

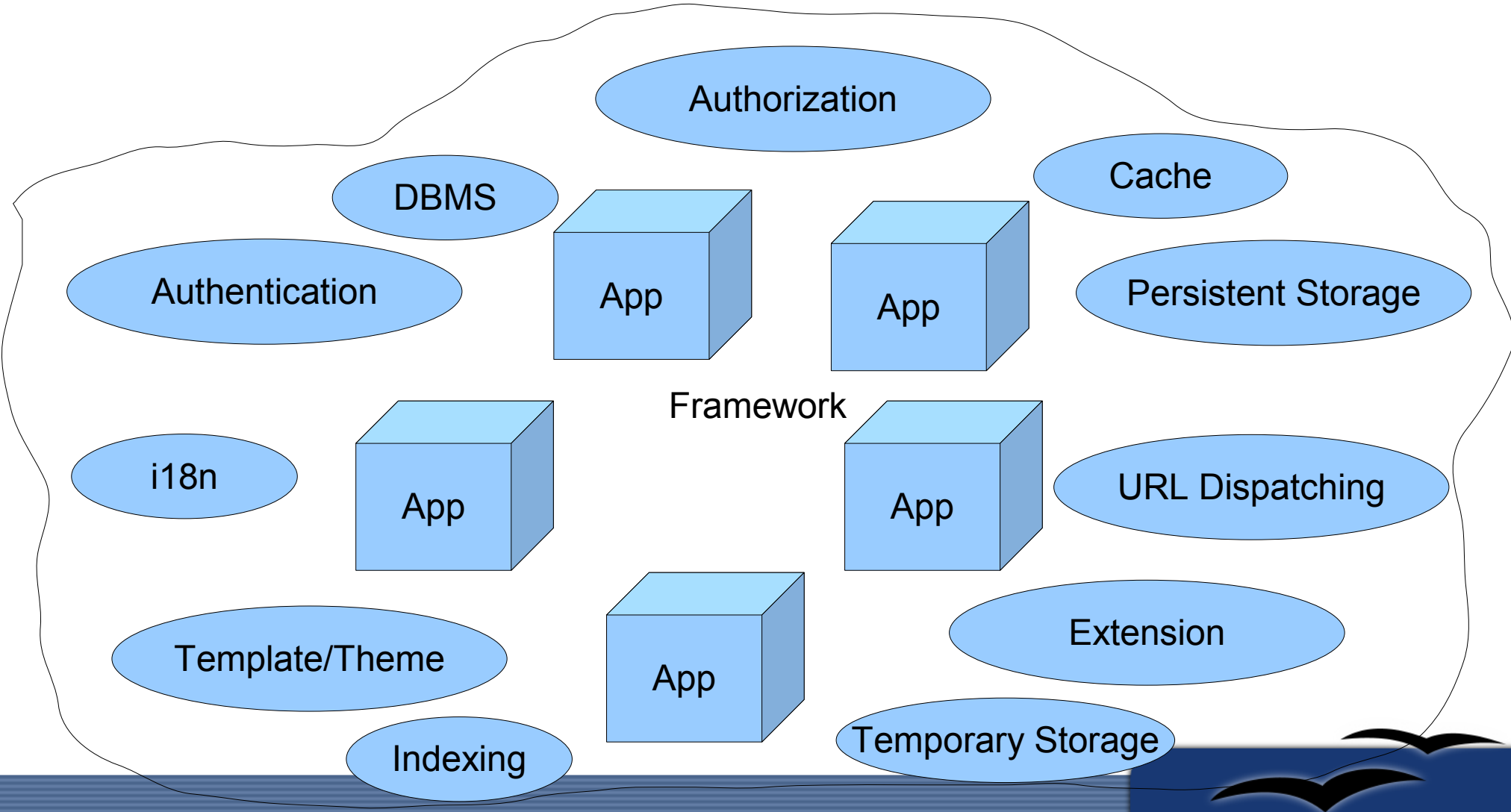
Python on Zope

Do not underestimate the power of Python

Sugree Phatanapherom
sugree_ph@thaigrid.or.th



Components of a Web



Zope

- Zope is not a content management system
- Zope is not a content management framework
- Zope is an application server
- Zope provides a content management framework
- Plone is a content management system running on top of Zope

Zope and Plone

Front-office

ThaiGrid Drug Design Portal

Better and Faster Drug Design with the Power of Grid Technology

User: HyperChem (2D Model) → Ligand Pre-OPT → GAMESS (Optimizer) → Ligand Post-OPT

GRID: AutoDock (Input, Output), Scheduler (SQMS)

Web Portal: Input, Output

Plone

Back-office

ZOPE

Logged in as admin | Zope Quick Start | Go

Type	Name	Size	Last Modified
Folder	Control_Panel (Control Panel)		2006-04-08 01:10
Folder	SiteWideVhost		2004-04-29 11:53
File	accessRule.py (Plone Access Rule)		2004-04-21 12:57
Folder	acl_users (User Folder)		2004-04-21 12:57
Folder	browser_id_manager (Browser Id Manager)		2004-04-21 12:57
Folder	drug_design_portal (Portal)		2006-01-18 13:51
Folder	error_log		2004-04-21 12:57
File	index_html	1 Kb	2004-04-21 12:57
Folder	session_data_manager (Session Data Manager)		2004-04-21 12:57
File	standard_error_message	1 Kb	2004-04-21 12:57
File	standard_html_footer	1 Kb	2004-04-21 12:57
File	standard_html_header	1 Kb	2004-04-21 12:57
File	standard_template.pt	1 Kb	2004-04-21 12:57
Folder	temp_folder		2006-04-08 01:10

Buttons: Rename, Cut, Copy, Delete, Import/Export, Select All

Zope

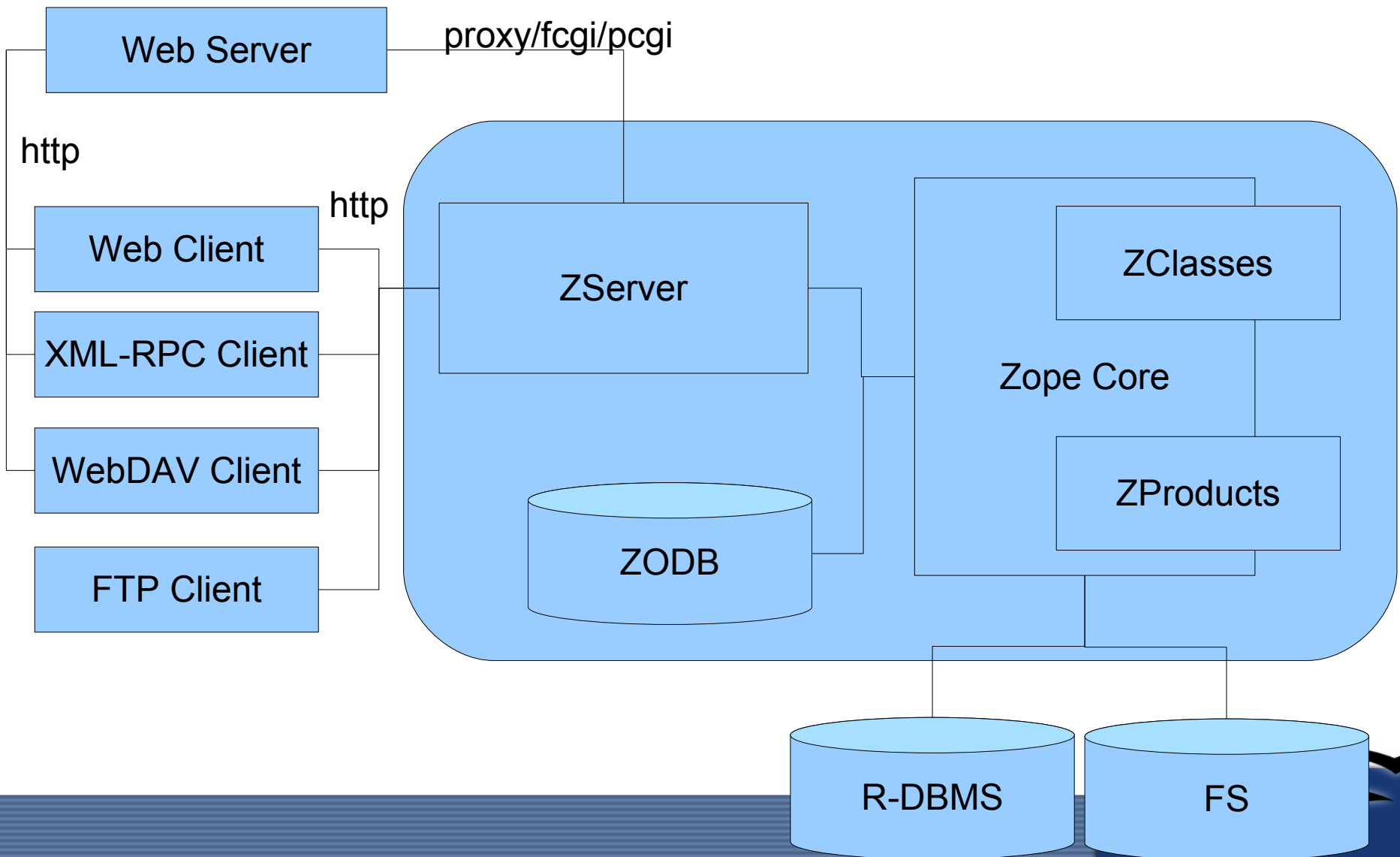


Zope's Concepts

- Object-Orientation
 - Everything are object
- URLs are RESTful
- All objects are stored in a ZODB
- Multi-ZODB is called ZEO

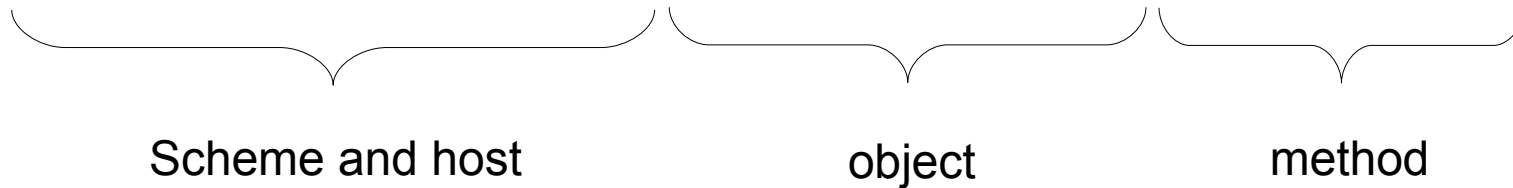


Architecture



URL Dispatching

`http://tgcc.cpe.ku.ac.th/Drugworkshop1/sendto_form`



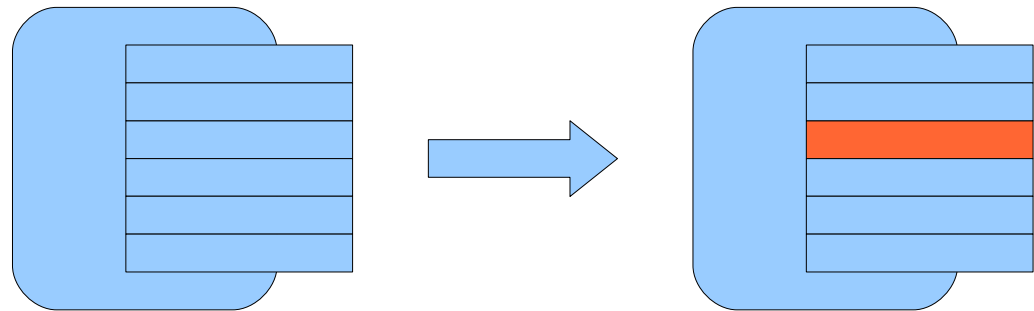
`Drugworkshop1.sendto_form()`



Extension

- Services
- Located by URL
- Standard Services
 - Access Rule
 - Cache
 - Mail
 - Log
 - Virtualhost
 - Indexing
 - Internationalization

- Overriding
 - Replace the object for any specific URL



Multi-site/Virtualhost

RewriteRule

```
^(.*)$ http://localhost:8090/VirtualHostBase/https/tgcc.cpe.ku.ac.th:443/drug_design_portal/VirtualHostRoot/$1  
[P,L]
```

RewriteRule

```
^/zmi[/]?(.*)$  
http://localhost:8090/VirtualHostBase/https/tgcc.cpe.ku.ac.th:443/VirtualHostRoot/_vh_zmi/$1  
[P,L]
```

RewriteRule

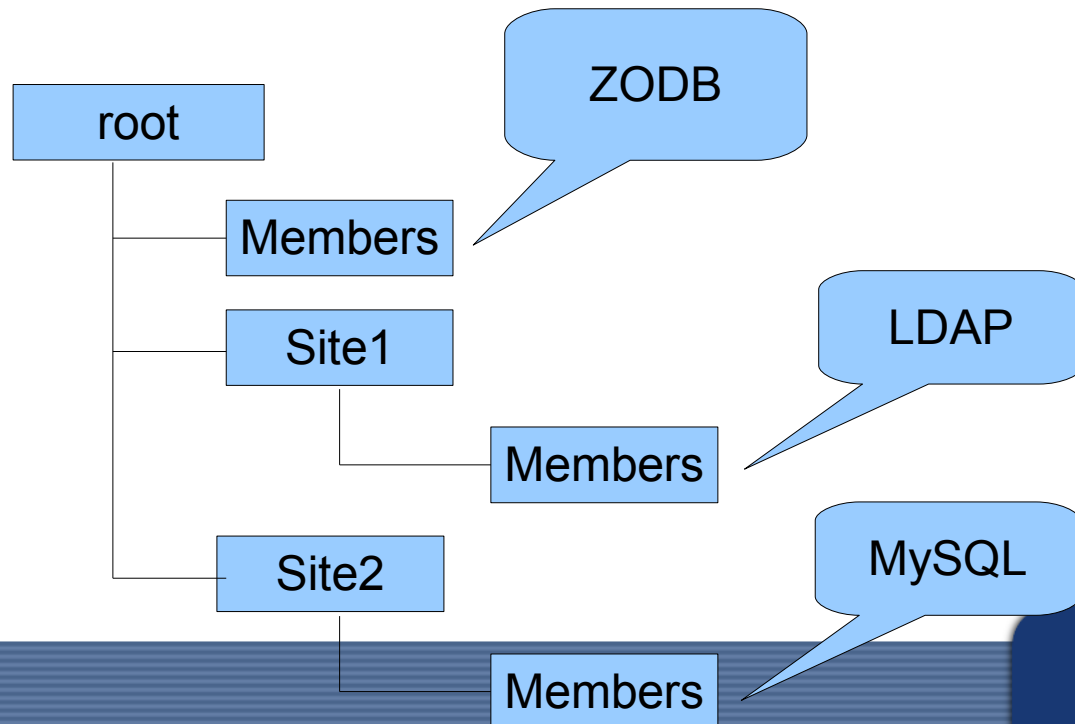
```
^(.*)$  
http://localhost:8090/VirtualHostBase/http/tgcc.cpe.ku.ac.th:80/drug_design_portal/VirtualHostRoot/$1  
[P,L]
```

/VirtualHostBase[Zope URL]/VirtualHostRoot[Path]



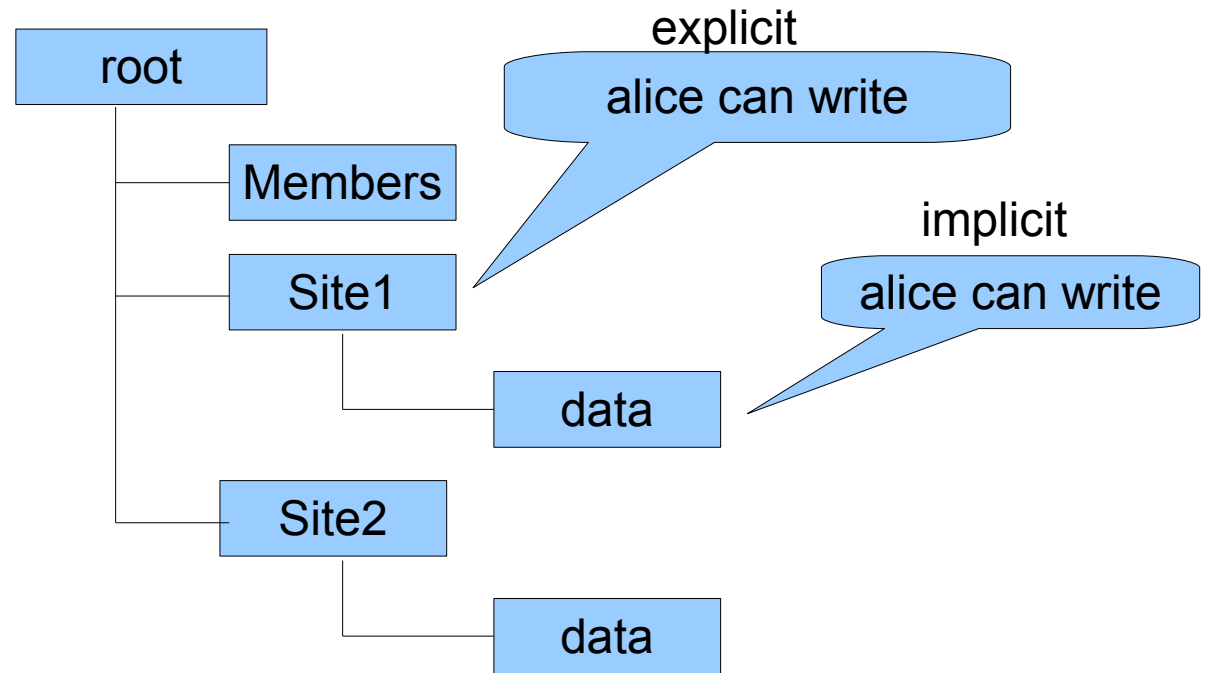
Authentication

- So called “User Folder”
- Hierarchical authentication
 - Upper level users have permissions on all lower levels



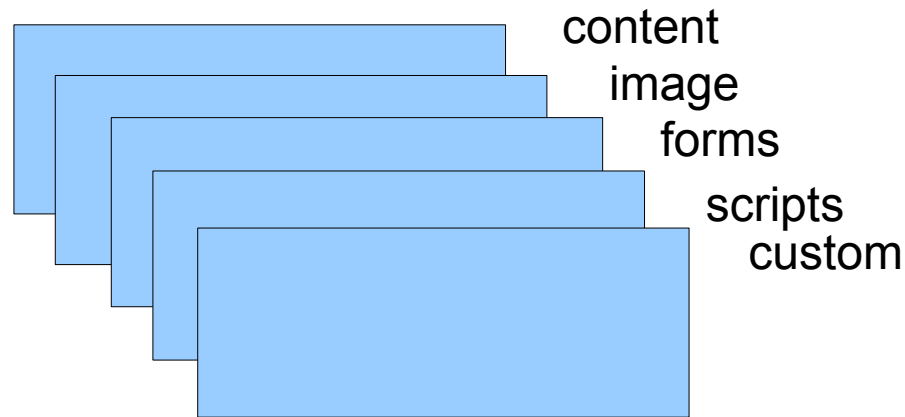
Authorization

- Role-based
 - Folder
 - Object
- Inherited



Template/Theme

- In Plone, a template is called “skin”
- A skin consists of multiple layers
- Layer is a folder
- Look up process is to find the file in each layer from first to last



Internationalization

- Translatable texts
 - ZPT
 - gettext po
 - Import
 - Export
- Translatable contents
 - I18NLayer
 - Transparent



Indexing

- Built-in catalog
- Programmatic catalog update
 - Extensible word filter
 - Splitter
 - CaseNormalizer



Storage

- Persistent Storage
 - Transaction-based
 - Undo-able
- Temporary Storage
 - Transaction-based
 - Refresh on next start



Cache

- HTTP
 - Static files
 - CSS
 - HTML
 - Images
- RAM
 - Python scripts
 - External executables



DBMS

- Connection to a DBMS is an object
- A query can also be an object
- The query method could be called programmatically

Z SQL Method at /drug_design_portal/ligand_query_method

Title	<input type="text" value="Ligand Query Method"/>
Connection Id	<input type="text" value="ligand_db_connection"/>
Arguments	<input type="text" value="criteria"/>

```
SELECT I.structureID,  
       I.formula,  
       N.stName,  
       C.casNum,  
       Pn.pntName  
FROM   structureInfo as I  
LEFT JOIN structureName as N on N.structureID = I.structureID  
LEFT JOIN stPlant as P on P.structureID = I.structureID  
LEFT JOIN PlantName as Pn on Pn.plantID = P.plantID  
LEFT JOIN casNumber as C ON C.structureID = I.structureID  
where C.casNum <> '' <!--#var criteria-->
```



Application

- Add/remove products
 - Zope
 - Plone

Product Management at [/Control_Panel/Products](#)

Type Name

- Archetypes (Installed product Archetypes (1.2.5-rc4))
- BTreeFolder2 (Installed product BTreeFolder2 (BTreeFolder2-1.0))
- CMFActionIcons (Installed product CMFActionIcons (0.9))
- CMFBoard (Installed product CMFBoard (2.0))

Add/Remove Products

[▲ Up to Plone Setup](#)

This is the Add-on Products install section, you can add and remove products in the lists below.

Products available for install

-  **CMFBoard 2.0**
[Product Description](#)
-  **CMFPhoto 0.4.2**
[Product Description](#)
-  **CMFPhotoAlbum 0.4final**
[Product Description](#)
-  **CMFSin 0.6.1**
[Product Description](#)



Application Development

- Logic
 - Restricted Python script
- Skin
 - Page Template
 - DTML
- Schema-driven
 - Archetypes



Archetypes

- Fields

- Object, Reference
- Date Time
- Photo, Image
- Float, Integer, String, Text, Fixed Point, Lines, Boolean
- Attachment, File
- Computed

- Widgets

- Integer, Decimal, String, TextArea, RichText, Boolean, Lines, Password
- Calendar, Selection, MultiSelection
- Id, Reference, Computed, Keyword
- Image, File, InAndOut
- Label, PickList

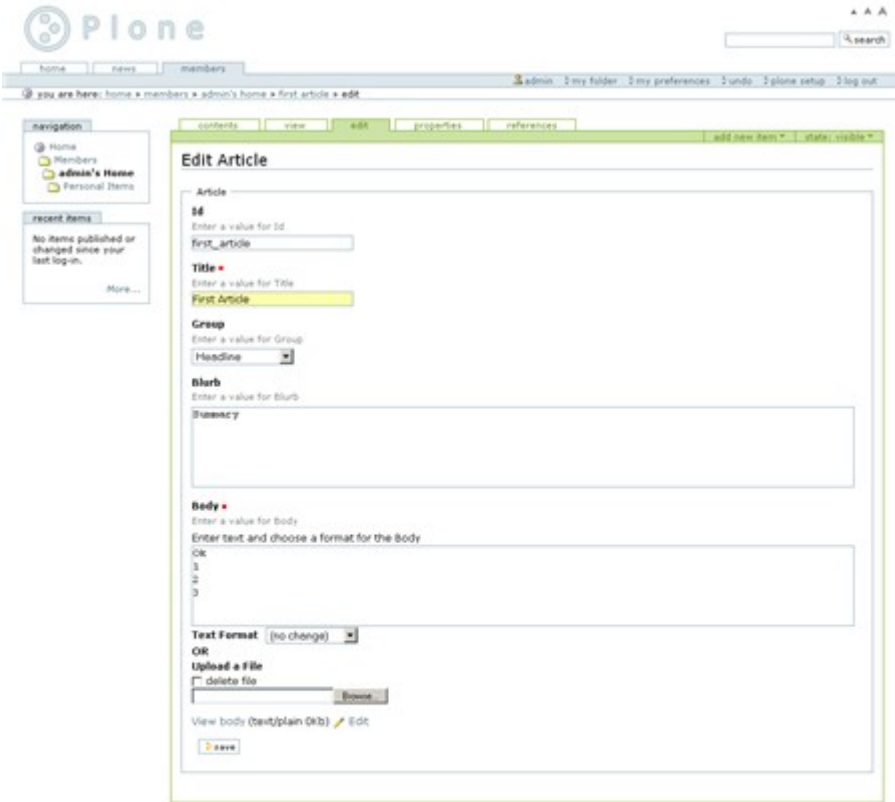


Product Structure

- Extensions
 - `__init__.py`
 - `Install.py`
- skins
 - blognone
 - `article_view.pt`
- `__init__.py`
- `README.txt`
- `config.py`
- `Article.py`



Product in Action



The screenshot shows the Plone 'Edit Article' interface. At the top left is the Plone logo and a search bar. Below it is a navigation menu with 'home', 'news', and 'members'. A breadcrumb trail reads 'you are here: home > members > admin's home > first article > edit'. The main content area is titled 'Edit Article' and contains several form fields: 'Id' (with value 'first_article'), 'Title' (with value 'First Article'), 'Group' (with value 'Headline'), 'Blurb' (with value 'Summary'), and 'Body' (with value 'Ok'). There are also options for 'Text Format' and 'Upload a File'. A 'save' button is at the bottom.



The screenshot shows the Plone 'First Article' view. At the top left is the Plone logo and a search bar. Below it is a navigation menu with 'home', 'news', and 'members'. A breadcrumb trail reads 'you are here: home > members > admin's home > first article'. The main content area is titled 'First Article' and displays the article's metadata: 'Id: first_article', 'Title: First Article', 'Group: Headline', 'Blurb: Summary', and 'Ok: 1, 2, 3'. A footer contains the text 'Plone and its visual design is Copyright © 2000-2006 by Alexander Limi, Alan Runyan, Vidar Andersen.'



The footer of the Plone interface, showing the copyright notice 'Plone and its visual design is Copyright © 2000-2006 by Alexander Limi, Alan Runyan, Vidar Andersen.' and a list of logos for various standards and protocols.



Schema and Actions (Article.py)

```
schema = BaseSchema + Schema((
    StringField('group',
        vocabulary=ARTICLE_GROUPS,
        widget=SelectionWidget(),
    ),
    StringField('blurb',
        searchable=1,
        widget=TextAreaWidget(),
    ),
    TextField('body',
        searchable=1,
        required=1,
        primary=1,
        default_output_type='text/html',
        allowable_content_types=('text/plain',
                                'text/structured',
                                'text/html'),
        widget=RichWidget(label1='Body'),
    ),
),
marshall=PrimaryFieldMarshaller(),
)
```

```
class Article(BaseContent):
    "This is a sample article"

    schema = schema

    actions = ({
        'id': 'view',
        'name': 'View',
        'action': 'string:${object_url}/article_view',
        'permissions': (CMFCorePermissions.View,)
    },)
```



article_view.pt

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US"
  lang="en-US"
  metal:use-macro="here/main_template/macros/master">
<body>
  <div metal:fill-slot="main">
    <div metal:define-macro="main"
      tal:omit-tag="">
      <h1 tal:content="here/title_or_id" />

      <!-- do as you will -->
      <tal:block tal:define="field_macro field_macro|here/widgets/field/macros/view;
        mode string:view;
        use_label python:1">
        <div tal:repeat="field python: here.Schema().filterFields(isMetadata=0)">
          <span tal:condition="python:field.getName() not in ['body']">
            <span metal:use-macro="field_macro"/>
          </span>
        </div>
      </tal:block>
      <span tal:content="structure python: here.getBody(mimetype='text/html')"/>

    </div>
  </div>
</body>
</html>
```



To Zope or not to Zope

- To Zope
 - Objects and folders web structure
 - OO in mind
 - Complex data structure
- Not to Zope
 - Single user blog
 - Flat data structure

