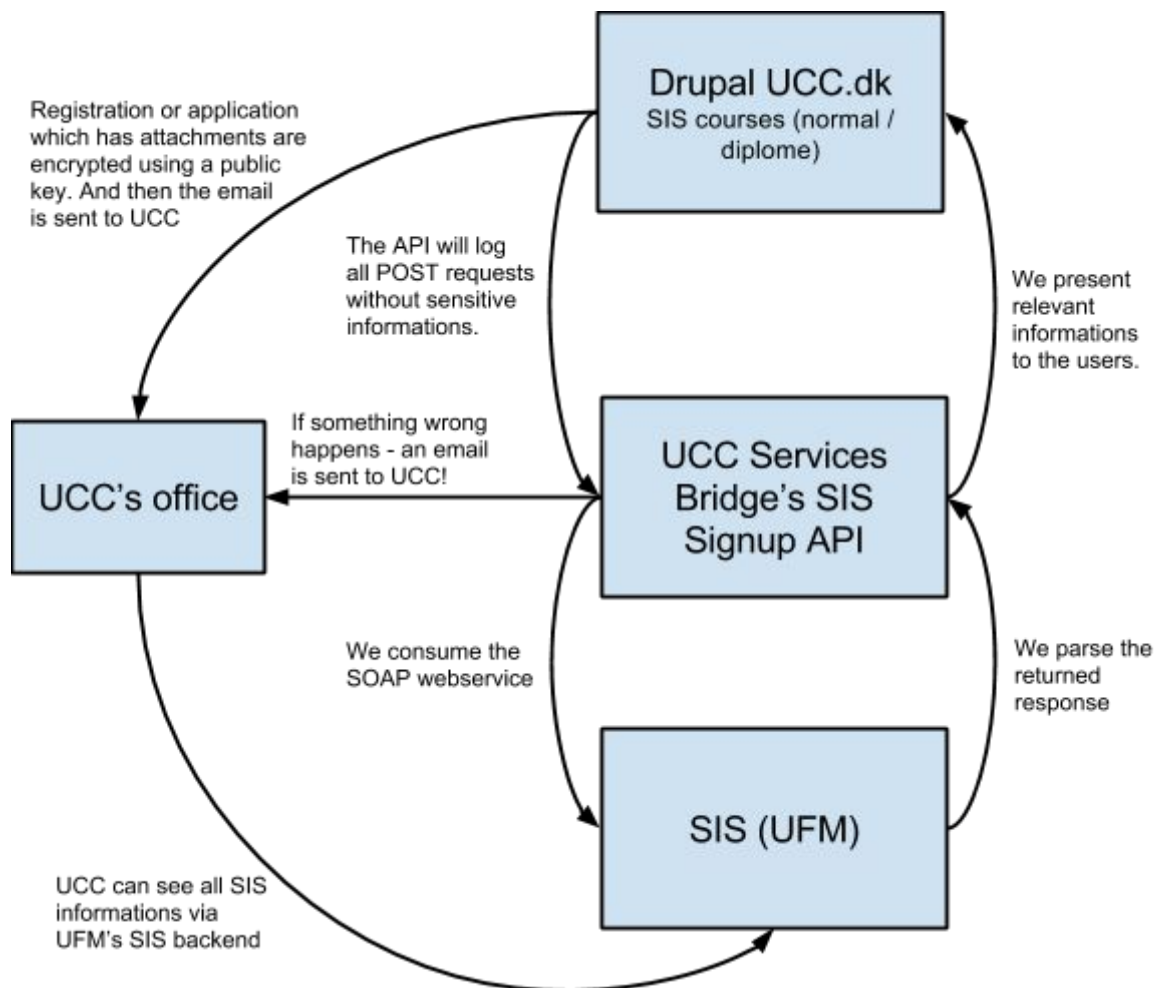
### SIS import

SIS courses/modules are imported from SIS through the UCC Service Bridge (<http://ucc-services-bridge.aws.peytz.dk/sis>) to the Drupal feed importer. The courses/modules are added to Drupal with content type 'Course and certification SIS' and includes a link to a SIS signup form.

The contact person on courses is found by matching the email address or initials from the feed with an existing employee from the UMS import. This is done in `ucc_site_feeds_presave()`, using the function `_ucc_site_get_employee_by_mail()` in `ucc_site.module`. The matching of employees and email address/initials is cached, so in case wrong employees start turning up on the imported courses, start by clearing the cache.

Courses and modules from SIS should be unpublished at start date + 14 days, but never later than end date. Courses/modules are usually removed from feed at start date, but Drupal makes sure not to unpublish it until 14 days later. If a course is removed from feed before start date, we assume it's cancelled, and it should be unpublished in Drupal.

## SIS signups



Anonymous users can apply for a course or a diploma module. SIS course application is implemented in `ucc_site` module file `ucc_site_sis_application_module.inc`. There are two kinds of SIS application forms:

- SIS diploma module: Content type 'Course and certification SIS' with type='Diplomuddannelser'. Url is `tilmelding/uddannelse/<node-id>` where node-id is NID of sis course node. Email should be sent to user and [tilmelding\\_diplom@ucc.dk](mailto:tilmelding_diplom@ucc.dk).
- SIS course: Content type 'Course and certification SIS' with type='Kurser'. Url is `tilmelding/kursus/<node-id>` where node-id is NID of sis course node. Email should be sent to user and [tilmelding\\_ufev@ucc.dk](mailto:tilmelding_ufev@ucc.dk).

When a user signs up for a course/module using this form, the info (except payment info and attached documents) is sent to SIS through the UCC Service Bridge (<http://ucc-services-bridge.aws.peytz.dk/signup/list>), and an email is sent to the user and to UCC by email in an encrypted state because of the confidential information user enters.

To send email, both text and attachments are encrypted using `openssl_pkcs7_encrypt()` that creates .p7m file, that is sent to UCC. Encryption is made using `ucc-peytz.pem` key located in `ucc_site` module in `certificates/` subdir.

Files that user attaches during signup are deleted right after signup process is finished.

## CUBE import

CFU courses are imported from CUBE through the UCC Service Bridge (<http://ucc-services-bridge.aws.peytz.dk/cube>) to the Drupal feed importer.

The courses are added to Drupal with content types 'Course CUBE' and 'Course CUBE lesson':

- 'Course CUBE' nodes contains information about the course, a 'Tilmeld dig' button with link to an external page, and course dates are listed in the 'Kursusdatoer' list in the bottom.
- 'Course CUBE lesson' nodes are the specific course dates related to a course and the info is shown in the 'Course CUBE' nodes.

Courses are unpublished at start date + 12 hours (the same for both one day courses and longer courses). If a course is cancelled, UCC webmaster deactivates/removes it manually on `ucc.dk` (under scheduling options).

## UMS import

Basic information about UCC employees is imported as xml from UCC through the UCC Service Bridge (<http://ucc-services-bridge.aws.peytz.dk/ums>) to the Drupal feed importer. The data is added to Drupal with content type 'Employee'.

The data is shown in the relevant employee pages (`/om-ucc/medarbejdere/<initials>`) in the top part, and contains image, name, contact info, position info and department.

When an employee is no longer present in the xml file, the employee node should be unpublished in Drupal. Drupal Feed importer should make sure that import to the Bridge is done before importing to Drupal. The import from the Bridge to Drupal Feed Importer is split into several imports, because cron job can't handle all content in one job

## PURE import

Additional information about UCC employees is imported from PURE (`ucviden.dk`) through the UCC Service Bridge (<http://ucc-services-bridge.aws.peytz.dk/pure>) to the Drupal feed importer.

The data is added to Drupal with content types 'Content PURE', 'Project PURE' and 'Publication PURE'. The data is shown in the relevant employee pages (`/om-ucc/medarbejdere/<initials>`) as a general text and tabs 'CV', 'Projekter' and 'Publikationer'. The tabs are only shown if data is available for the specific tabs.

- 'Content PURE' contains information (presentation and primary/secondary working areas) shown on top of the page together with info from UMS. It also contains info for the CV tab (Education, qualifying education, recruitments, learning programs, language skills, organizational skills)
- 'Project PURE' contains information about projects involving the employee (title with link to relevant page on ucviden.dk, type and year) and is shown in the 'Projekter' tab.
- 'Publication PURE' contains information about publications where employee is author (title with link to relevant page on ucviden.dk, type and tags), and is shown in the 'Publikationer' tab.

PURE import should always be done after UMS imports are done. PURE import should only create Content/Project/Publication PURE node if the corresponding Employee node exists in Drupal.

As a part of the PURE import we check existing Content/Project/Publication PURE nodes in Drupal. If the corresponding Employee node is deleted from the site, then the nodes should be unpublished.

The import from the Bridge to Drupal Feed Importer is split into several imports, because cron job can't handle all content in one job

The scheme for importing is realised using elysia cron module and should guarantee that all PURE content and employees (UMS import) are imported during the night with minimal impact on performance.

## PeopleXS import

Job ads are imported from an external recruitment service through the UCC Service Bridge (<http://ucc-services-bridge.aws.peytz.dk/peoplexs>) to the Drupal feed importer.

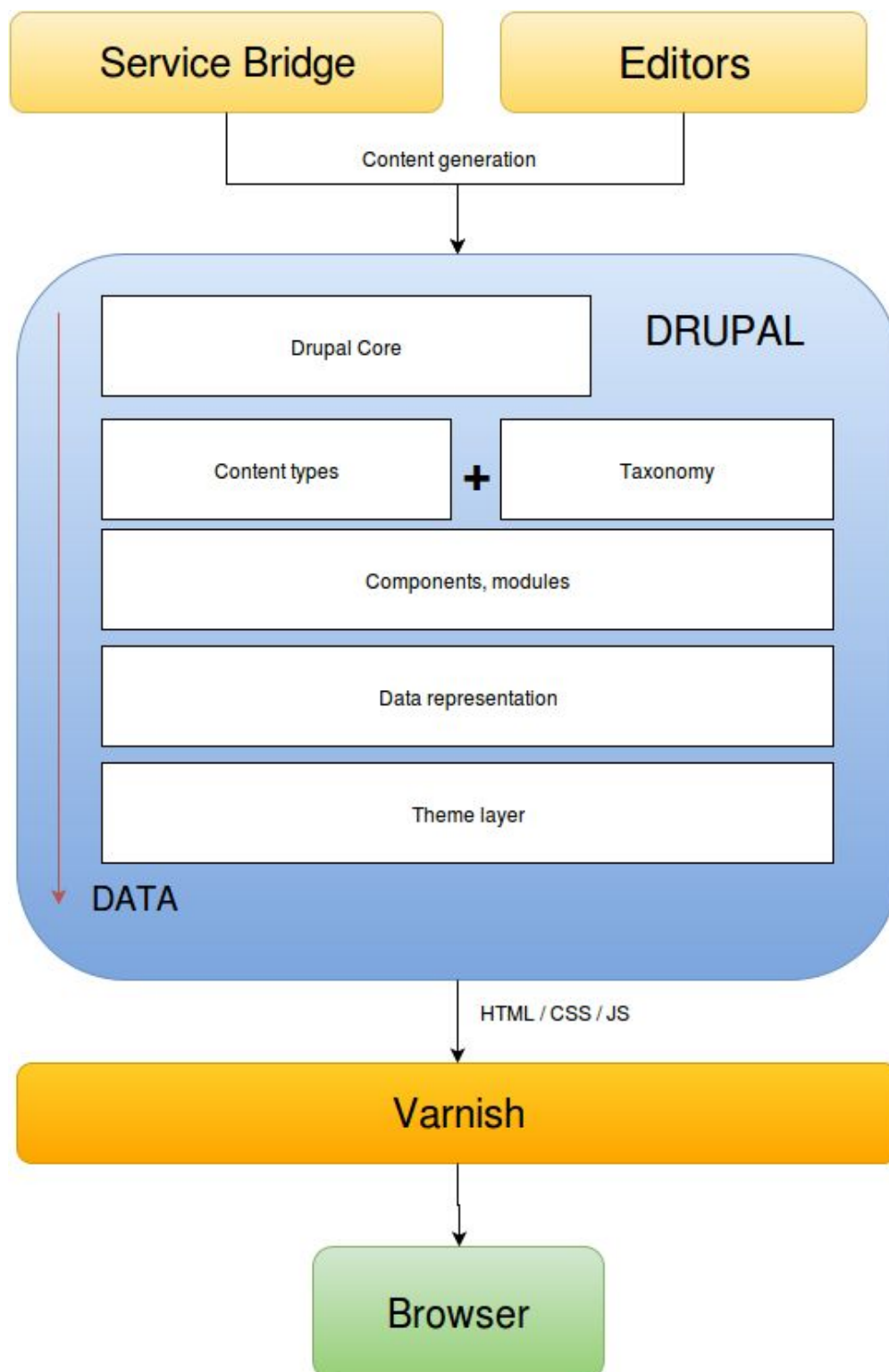
The data is added to Drupal with content type 'Job ads'. The contact person for the job ad is found by matching the email address from the feed with an existing employee from the UMS import. A button 'Søg jobbet' opens an application form from PeopleXS using iframe.

Job ads should be unpublished in Drupal at End application date (ansøgningsfrist) at 23.45. If a job ad is cancelled, UCC webmaster deactivates/removes it manually on ucc.dk.

# Drupal

## Drupal logical structure

Drupal logical structure of content generation and presentation:



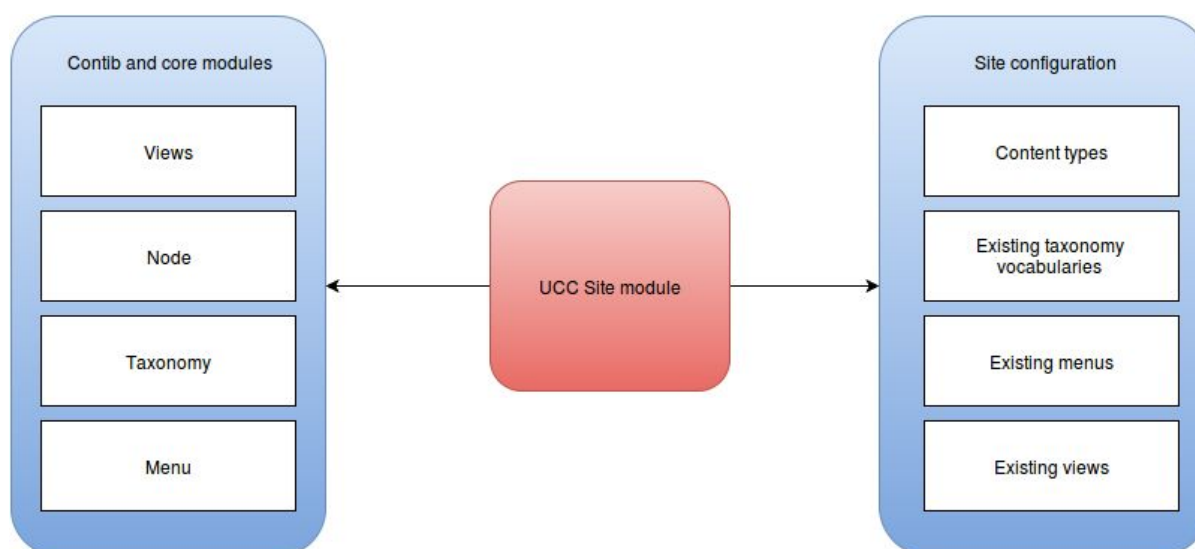


## Drupal core and modules

We use the current Drupal version 7.39

List of contrib modules and other modules, see appendix 1.

Modules are encapsulated and provides finished functionality. With one exception - `ucc_site` module. `ucc_site` is heart of ucc project, it depends on various contrib/core modules and existing site's configuration. It interacts with ucc specific content types, taxonomy vocabularies, menus etc



### Use of Cron and Cache

Cron is used to import content. From the Drupal side elysia cron module is used.

Varnish is used for caching. There are 2 modules to control varnish behaviour: expire and purge - first one is used to handle situations when cache needs to be cleared, and second communicates directly with varnish, ie purges varnish cache for some of urls.

### Patches

1. Expire module - Expire image styles when files are updated -  
<http://www.drupal.org/node/2490954>  
<http://www.drupal.org/files/issues/expire-image-styles-2490954-4.patch>
2. facetapi\_collapsible - Add collapsible display widget with checkboxes  
<https://www.drupal.org/node/1985648>  
[http://drupal.org/files/1985648\\_facetapi\\_collapsible\\_checkboxes\\_2.patch](http://drupal.org/files/1985648_facetapi_collapsible_checkboxes_2.patch)
3. Feeds - Patch to unpublish nodes not in feed (particularly useful for SIS).  
<https://drupal.org/node/1470530>  
<http://drupal.org/files/issues/unpublish-delete-entities-not-in-feed-1470530-134.patch>
4. file\_lock - Custom patch to satisfy ucc specific file handling algorithm. In order to satisfy privacy politics. All uploaded files are locked on site, but files that were uploaded in webforms are not locked intentionally, because they should be deleted alongside with webform submissions they are part of.

[http://gitlab.peytz.dk/drupal-projects-contrib/patches/raw/master/projects/file\\_lock/d7-file\\_lock-ucc\\_specific.patch](http://gitlab.peytz.dk/drupal-projects-contrib/patches/raw/master/projects/file_lock/d7-file_lock-ucc_specific.patch)

5. Media - Embedded documents in the WYSIWYG can be very hard to delete  
<https://www.drupal.org/node/2028231>  
[https://www.drupal.org/files/issues/media-embeddd\\_documents-2028231-15.patch](https://www.drupal.org/files/issues/media-embeddd_documents-2028231-15.patch)

## Sections on the site

UCC.dk is structured with a strong focus on the menu tree and less on taxonomies. However, the sections are taxonomy terms that are applied to top level. If top level menu item has section (it should have in normal conditions), then all its child menu items have same section. This fact helps us to render relevant sidebars across a section that are typical for this concrete section user is visiting at the moment. This is detected and added to a static variable in `ucc_site_init()` and is retrieved throughout the system with `drupal_static('ucc_site_section')`.

The section value can be added to all pages, but it will only have impact if the page is used as top level menu item. There are several taxonomy vocabularies, that are used as catalogues, eg 'institutions' or 'job titles'. But 'sections' vocabulary is used to separate content on sections and to grant editor access to edit nodes only in section he belongs to (see 'Access and permission rules').

## International section

There is an international section on UCC.dk site - <http://ucc.dk/international> which is translated in English. All logic is concentrated in `hook_language_negotiation_info` (see `ucc_site_language_negotiation_info()` function for details), which defines language provider that determines English on all urls beginning with "international".

## Node types

There are many almost identical node types. First of all these node types support the different business objectives of UCC, and also it makes it easier for UCC to finetune the permissions for the many roles, as they control the roles and permissions themselves.

The node types can be divided into these categories in terms of display:

- **Full width pages**
  - **Front page** - Full width page with the option of a large image and a call to action splash at the top of the page, menu links and teaser boxes.
  - **Section frontpage, Campaign page and Theme page** - Full width pages with the option of a large image and a call to action splash at the top of the page, text and teaser boxes.
- **Panel based box pages**
  - **Teaching material** - Visually close to the section frontpage but the right column teasers are added in a separate field, making it possible to keep a meaningful succession of teasers when below the smaller breakpoints

- **Text pages**
  - **Basic page** - Page with text, images, documents, gallery and possibility to add reference to an Employee node and right column teasers.
  - **News** - Page with text, images, gallery and possibility to add reference to right column teasers.
  - **Magazine frontpage** - Page with text, image and UCC magazine reference.
  - **Magazine article** - Page with text, UCC magazine reference and a possibility to add an image as background image or left side image and add a color scheme.
  - **Conferences and events** - Page with text, images, documents, practical info and categorization data. Possibility to add reference to sign up form and Employee node. Possibility to add right column teasers.
- **Course pages**
  - **Course and certification SIS** - courses imported from SIS showing text, practical info and categorization data, a reference to Employee node and a sign up link using SIS signup form.
  - **Course CFU** - courses imported from CFU showing text, practical info and categorization data, a reference to Employee node and a sign up link to external page. In the bottom of the page, references to course dates (Course CUBE lesson) will be shown.
  - **Course AMU** - manually created course pages for AMU courses showing text, practical info and categorization data, a reference to Employee node and a sign up link to external page. In the bottom of the page, references to course dates (Course AMU single) will be shown.
  - **Course misc** - Manually created course pages for miscellaneous courses showing text, practical info and categorization data, a reference to Employee node, right column teasers and a sign up link using webform node.
- **List pages**
  - **Diploma education** - Page with text, documents, right column teasers and the filters SIS codes and SIS diplome codes to retrieve a list with SIS modules.
- **Proxy pages**
  - **Menu proxy** and **Section page proxy** - Display content from an internally referenced page to make content appear more than once in the menu and keep the displayed menu.
- **Attached content**
  - **Course AMU single** and **Course CUBE lesson** - Course dates shown on Course AMU and Course CUBE pages.
  - **Content PURE, Project PURE** and **Publication PURE** - Info about employees imported from PURE and shown on Employee pages.
- **Employees** - The only node type that can appear as teaser and in full view. In full view it also shows related content from PURE.
- **Job ads** - Job ads imported from PeopleXS showing text, application info and link and reference to Employee node.
- **Webform** - Webform with text, form components, validation, submission handling, email recipients.

- **Iframe** - Page showing content in Iframe.
- **Sidebar and topbar**
  - **Layout: Sidebar** and **Layout: Topbar** are teaser size bars shown in side or top of the page according to section (see 'Sections on the site')
- **Teasers (quite a few)**. Explained below.

## Teasers

The teaser 'system' is based on the entity connect field type. There are two teaser fields, `teaser_boxer` and `teaser_right_column`. Every teaser is a single node and can be reused in different parts of the site.

Teasers are always displayed in the teaser view mode. Even when previewing. The full view is forced through `node_view()` with the view mode set to 'teaser'.

There is 17 different teaser types:

- **Text based teasers**
  - **Teaser: Bogomtale** - Book review with title, text, left aligned image and link.
  - **Teaser: Knapboks** - Title and text. Possible to add a left aligned image and a link in bottom.
  - **Teaser: Knapboks, lille** - same as 'Teaser: Knapboks' but half height.
  - **Teaser: Tekstboks med/uden billede** - Title, text and image. Possible to choose color scheme and select different display types: Image in top, Image in right side, colored title and full color either full height or half height.
  - **Teaser: Topbar-boks** - Title, text as link and small image.
  - **Teaser: Citatboks** - Quotation box for use in teaser grid. Possible to choose color scheme.
- **Media based teasers**
  - **Teaser: Billedgalleri, bred boks** - Double width teaser showing the first three images from the galleri. By clicking the teaser, the gallery opens in an overlay.
  - **Teaser: Medieboks** - Can either show a clickable image or a video, that can be played directly in the teaser.
- **Link lists:**
  - **Teaser: Dynamisk liste** - List with newest content filtered by news tags and content types. Possible to add link in bottom.
  - **Teaser: Linkliste** - List with links. Possible to select display types Simple, Boxed and Arrows.
- **References**
  - **Teaser: Kontaktboks** - Contact box with title, text and possibility to add reference to an 'Employee' node that shows name, image and contact info. A 'Kontakt' link in bottom of the teaser opens a contact form in an overlay.
- **Search**
  - **Teaser: Course search** - Search field and drop down with course types. Possible to choose between normal teaser or full width teaser. Possible to choose color scheme.

- **Theme boxes**
  - **Teaser: Temaboks A, høj, vertikalt ord, Yume** - Double height teasers with possibility to add vertical text with Yume font. Possibility to add images and links and color scheme.
  - **Teaser: Temaboks B, høj, vertikalt ord, FS PeleOne** - Double height teasers with possibility to add vertical text with FS PeleOne font. Possibility to add images and links and color scheme.
- **Pop up**
  - **Teaser: Alarmboks** - Teaser used to show message to user in a pop up.
  - **Teaser: survey** - Teasers used to show either message or survey from Surveyxact to end users in pop ups. Possibility to let pop up be shown after a certain amount of page views.
  - **Teaser: Newsletter signup** - Sign up for newsletter.

## List and search pages

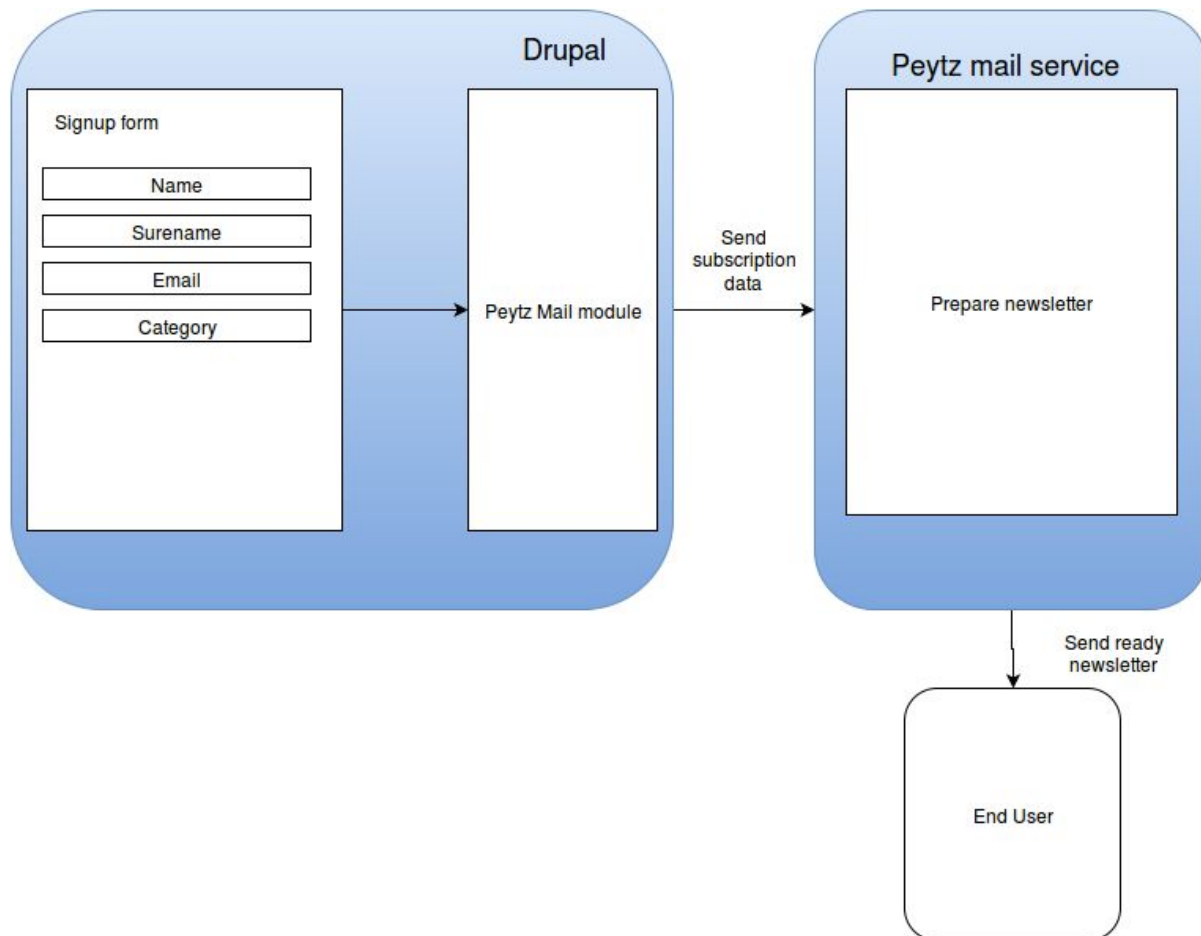
There is a couple of panel based list pages on the site:

- **News list**  
/aktuelt/nyheder  
Shows teasers for published News nodes and search fields with free text search.
- **Job page**  
/om-ucc/job-i-ucc/ledige-stillinger  
Shows teasers for published Job ads nodes or a default text 'Vi har i øjeblikket ingen ledige stillinger.' if there is none.
- **Search page**  
/soeg  
Search page with free text search and filters 'Informationssider' (Basic page), 'Kurser og videreuddannelser' (Course and certification SIS, Course CFU, Course AMU, Course misc), 'Konferencer' (Conferences and Events), 'Nyheder og magasinartikler' (News, Magazine frontpage, Magazine article) and 'Medarbejdere' (Employees).
- **Employee search**  
/om-ucc/medarbejdere  
Shows teasers for published Employee nodes and search fields with free text search and filter for Institutions taxonomy.
- **Course search**  
/videreuddannelse/find  
Shows teasers for published nodes with content types Course and certification SIS, Course CFU, Course AMU, Course misc and Conferences and events. Search fields lets the end user use free text search and sort by date or by alphabetical order. Also these filters are shown in left side: Type, Working area, Course locations, Semesters and Course category.



## Peytz Mail integration

UCC site is integrated with Peytz Mail service by `peytz_mail` and `ucc_site` modules. The main scope of peytz mail module is prepare and send data to peytz mail service. The data will be used by peytz mail to compose and send newsletter to users providing them maximum relevant information.



## Search and indexing

Apache Solr is used for search functionality. Solr index is replenished with every cron run - every 15 min. It means, that new created node doesn't find place in index right after creation. `Search API module` is used to integrate Drupal with Solr. All search pages and blocks are developed using views, that are integrated with Solr through SearchAPI.

Content types that are indexed:

- Basic page
- Campaign page
- Conferences & events
- Course AMU
- Course AMU single
- Course CUBE
- Course and certification SIS

- Course misc
- Diploma education
- Employee
- Front page
- Iframe
- Job ads
- Magazine article
- Magazine front page
- Menu proxy
- News
- Project PURE
- Publication PURE
- Section frontpage
- Section page proxy
- Teaching material
- Theme page

Indexed fields:

Title, Manchet, The main body text, Created date, Author, Author external, Section, UCC magazine, News tags, Name, Email, Phone number, Job title, Department, Institution ,UMS id, Positions, Resumé, Term ID, Employee first/last name, Employee phone without space, Working place, PeopleXS ID, Course ID, Course category, Price, Course location, Contact email, Kursusnr, Course city, Type school, Working area, Course type, Start semester, Course date, Start date, End date, Description, Education code, Education description, PriceSIS codes, SIS description, SIS ID, SIS specification, SIS diplome codes, Publication title, Publication description, Project description, Project title, priority\_search Position,

There are some customizations in search mechanism, in order to provide better search results. See `ucc_site_entity_property_info_alter` function. For example, some fields as name, surname and phone were assembled into one mate field, it improves search result

## Access and permission rules

There are quite a few roles with very specific permissions in Drupal, because there are many employees that need to have access to modify different parts of the site:

- Administrator - Developers.
- Webmaster - A few people in UCC.
- Section editor - About 50 editors placed in different parts of the country. Section editors only have access to some parts (sections) of the site and to own content - see 'Logic for Section editor'. They often have some extra roles as well.
- Extra permissions for the editors: Editor, Kommunikation, Superbruger teasere, AMU, CFU, VU-admin, Kursusredaktør, HR, Forlag, Webform.



## Logic for Section editor

This is the logic implemented for the content types Basic page, Teaching material and Webform:

1. Section editors gets access according to Permissions first of all, so a setting in Permissions is capable of overruling or disabling the logic. Section editors should have permissions to edit own pages, but they should not have permissions to edit all pages, as it will disable the logic.
2. If a section editor has been granted rights to a section in his profile (field 'User sections'), he will have access to edit all pages with this section.
3. If a page doesn't have a section, but is a child (or child of a child...) of a parent page with a section, then a section editor - with permissions to this section - has access to edit the child page. Parent page is determined by parent menu item.
4. If none of the above checks succeeded, we deny access.

Current solution is implemented in custom module `ucc_site` in `hook_form_alter()`.

The sections are controlled by the page in the first level of the menu tree. When displaying a page, the preprocess functions are looking at the top level page for a value in `field_section` which is used for displaying the correct sidebar.

With this logic, and given that permissions allow it, Section editor can:

- Create and edit nodes in his/her sections (i.e. editor's sections are selected in his/her profile).
- Add a parent menu item to nodes within his/her sections.
- Edit and delete own nodes (i.e. the editor is the author of the node).

And Section editor can not:

- Delete nodes that is not his/her own (i.e. author is someone else).
- Add a top level menu item (main menu, secondary menu and education menu) as parent menu item to a node.
- Add a parent menu, that is not within his/her own sections, to a node.
- Edit parent menu item for nodes that is his/her own, but is not within his/her sections.

Logic for other content types:

- Section frontpage: Section editors can edit Section frontpage node that is inside his/her sections, but he/she doesn't have access to edit parent menu item for this content type.
- Teaser content types and Layout sidebar: Section editor can only edit and delete

nodes that is his/her own (i.e. the user is the author of the node).

## Theming

The theme ucc is a subtheme of Adaptive Themes.

## Coding standards and documentation

The development and maintenance of the site has followed Drupal coding standards - <https://www.drupal.org/coding-standards>.

Furthermore the code is inline documented (comments to the individual parts of a function) with a focus on describing why the code looks like it does and what its purpose is and not what it does.

## Logging

### Server Logging

Syslog-ng is installed on all servers. Syslog-ng is an open source implementation of the syslog protocol for \*nix systems. All logs are being stored on a dedicated logserver in order to prevail old logs from several month back.

In order to create the best working- and debugging environment for developers, all logs are live streamed from all servers directly to a dedicated Graylog server. Graylog is a fully integrated open source log management platform for collecting, indexing, and analyzing both structured and unstructured data. This way all developers will have instant access to all errors and furthermore developers are able to get alerts on all sorts of messages and error codes.

### Logging in Drupal

Basically all logging information is stored in `ucc.dk/admin/reports/dblog` and in Graylog.

UCC specific logging:

1. Import logging - import start time, end time, number of nodes updated after import.
2. Sis sign up logging - date and time of sis signup request and fails.
3. peyzt mail - logging of signups and error occurrence.

### Logging in UCC Services Bridge

The services bridge has two separate logging systems:

1. General application logging system.
2. Import specific logging system.

The general logging system is based on monolog which keeps a local log files of all events. Any event, that classified as an error, is also sent to a remote event tracking system called sentry. Sentry will notify the developers by email when errors occur.

The import specific logging system is available within the bridge application. Whenever an import is triggered, either automatically or manually, the parameters for the import is stored along with any errors that might have occurred.

Any error encountered during an import is also logged in the general logging system and will trigger notifications in sentry. Additionally the system will also send notifications by email to a number of preconfigured addresses.

Below is a screenshot of the import log:

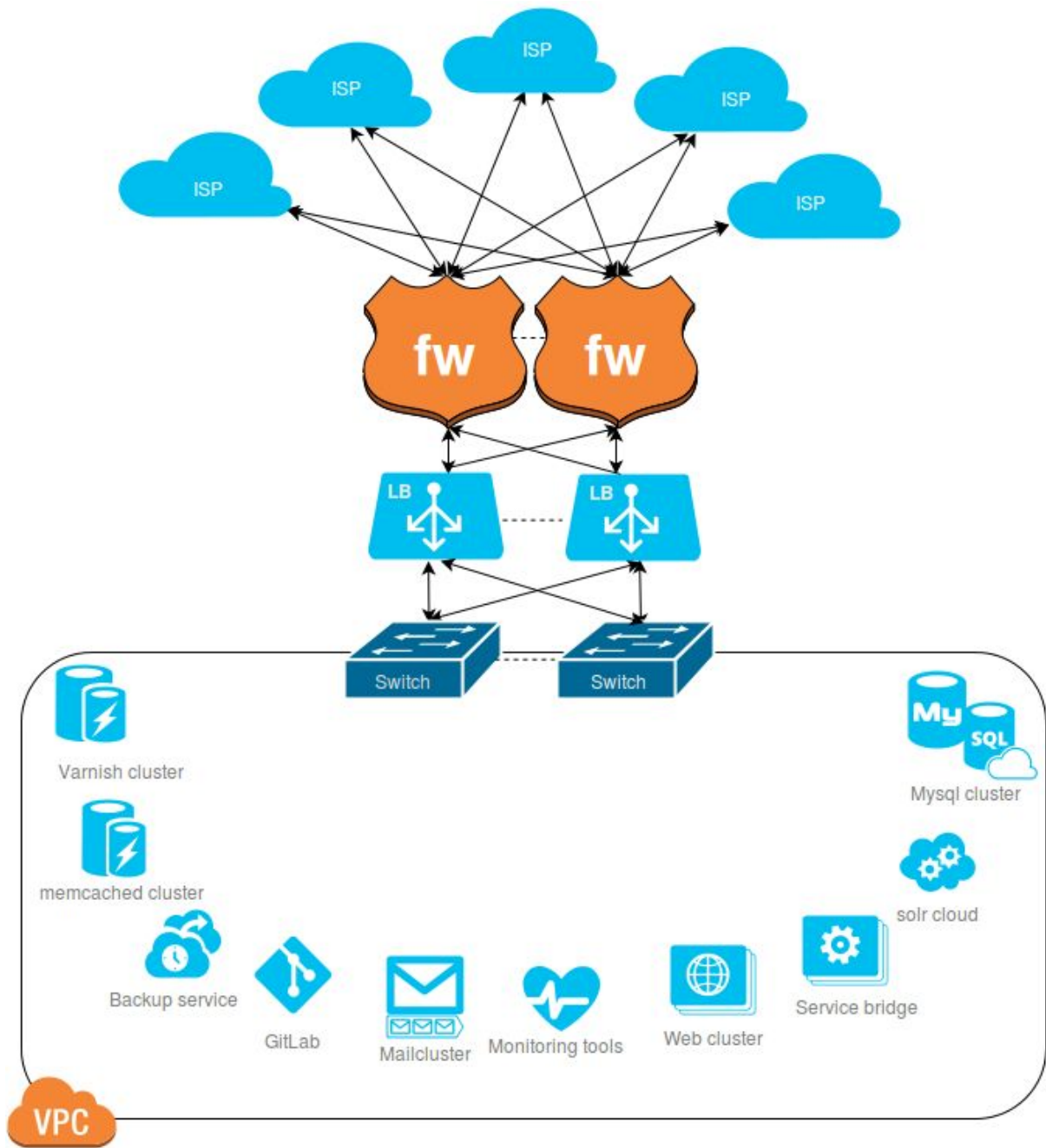
Status	Import Id	Started at	Imported at	Filename	Context	Total imported	Error
✓	<a href="#">27619</a>	October 27, 2015 12:05	October 27, 2015 12:05	<a href="#">20151027120502-cube.xml</a>	CRON	83	
✓	<a href="#">27616</a>	October 27, 2015 10:05	October 27, 2015 10:05	<a href="#">20151027100502-cube.xml</a>	CRON	83	
✓	<a href="#">27589</a>	October 26, 2015 16:05	October 26, 2015 16:05	<a href="#">20151026160502-cube.xml</a>	CRON	86	
✓	<a href="#">27585</a>	October 26, 2015 14:05	October 26, 2015 14:05	<a href="#">20151026140502-cube.xml</a>	CRON	86	
✓	<a href="#">27581</a>	October 26, 2015 12:05	October 26, 2015 12:05	<a href="#">20151026120502-cube.xml</a>	CRON	86	
✓	<a href="#">27578</a>	October 26, 2015 10:05	October 26, 2015 10:05	<a href="#">20151026100502-cube.xml</a>	CRON	86	
✓	<a href="#">27551</a>	October 25, 2015 16:05	October 25, 2015 16:05	<a href="#">20151025160501-cube.xml</a>	CRON	89	
✓	<a href="#">27547</a>	October 25, 2015 14:05	October 25, 2015 14:05	<a href="#">20151025140502-cube.xml</a>	CRON	89	
✓	<a href="#">27543</a>	October 25, 2015 12:05	October 25, 2015 12:05	<a href="#">20151025120502-cube.xml</a>	CRON	89	
✓	<a href="#">27540</a>	October 25, 2015 10:05	October 25, 2015 10:05	<a href="#">20151025100502-cube.xml</a>	CRON	89	

Here is an example from a failed import:

Provider	Occurred At	Preceding entries	Message
Sis	2015-09-29 22:01:04	0	failed with: Could not connect to host

# Infrastructure

## Overview



## Servers

Webservers [Apache(2.4)]

Database master/slaves [Mysql(5.6-Percona)]

Enterprise search platform [SolrCloud (4.8)]

External backup service [PeytzBackup]

Database cache [*memcached (1.4)*]  
Web cache [*varnish (4.0)*]  
Logging (*syslog-ng and graylog*)  
Monitoring tools (*observium, newrelic, nagios etc.*)  
Mail servers (*postfix*)  
repository manager (*GitLab 7.12*)

## Hosting

ucc.dk is hosted in a Danish hosting facility in order to be as close to the end users as possible and thereby achieve the lowest possible page response time and give the highest possible protection of sensitive data.

The setup is owned and managed by Peytz & Co and is compliant with the ISO27001 standard for information security. Peytz & Co has created a virtual environment in order to ensure full redundancy and the possibility to scale quickly. Since Peytz & Co not only owns the virtual servers but also the physical hardware, Peytz & Co have full control over every layer of the setup and is therefor not shared like other cloud solutions.

In order for ucc to develop at test new features together with Peytz & Co a complete replicate of every production server is created in a staging environment for test purposes only. This is referred to as ucc.d7.stage.

## Service Bridge

The Service Bridge is hosted in AWS as a standalone setup and has the following dependencies:

- Varnish - <https://www.varnish-cache.org/>
- Apache2 - <https://httpd.apache.org/>
- Memcached - <http://memcached.org/>
- Mysql - <https://www.mysql.com/>
- Flint - <http://henrikbjorn.github.io/Flint/>
- Doctrine - <http://doctrine-project.org/>
- Symfony - <http://symfony.com>
- raven-php - <https://www.getsentry.com>
- PHPUnit - <http://phpunit.de>
- Bootstrap - <http://getbootstrap.com>
- jQuery - <http://jquery.com>

## Links

Live site: <https://ucc.dk>

Stage site: <http://ucc.d7.stage.ng.peytz.dk>

UCC Services Bridge: <http://ucc-services-bridge.aws.peytz.dk>

# Appendix 1

## list of modules

Package	Name	Version
Administration	Administration menu (admin_menu)	7.x-3.0-rc5
Administration 7.x-3.0-rc5	Administration menu Toolbar style (admin_menu_toolbar)	
Administration	Module filter (module_filter)	7.x-2.0
Chaos tool suite	Chaos tools (ctools)	7.x-1.9
Chaos tool suite 7.x-1.9	Custom content panes (ctools_custom_content)	
Chaos tool suite	Page manager (page_manager)	7.x-1.9
Chaos tool suite	Views content panes (views_content)	7.x-1.9
Date/Time 7.x-2.8	Date (date)	
Date/Time	Date API (date_api)	7.x-2.8
Date/Time	Date Popup (date_popup)	7.x-2.8
Date/Time	Date Tools (date_tools)	7.x-2.8
Date/Time	Date Views (date_views)	7.x-2.8
Development	Devel (devel)	7.x-1.5
Features	Features (features)	7.x-2.6
Fields	Entity Reference (entityreference)	7.x-1.1
Fields	Field collection (field_collection)	7.x-1.0-beta8
Fields	Field Group (field_group)	7.x-1.4
Fields	Link (link)	7.x-1.3
Form Elements	Hierarchical Select (hierarchical_select)	Dev version
Mail 7.x-2.34	Mail System (mailsystem)	
Mail 7.x-1.0-beta4	Mime Mail (mimemail)	
Mail 7.x-1.0-beta4	Mime Mail CSS Compressor (mimemail_compress)	
Media 7.x-2.0-beta2	File Entity (file_entity)	
Media	File Lock (file_lock)	7.x-2.0
Media	Media (media)	7.x-2.0-alpha4
Media	Media Field (mediafield)	7.x-2.0-alpha4
Media 7.x-2.0-alpha4	Media Internet Sources (media_internet)	
Media 7.x-2.0-alpha4	Media WYSIWYG (media_wysiwyg)	
Media	Media: 23video (media_23video)	
Media	Media: Vimeo (media_vimeo)	7.x-2.1

Media	Media: YouTube (media_youtube)	7.x-3.0
Multilanguage	Administration language (admin_language)	
7.x-1.0-beta1		
Multilingual	Localization update (l10n_update)	7.x-2.0
Multilingual - Internationalization	Field translation (i18n_field)	7.x-1.13
Multilingual - Internationalization	Internationalization (i18n)	7.x-1.13
Multilingual - Internationalization	Multilingual content (i18n_node)	7.x-1.13
Multilingual - Internationalization	String translation (i18n_string)	7.x-1.13
oEmbed	oEmbed (oembed)	7.x-1.0-rc2
Other	Entity API (entity)	7.x-1.6
Other	Entity Connect (entityconnect)	7.x-1.0-rc1
Other	Entity tokens (entity_token)	7.x-1.6
Other	Entity view modes (entity_view_mode)	7.x-1.0-rc1
Other	Google Content Experiments (content_experiments)	7.x-1.4
Other	ImageAPI Optimize (imageapi_optimize)	7.x-1.2
Other	ImageMagick (imagemagick)	7.x-1.0
Other	ImageMagick Advanced (imagemagick_advanced)	7.x-1.0
Other	Job Scheduler (job_scheduler)	
7.x-2.0-alpha3		
Other	Jquery Colorpicker (jquery_colorpicker)	7.x-1.1
Other	Libraries (libraries)	7.x-2.2
Other	Menu Block (menu_block)	7.x-2.7
Other	Menu Block Export (menu_block_export)	7.x-2.7
Other	Menu position (menu_position)	7.x-1.1
Other	Node clone (clone)	7.x-1.0-rc2
Other	Node clone tab (node_clone_tab)	7.x-1.1
Other	Node Edit Protection (node_edit_protection)	7.x-1.1
Other	Page Preview (pagepreview)	7.x-1.x-dev
Other	Pathauto (pathauto)	7.x-1.2
Other	Quicktabs (quicktabs)	7.x-3.6
Other	Rabbit Hole (rabbit_hole)	7.x-2.23
Other	Rabbit Hole nodes (rh_node)	7.x-2.23
Other	Redirect (redirect)	7.x-1.0-rc3
Other	Scheduler (scheduler)	7.x-1.3
Other	Search API Solr Overrides (search_api_solr_overrides)	
7.x-1.0-rc1	Other	Strongarm (strongarm)
	7.x-2.0	
Other	Token (token)	7.x-1.6
Other	Token tweaks (token_tweaks)	dev version
Other	Transliteration (transliteration)	7.x-3.2
Panels	Mini panels (panels_mini)	7.x-3.5
Panels	Panels (panels)	7.x-3.5
Panels	Panels In-Place Editor (panels_ipe)	7.x-3.5
Path management	Global Redirect (globalredirect)	7.x-1.5
Performance and scalability	Cache Expiration (expire)	7.x-2.0-rc4
Performance and scalability	Entity modified (entity_modified)	7.x-1.2



Performance and scalability	Memcache (memcache)	
7.x-1.5		
Peytz & Co	Peytz Mail (peytz_mail)	1.0
Peytz & Co	Peytz Mail Subscribe (peytz_mail_subscribe)	1.0
Peytz & Co.	Oprop API (oprop_api)	
7.x-1.0-alpha2		
Peytz & Co.	Oprop Webform (oprop_webform)	
Search	Search API (search_api)	7.x-1.15
Search	Search facets (search_api_facetapi)	7.x-1.15
Search	Search views (search_api_views)	7.x-1.15
Search Toolkit	Facet API (facetapi)	7.x-1.5
Search Toolkit	Facet API Bonus (facetapi_bonus)	7.x-1.1
Search Toolkit	Facetapi Collapsible Links (facetapi_collapsible)	7.x-1.1
SEO	Metatag (metatag)	7.x-1.7
SEO	Metatag: Facebook (metatag_facebook)	7.x-1.7
SEO	Metatag:OpenGraph (metatag_opengraph)	
7.x-1.7		
Statistics	Google Analytics (googleanalytics)	7.x-2.1
UCC	UCC editor properties (ucc_editor_properties)	7.x-0.1
UCC	UCC global content fields (ucc_global_content_fields)	7.x-0.1
UCC	UCC Hierarchical Select Menu (ucc_hs_menu)	
UCC	UCC search base (ucc_search_base)	7.x-0.1
UCC	UCC site custom. (ucc_site)	
User interface	CKEditor (ckeditor)	7.x-1.16
User interface	fancyBox (fancybox)	
7.x-2.0-beta6		
User interface	jQuery Update (jquery_update)	7.x-2.6
User interface	Wysiwyg Template Plugin (wysiwyg_template)	
7.x-2.11		
Variable	Variable (variable)	7.x-2.5
Views	Views (views)	7.x-3.11
Views	Views Bulk Operations (views_bulk_operations)	
7.x-3.3		
Views	Views JSON (views_json)	
7.x-1.0-alpha2		
Views	Views PHP (views_php)	
7.x-1.0-alpha1		
Views	Views UI (views_ui)	7.x-3.11
Webform	Webform (webform)	7.x-3.24
Webform	Webform Validation (webform_validation)	7.x-1.10