

v2.helsinki.at
Django und Plone

Ernesto Rico-Schmidt Johannes Raggam

Grazer Linxstage 2011
April 2011

Inhalt

Einführung

Inhalt

Einführung

Django

Inhalt

Einführung

Django

Plone

Inhalt

Einführung

Django

Plone

Die Integration

Einführung

Die bisherige Lösung

- ▶ `http://helsinki.at`, seit Jänner 2001
- ▶ PHP/MySQL
- ▶ Kein Framework
- ▶ Kein Content Management System
- ▶ Erstes Web und PHP-Projekt
- ▶ HTML-Tabellen
- ▶ Zwei SQL-Tabellen
- ▶ 2005 Sendungen seit 2001

Einführung

Die Anforderungen

- ▶ Neues Design
- ▶ Content Management System
- ▶ Neue Programmverwaltung
- ▶ Direkte Verlinkung mit der CBA (Cultural Broadcasting Archiv, <http://cba.fro.at>)
- ▶ Einfache Möglichkeit Kommentare zu Sendungen abzugeben

Einführung

Die Optionen

- ▶ Django
- ▶ TurboGears
- ▶ Plone
- ▶ Ruby on Rails
- ▶ CakePHP, Symfony, Zend

Einführung

Die Optionen

- ▶ Django
- ▶ TurboGears
- ▶ Plone
- ▶ Ruby on Rails
- ▶ CakePHP, Symfony, Zend (nicht wirklich)

Einführung

Django

- ▶ <http://www.djangoproject.com>
- ▶ ursprünglich von Adrian Holovaty, Simon Wilson, Jacob Kaplan-Moss und Wilson Miner für World-Online in 2003 entwickelt.
- ▶ offen (BSD-Lizenz) seit 2005
- ▶ Version 1.0 in September 2008
- ▶ Version 1.1 in Juli 2009
- ▶ Version 1.2 in Mai 2010
- ▶ Version 1.3 in März 2011

Prominentes Beispiel: <http://disqus.com/>

“Serving 400 million people with Python”

Einführung

Django

- ▶ Model-View-Controller bzw. Model-Template-View
- ▶ Objektrelationales Mapping (ORM)
- ▶ Automatische Administrationsoberfläche
- ▶ Elegante URL-Konfiguration
- ▶ Templates
- ▶ Views (generische)
- ▶ Cache
- ▶ Internationalisierung

Einführung

Plone

- ▶ CMS System
- ▶ Programmiersprache: Python
- ▶ Applicationserver: Zope (Zope Object Publishing Environment)
- ▶ Datenbank: ZODB (Objektorientiert)
- ▶ Benutzerfreundlich (User), Umfangreich (Admins), Flexibel (Developer)

Die Programmverwaltung

Django-Verwaltung

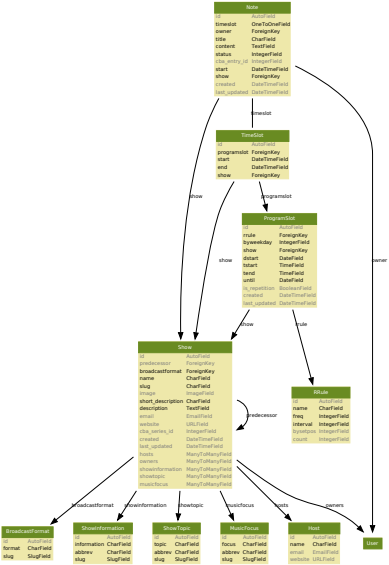
Willkommen, ers. Passwort ändern / Abmelden

Start > Program

Program-Verwaltung

Program		
Broadcast formats	 Hinzufügen	 Ändern
Hosts	 Hinzufügen	 Ändern
Music focus	 Hinzufügen	 Ändern
Notes	 Hinzufügen	 Ändern
Program slots	 Hinzufügen	 Ändern
Show information	 Hinzufügen	 Ändern
Show topics	 Hinzufügen	 Ändern
Shows	 Hinzufügen	 Ändern

Die Programmverwaltung



Die Programmverwaltung

Manager

```
# program/models.py
class TimeSlotManager(models.Manager):
    def get_or_create_current(self):
        try:
            return TimeSlot.objects.get(
                start__lte=datetime.now(),
                end__gt=datetime.now())
        except ObjectDoesNotExist:
            once = RRule.objects.get(pk=1)
            today = date.today().weekday()
            default = Show.objects.get(pk=1)
            previous = TimeSlot.objects.filter(
                end__lte=datetime.now()).order_by('-start')[0]
            next = TimeSlot.objects.filter(
                start__gte=datetime.now())[0]
```

```
# program/models.py
    dstart = previous.end.date()
    tstart = previous.end.time()
    until = next.start.date()
    tend = next.start.time()
    new_programslot = ProgramSlot(
        rrule=once,
        byweekday=today,
        show=default,
        dstart=dstart,
        tstart=tstart,
        tend=tend,
        until=until)
    new_programslot.save()
    return new_programslot.timeslots.all()[0]
```


Die Programmverwaltung

Administration

```
# program/admin.py
class NoteAdmin(admin.ModelAdmin):
    def queryset(self, request):
        qs = super(NoteAdmin, self).queryset(request)
        if request.user.is_superuser:
            return qs
        else:
            return qs.filter(owner=request.user)
    def save_model(self, request, obj, form, change):
        obj.owner = request.user
        obj.save()
```

Die Programmverwaltung

Administration

```
# program/admin.py
def formfield_for_foreignkey(
    self, db_field, request, **kwargs):
    if db_field.name == 'timeslot':
        if request.user.is_superuser:
            kwargs['queryset'] = \
                TimeSlot.objects.filter(start__gt=datetime.now)
        else:
            shows = request.user.shows.all()
            kwargs['queryset'] = \
                TimeSlot.objects.filter(
                    show__in=shows, start__gt=datetime.now)
    return \
        super(NoteAdmin, self).formfield_for_foreignkey(
            db_field, request, **kwargs)
```

Die Integration

- ▶ Solr: <http://lucene.apache.org/solr/>
- ▶ Haystack: <http://haystacksearch.org/>
- ▶ LDAP: <http://packages.python.org/django-auth-ldap/>

Die Solr-Integration

```
# settings.py
INSTALLED_APPS = (
    ..
    'haystack',
    ..
)

HAYSTACK_SITECONF = 'helsinki.search_sites'
HAYSTACK_SEARCH_ENGINE = 'solr'
HAYSTACK_SOLR_URL = 'http://localhost:8983/solr'
```

Die Solr-Integration

```
# program/search_indexes.py
from haystack.indexes import CharField, DateTimeField, \
    SearchIndex
from haystack import site
from datetime import datetime
from program.models import Show

class ShowIndex(SearchIndex):
    text = CharField(document=True, use_template=True)
    last_updated = DateTimeField(model_attr='last_updated')

    def get_queryset(self):
        return Show.objects.filter(
            last_updated__lte=datetime.now())

site.register(Show, ShowIndex)
```

Die Solr-Integration

```
{# templates/search/indexes/program/show_text.txt #}  
{{ object.name }}  
{{ object.description }}  
{{ object.short_description }}
```

```
$ ./manage.py build_solr_schema > schema.xml  
$ ./manage.py rebuild_index
```

Die LDAP-Integration

```
# settings.py
import ldap
from django_auth_ldap.config import LDAPSearch, \
    PosixGroupType
AUTH_LDAP_SERVER_URI = "ldap://ldap.helsinki.at"
AUTH_LDAP_BIND_DN = "cn=reader,dc=helsinki,dc=at"
AUTH_LDAP_BIND_PASSWORD = ".."
AUTH_LDAP_USER_DN_TEMPLATE = \
    "uid=%(user)s,ou=people,dc=helsinki,dc=at"
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "ou=groups,dc=helsinki,dc=at",
    ldap.SCOPE_SUBTREE,
    "(objectClass=posixGroup)"
)
AUTH_LDAP_GROUP_TYPE = PosixGroupType()
```

Die LDAP-Integration

```
# settings.py
AUTHENTICATION_BACKENDS = (
    'django_auth_ldap.backend.LDAPBackend',
)
AUTH_LDAP_USER_ATTR_MAP = {
    "first_name": "givenName",
    "last_name": "sn",
    "email": "mail"
}
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    "is_staff": "cn=pvstaff,ou=groups,dc=helsinki,dc=at",
    "is_superuser": "cn=Technik,ou=groups,dc=helsinki,dc=at"
}
AUTH_LDAP_ALWAYS_UPDATE_USER = True
AUTH_LDAP_FIND_GROUP_PERMS = True
AUTH_LDAP_MIRROR_GROUPS = True
```




Plone is among the top 2% of all open source projects worldwide, with 340 core developers and more than 300 solution providers in 57 countries. The project has been actively developed since 2001, is available in more than 40 languages, and has the best security track record of any major CMS. It is owned by the Plone Foundation, a 501(c)(3) not-for-profit organization, and is available for all major operating systems.

Sources: CVE and Ohloh.

Quelle: plone.org

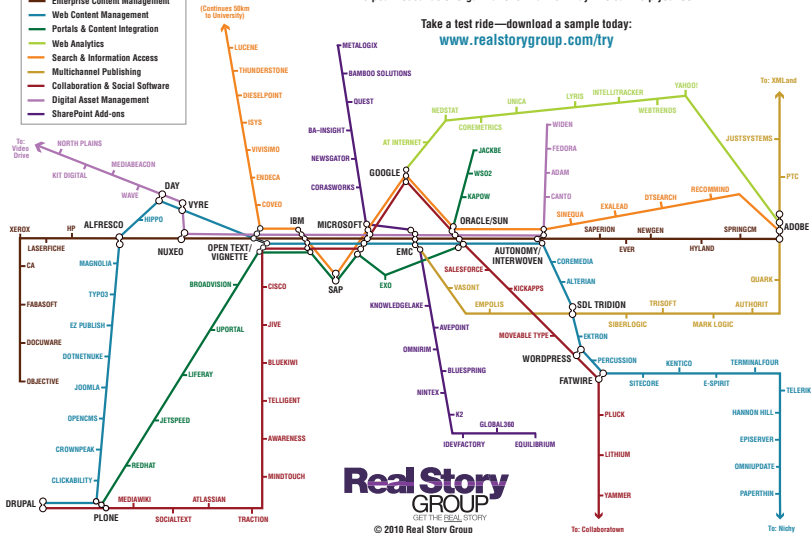
2010 Content Technology Vendor Map

KEY	
—	Enterprise Content Management
—	Web Content Management
—	Portals & Content Integration
—	Web Analytics
—	Search & Information Access
—	Multichannel Publishing
—	Collaboration & Social Software
—	Digital Asset Management
—	SharePoint Add-ons

Real Story Group's vendor evaluations and advisory services have helped thousands of organizations find their way. We can help you too.

Take a test ride—download a sample today:

www.realstorygroup.com/try



Real Story
GROUP
GET THE REAL STORY

© 2010 Real Story Group
www.realstorygroup.com/vendorsmap/

To: Collaboration

To: Niche

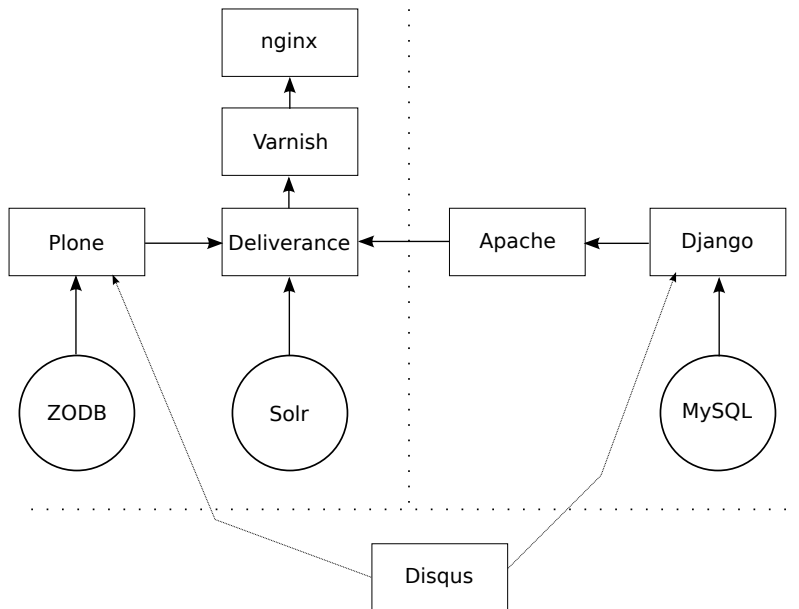
CMS Vergleich

	WordPress	Joomla 1.5.21 1.6	Drupal 6.19 7.0	Plone
Ease of Hosting and Installation	●	●	●	○
Ease of Setup: Simple Site	●	○	○	○
Ease of Setup: Complex Site	●	●	○	○
Ease of Use: Content Editors	●	○	○	●
Ease of Use: Site Administrator	●	○	○	○
Graphical Flexibility	●	●	●	●
Accessibility and Search Engine Optimization	○	○ ●	○	●
Structural Flexibility	○	○	●	●
User Roles and Workflow	○	○ ●	○	●
Community/Web 2.0 Functionality	●	○	●	○
Extending and Integrating	●	●	●	●
Security	○	○	○	●
Support/Community Strength	●	●	●	●

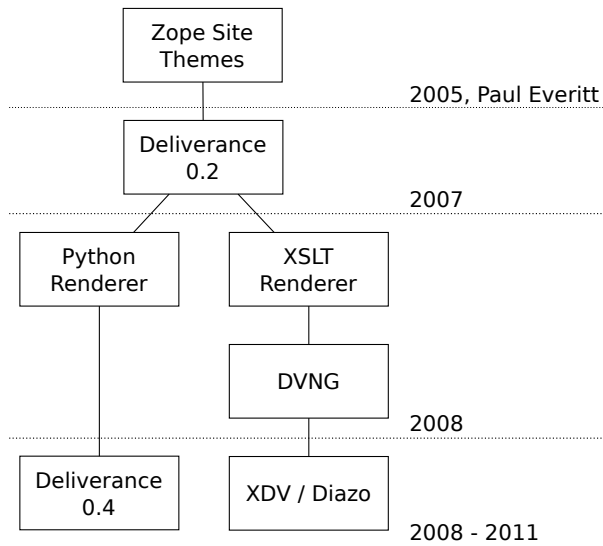
• **None** ○ **Fair** ○ **Solid** ● **Excellent** ●

Source: idealware - December 2010 - Comparing Open Source Content Management Systems

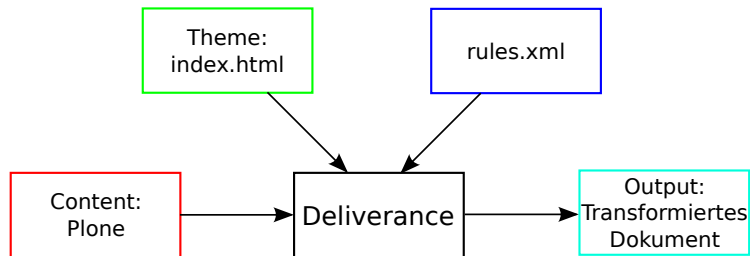
Systemskizze



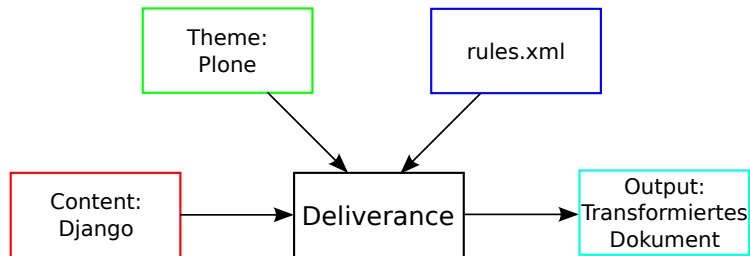
Deliverance - History



Deliverance - Plone Proxying



Deliverance - Django Proxying



Weitere Integrationskomponenten

- ▶ Buildout
- ▶ Solr
- ▶ Disqus

Ausblick

Start: April 2011

- ▶ <http://helsinki.adm.at>

Ausblick

Start: April 2011

- ▶ `http://helsinki.adm.at`
- ▶ `https://github.com/nnrorschmidt/helsinki`

Ausblick

Start: April 2011

- ▶ <http://helsinki.adm.at>
- ▶ <https://github.com/nnrorschmidt/helsinki>
- ▶ <https://github.com/thet/thet.helsinki.buildout>
- ▶ <https://github.com/thet/helsinki-theme>