

Linked Data Publishing with Drupal

Joachim Neubert

ZBW – German National Library of Economics

Leibniz Information Centre for Economics

SWIB13 Workshop

Hamburg, Germany

25.11.2013

My background

- Scientific software developer at ZBW – German National Library for Economics, mainly concerned with Linked Open Data and knowledge organization systems and services
- Published 20th Century Press Archives in 2010, with some 100,000 digitized newspaper articles in dossiers (<http://zbw.eu/beta/p20>, custom application written in Perl)
- Published a repository of ZBW Labs projects recently – basically project descriptions and a blog (<http://zbw.eu/labs>, Drupal based)

Workshop Agenda – Part 1

1) Drupal 7 as a Content Management System: Linked Data by Default

Hands-on: Get familiar with Drupal and its default RDF mappings

2) Using Drupal 7 as a *Framework* for Content Management

Hands-on: Create a custom content type and map it to RDF

Workshop Agenda – Part 2

- ❖ Produce other RDF Serialization Formats: RDF/XML, Turtle, Ntriples, JSON-LD
- ❖ Create a SPARQL Endpoint from your Drupal Site
- ❖ Cool URIs
- ❖ Create Out-Links with Web Taxonomy
- ❖ Current limitations of RDF/LD support in Drupal 7
- ❖ Outlook on Drupal 8

Drupal as a CMS (Content Management System) ready for RDF and Linked Data

Why at all linked data enhanced publishing?

- Differentiate the subjects of your web pages and their attributes
- Thus, foster data reuse in 3rd party services and applications
 - Mashups
 - Search engines
- Create meaningful links, adding value for users

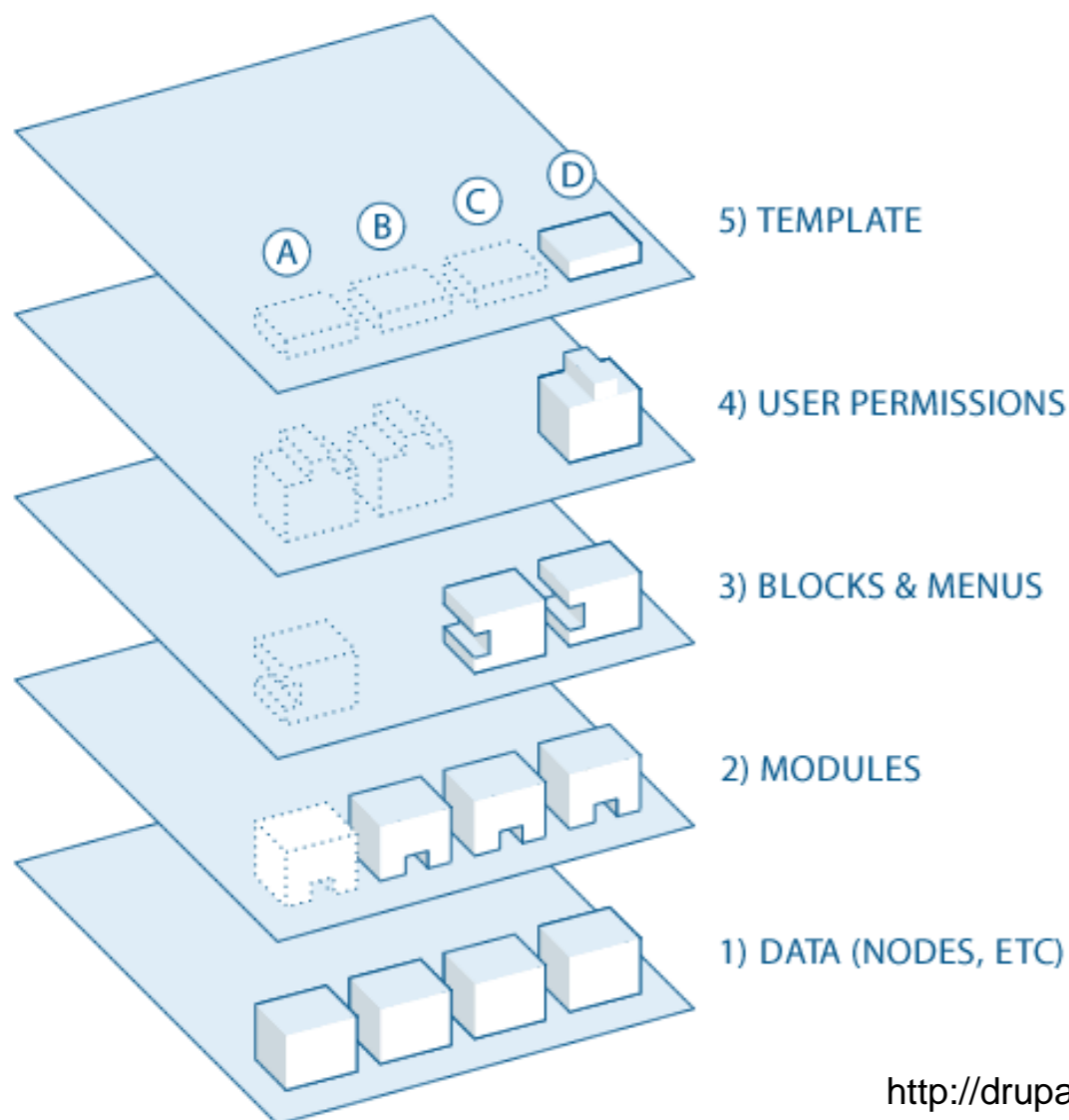
Why use a content management system?

- Standard tasks (browser compatibility, page templates, responsive css, site navigation, search, form handling, calendar, wysiwyg, revisions, translations, permissions, data management , security) made easy
- Easy-to-add web 2.0 features (blogging and comments, tags, rating, forums, ...)
- Know-how available outside a single development team

Why Drupal?

- More than 1 million sites worldwide
- 2 % of the web
- Large institutional sites (whitehouse.gov, amnesty.org, economist.com, examiner.com, louvre.fr)

Why Drupal?



- Open & modular architecture
- Extensible by modules
- Standards-based
- Scalable
- Vibrant open source community, and commercial services, too

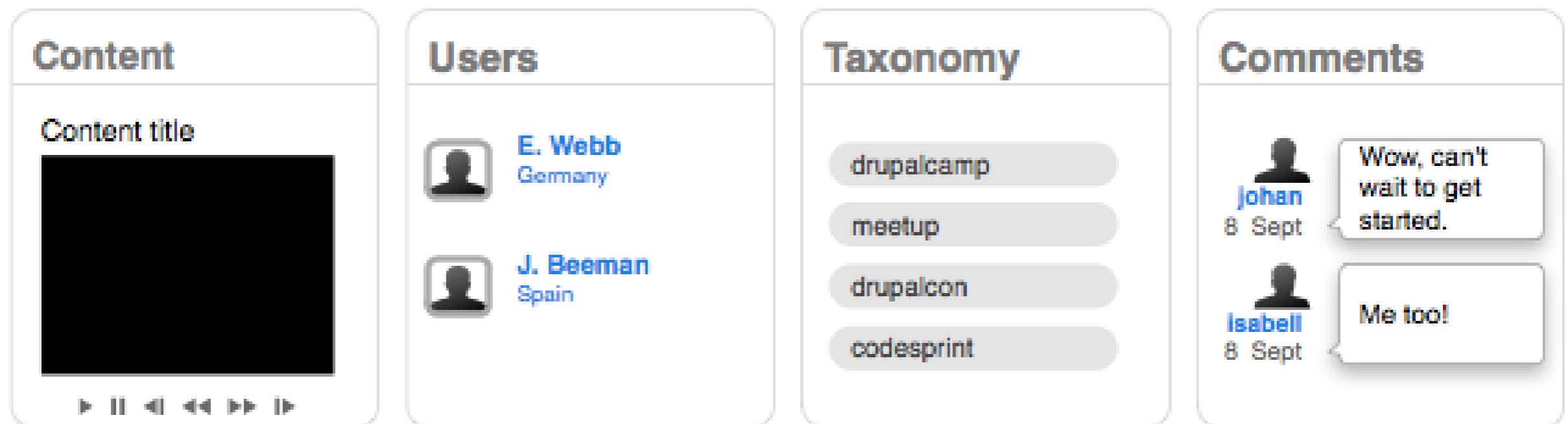
<http://drupal.org/getting-started/before/overview>

<http://de.slideshare.net/scorlosquet/drupal-as-a-semantic-web-platform>

The Drupal Community

- More than 30,000 developers
 - More than 5,000 contributed modules (Drupal 7, actively maintained)
 - Activities organized through issue queues
 - Community contributed documentation
 - Thematic discussion groups:
 - <https://groups.drupal.org/semantic-web>
 - <https://groups.drupal.org/libraries>
 - Planet Drupal (aggregated blog entries): <https://drupal.org/planet>
-

Drupal entity types



**“Node” in
Drupal jargon**

From <https://drupal.org/node/1261744>

Nodes: Drupal's basic structure for content

- Title
- Author & created/modified date
- Body
- May have tags (taxonomy), and/or comments, and/or images
- Additional features:
 - Revisions, Diffs
 - Translations

RDF mappings

- Content types (or bundles of other entity types) are mapped to RDF classes
- Fields are mapped to RDF properties
- Defined on a the data layer (independent of the output system)
- Drupal takes care of inserting it into the chosen HTML layout as RDFa attributes:

```
<div typeof="foaf:Person">  
  <h2 property="foaf:name">Jane Doe</h2>  
</div>
```

ZBW Labs as Linked Open Data

2013-04-24 by [Joachim Neubert](#)

As a laboratory for new, Linked Open Data based publishing technologies, we now develop the ZBW Labs web site as a Semantic Web Application. The pages are enriched with RDFa, making use of Dublin Core, DOAP (Description of a Project) and other vocabularies. The schema.org vocabulary, which is also applied through RDFa, should support search engine visibility.

With this new version we aim at a playground to test new possibilities in electronic publishing and linking data on the web. At the same time, it facilitates editorial contributions from project members about recent developments and allows comments and other forms of participation by web users.

As it is based on Drupal 7, RDFa is "build-in" (in the CMS core) and is easy done by configuration on a field level. Enhancements are made through the RDFx, Schema.org and SPARQL Views modules. A lot of other ready-made components in Drupal (most noteworthy the Views and the new Entity Reference modules) make it easy to provide and interlink the data items on the site. The current version of Zen theme enables the HTML 5 and the use of RDFa 1.1, and permits a responsive design for smartphones and pads.

[ZBW Labs Website](#)

[Linked Open Data](#) [Drupal](#) [Publishing Technologies](#)

[Log in or register to post comments](#)



Search

Username *

Password *

[Create new account](#)

[Request new password](#)

Log in

Output in RDFa

- Drupal renders RDF mappings as HTML attributes

```
<span property="schema:name doap:name dc:title">ZBW Labs Website</span>
```

- No frickling in HTML producing code or templates
- Works out of the box for different Drupal themes (screen designs)
- In Drupal 7, by default XHTML/RDFa 1.0
- Themes for HTML5/RDFa 1.1 available (e.g., Zen)

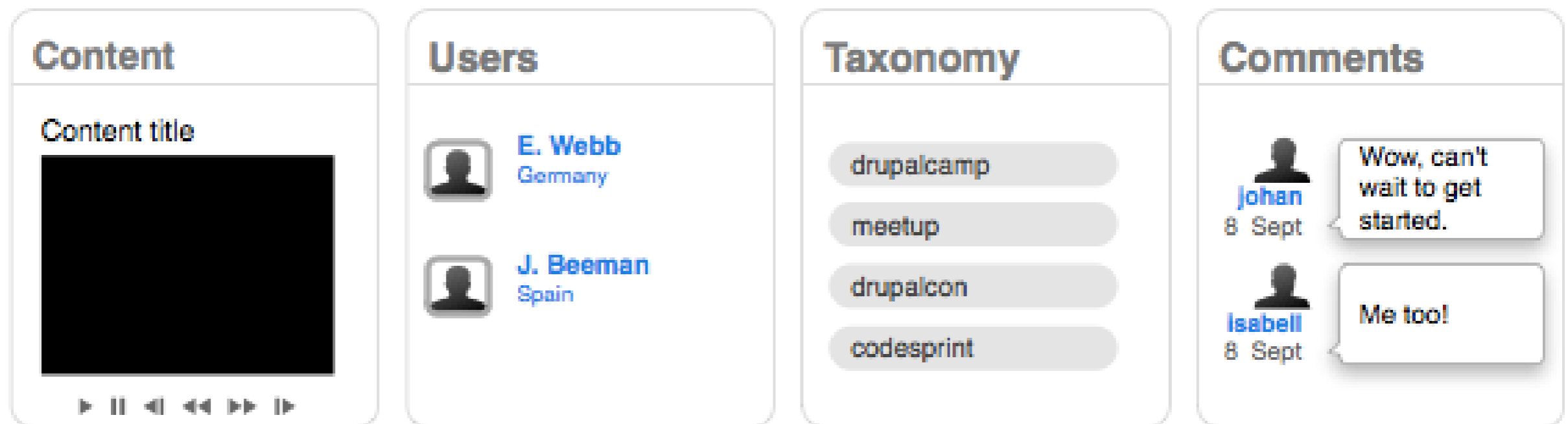
Hands-on, part 1: Create Articles

Workshop pad:

<http://etherpad.lobid.org/p/swib13-drupal-ws>

Drupal 7 as a Content Management Framework

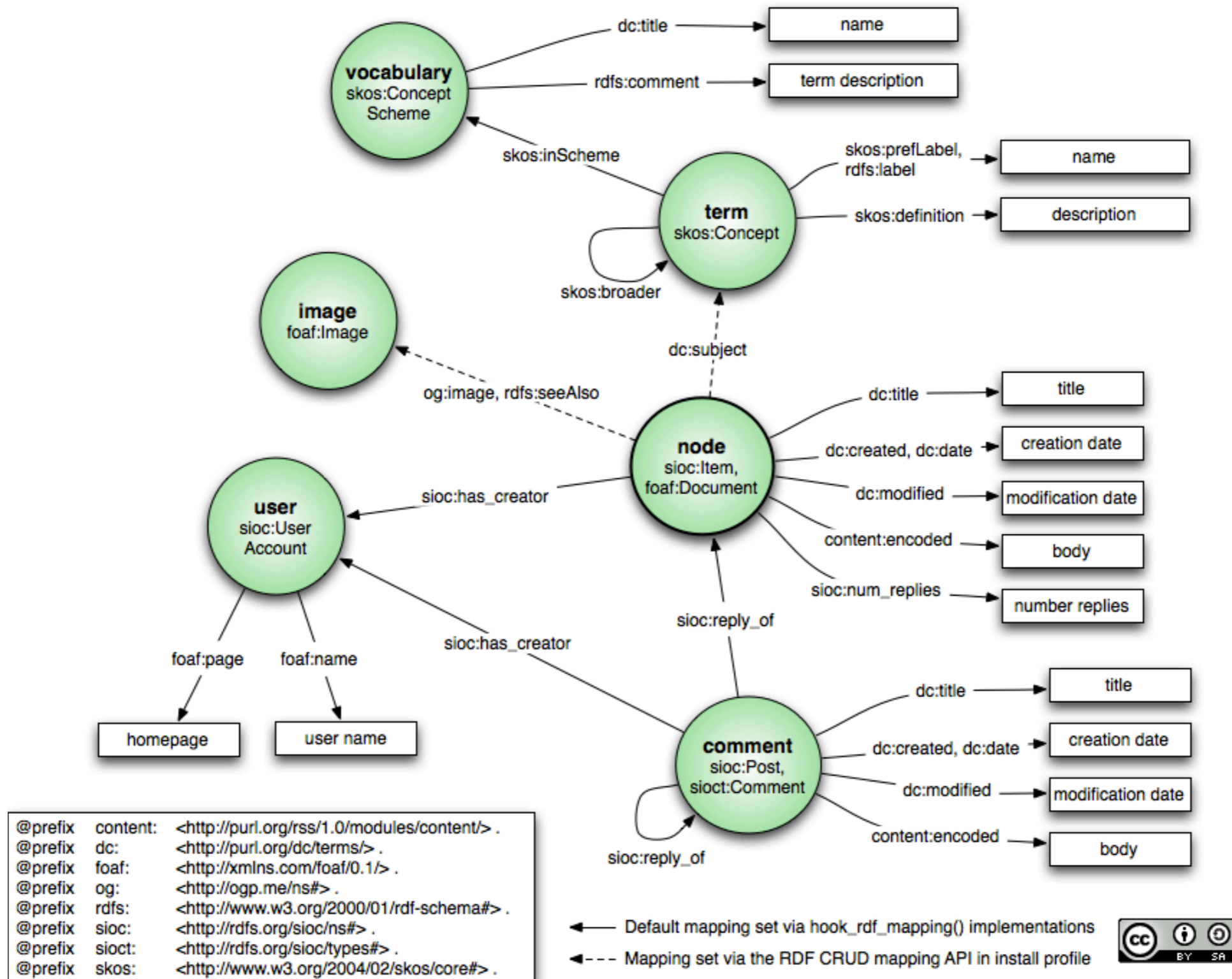
Drupal entity types



“Node” (in
Drupal jargon)

From <https://drupal.org/node/1261744>

Drupal 7 default RDF schema



Drupal fields

- Fields can be defined in Drupal for custom data
- *Drupal fields are different from what we know as database fields*
- Fields are attached to entities
- Single or multiple occurrence
- Various technical field types (text, integer, file, ...)
- Custom modules can add their own field types
- Some field types are supported by input widgets (such as a pop-up calendar for dates)

Bundles allow for sub-types of an entity

- “Bundles” refer to Drupal fields
- “Content type” means a bundle for the node entity type
- Predefined content types are:
 - Basic page (just title and body – for static content, such as an “About” page)
 - Article (with tags and an image – for blog articles, news, ...)
- Custom modules can add their own content types, or even entity types

Hands-on, part 2: Create a “project” content type

Hands-on, part 2: Preparations

1. Enable Modules

- RDF UI
- Date and Date Popup (with dependencies)
- Entity Reference

2. Add RDF Namespaces (Configuration > RDF publishing settings > RDF namespaces Tab)

- schema <http://schema.org/>
- doap <http://usefulinc.com/ns/doap#>

3. Create a “Categories” taxonomy

Hands-on, part 2: Fields and mappings

Type:

Project (doap:Project, schema:CreativeWork)

Fields:

Name (doap:name, schema:name, dc:title): (title)

Short Description (doap:shortdesc, schema:summary): Long text

Started (doap:created, schema:startDate): Date

Lead (doap:maintainer, schema:accountablePerson): Entity reference

Article (rdfs:seeAlso): Entity reference (Blog Article)

Category (doap:category, schema:about, dc:subject): Term reference

Hands-on, part 2: Add example content

What did we achieve?

- Learned about Drupal RDF output produced by default
- Created a custom content type and attached fields
- Added arbitrary RDF classes and properties
- Learned how to interlink content and other entities

Extending Drupal even further

As powerful a **Content Management Framework**, Drupal provides

- Well defined APIs (database abstraction layer, Field API, Form API, Entity API, ...)
 - In particular, RDF Mapping API allows create the mappings programmatically, which we be built through the User Interface
 - Entity API allows building custom entities with arbitrary properties
 - ... even residing in remote databases
- requires substantial programming skills

Coffee break



Selected Linked Data-related Drupal stuff

- What I introduced up to now, is quit well production ready (and working on hundred thousands of sites, as RDF is enabled by default). Same is true for the field subsystem, for entity references, etc.
- However, much less Drupal sites deliberately work with RDF, and the module I now will introduce are often in beta or even early alpha state

Produce other RDF serialization formats

- Serialize Drupal RDF data with rdfx and restws modules in
 - RDF/XML
 - Turtle
 - N-Triples
- Add JSON-LD module

* currently does not work with PostgreSQL – for a workarround, see <http://drupal.org/node/1999754#comment-7438562>

Hands-on: Serialization formats

Providing a SPARQL endpoint for your site

- SPARQL is – like SQL – a general purpose query language for RDF data
- Let you select data from all over your site in flexible and unforeseen ways
- Let even combine you data from your site and from others in a single query

Hands-on: SPARQL

SPARQL: Some restrictions

However, somebody who wants to query the store, has to know, or to figure out somehow,

- that there are articles and projects, which are interlinked by `rdfs:seeAlso`
- in which direction of the `rdfs:seeAlso` connection was created
- that projects actually have a `schema:about` property

SPARQL: Production-ready?

- An open SPARQL endpoint on a production server is like an open SQL interface: performance and security issues
 - A much finer tunable module combination for SPARQL queries was announced on the Semantic Web Drupal group:
 - Limits for processing time and number of results
 - Selected indexing, batches of index jobs
 - see really good step-by-step description at <https://drupal.org/node/2028111>
 - Even then, for the regular pages your users should get on your site, you should better use the Drupal Views module.
-

Cool URIs for Linked Data

- **Cool URIs don't change**

Tim Berners-Lee, <http://www.w3.org/Provider/Style/URI>

- No technology-dependent parts
- More on Linked-Data-URIs: Chapter 2 of Bizer/Heath: Linked Data (2011) <http://linkeddatabook.com/editions/1.0/>

Cool URIs for Drupal require work

1) Drupal's out-of-the-box default URI

<http://drupal-lod/?q=node/25>

2) with the „Clean URLs“ feature enabled

This has already be done by the TurnKey Linux Drupal 7 installation

<http://drupal-lod/node/25>

3) with the (core) „Path“ module enabled and an alias defined

<http://drupal-lod/project/zbw-labs>

Also good for seach engines! For more on Drupal support for redirects: check Global Redirect module (<https://drupal.org/project/globalredirect>)

Hands-on: Cool URIs

Even more work if ...

1) You have a multilingual site, and want to have only one URI for a project,

e.g., instead of

<http://zbw.eu/labs/de/project/zbw-labs>

<http://zbw.eu/labs/en/project/zbw-labs>

just

<http://zbw.eu/labs/project/zbw-labs>

2) Or you want, „by the rules“, to distinguish between the thing itself (the project) and the web page (about the project)

Linked Data URIs „by the textbook“

No httpRange-14 conformant standard solution in Drupal known to me
– yet workarounds:

- Add code to set the „about“ attribute to the generic (language-agnostic) URI of the entity (for multilingual sites)
(code example in <http://groups.drupal.org/node/247058#comment-798823>)
- Or add code to set the “about” attribute to the page URI extended with a fragment identifier (e.g., “#resource” – so called “hash URI”)

Web Taxonomy: Using vocabularies from the web

- Autocomplete widget for Drupal fields, powered by vocabularies maintained elsewhere
- Local taxonomy works as a proxy for the web taxonomy terms in use

Web Taxonomy: Using vocabularies from the web

- Prerequisites:
 - a already built plugin for the vocabulary (experimental examples for VIAF, STW, Dbpedia, Agrovoc, GND economists)

or

- a web-accessible autosuggest service which delivers terms and their URIs (may be JSON, SPARQL results or even SOAP)
- a custom coded plugin to access the service

Plugin example: Economics Taxonomies

- Third party thesauri, such as STW Thesaurus for Economics, can be re-used for indexing a collection

Subjects (suggested from STW descriptors web taxonomy)

Account
Bookkeeping
Commercial professions
Balance sheet date
Accounting information system
Accounting law
Accounting obligations
Accounting policy
Financial ratio
Accounting standards

nk

Code downloadable and installable from <https://drupal.org/sandbox/jneubert/1447918>

Hands-on: Web taxonomy

Current limitations of RDF/LD support in Drupal 7



Nested RDF structures only with custom code

Workaround example:

Git repository URI in DOAP ontology demands a separate node, e.g.,

```
<> a schema:CreativeWork, doap:Project;
    doap:repository [ a doap:GitRepository;
                     doap:location "http://github.com/some/id.git" ];
```

- Create field_gitrepository and map to doap:location
- Create a custom template file for the field (field--field_gitrepository--lproject.tpl.php)

```
<div rel="doap:repository" class="field-items"<?php print $content_attributes; ?>>
  <div about="[:repos]" typeof="doap:GitRepository">
    <?php foreach ($items as $delta => $item): ?>
      <div class="field-item"<?php print $item_attributes[$delta]; ?>><?php
print render($item); ?></div>
    <?php endforeach; ?>
  </div>
</div>
</div>
```

Further limitations in Drupal 7

- RDFa support currently works for single entities – pages with multiple entities (search results, term pages, views, etc.) are not supported
- RDFa output may break under certain special conditions
(<http://drupal.org/node/1778226>)

Outlook to Drupal 8

- Drupal base functionalities refactored for using Symfony framework
- Aimed at an extended service-oriented architecture – design goal:
“Each piece of content gets its own URL”
(<http://www.unleashedmind.com/en/blog/sun/drupal-8-the-path-forward>)
- Web services based on HAL (Hypertext application language)
- Configuration in files (version control, portability)
- Multilingual unified and improved

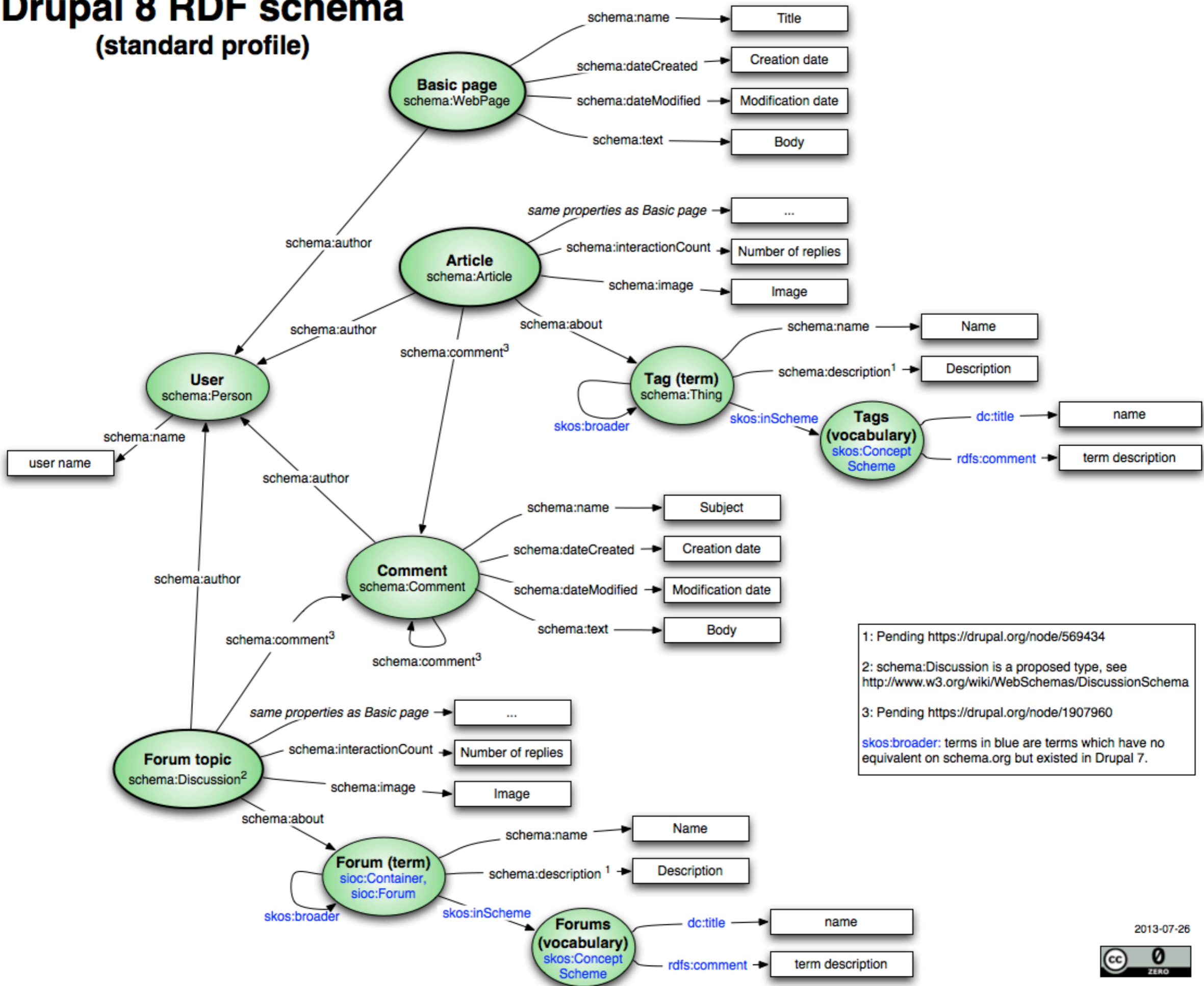
Outlook to Drupal 8 (cont.)

- Date, Views, Entity Reference, ... modules in core
- Wysiwyg and in-place editing out of the box

RDF

- Solely use RDFa 1.1 lite (@property, determined by attribute placement, instead of @rel/@rev)
<https://drupal.org/node/1780090>
- Default mappings based almost completely on schema.org
(see <https://drupal.org/node/1784234>)

Drupal 8 RDF schema (standard profile)



2013-07-26



To sum up

- Linked data publishing via a CMS, in particular Drupal, is a valid option
- If your data can be mapped to an essentially flat RDF data structure, linked data can be added mostly by site builders, without much additional effort
- Sometimes research is required on how to solve problems, and at times glue code has to be written
- But: most of the code for your web application is already there, and is supported by a large and helpful Open Source community

Thank you!

Joachim Neubert

ZBW – Leibniz Information Centre for Economics

j.neubert@zbw.eu

<http://zbw.eu/labs>