ZOPE 3

Christophe Combelles

- 2004 : Indépendant (Gorfou sarl)
- 2006 : Python / Zope3
- 2007 : Afpy

- E-mail : ccomb@free.fr
- Jabber : ccomb@jabber.org
- IRC : #afpy, #zope3-fr, #zope3-dev

Gorfou

Plan

- Aperçu général
- Component architecture
- Exemples d'application, outils, (+ démo ?)

Historique

- 1998 : Bobo + Principia → Zope 1.9, 2.x
- 2001 : Digital Creations → Zope Corp.
- 2001 : début de réflexion sur Zope 3 (Component Architecture)
- 2005 : Zope 3.0, Zope 3.1
- 2006 : Zope 3.2
- 2007 : Zope 3.3
- 2008 : Zope 3.4

Pourquoi une réécriture ?

- pour éviter :
  - Héritage en cascade
  - Méthodes obscures
  - ZODB surchargée
  - Acquisition, Zclasses, DTML
- Pour garder :
  - Publication d'objets, ZODB, ZPT
- Pour moderniser, pythoniser, standardiser, simplifier, modulariser, agiliser, fiabiliser, réutiliser, s'ouvrir :
  - Code dans des fichiers
  - Code + pythonique, plus lisible
  - Interfaces / Composants réutilisables
  - Design patterns
    - Délégation / adaptation / Proxy
  - WSGI, Buildout, Eggs

Crise d'identité

- Serveur d'application ? Framework ? Bibliothèque ?
- Éclatement en eggs
  - 500 paquets sur le cheeseshop

  - Retard dans la release 3.4
- Zope en tant que dépendance de votre appli

Zope 3 aujourd'hui

- Serveur d'application complet compatible WSGI
- Ensemble de bibliothèques Python
- Méthode de programmation par composants
- Outils web :
  - Documentation APIDOC
  - Debug debug wsgi, skin de debug
  - Gestion ZMI (Skin Rotterdam)
- Outils en ligne de commande
  - Debug Instance en ligne de commande
  - Démarrage rapide zopeproject, grokproject
- Des entreprises et une communauté

Bibliothèques Zope 3

Component Architecture
zope.interface
zope.component

zope.app.*
zope.app.basicskin
zope.app.container
zope.app.folder
zope.app.file
zope.app.intid
etc.....

zope.*
zope.schema
zope.formlib
zope.publisher
zope.traversing
zope.viewlet
zope.pagetemplate
zope.rdbms
zope.sqlachemy
etc.....

z3c.*
z3c.form
z3c.template
z3c.pagelet
z3c.macro
z3c.layer.*
etc.....

zc.*
zc.catalog
zc.relation
zc.comment
etc.....

hurry.*
hurry.query
hurry.workflow

lovely.*

ks.*

z3ext.*

gp.*
gp.svnfolder

ZODB
ZODB, persistent, transaction, BTrees

Point de vue fonctionnel

- Création de modèles / schémas
- Templating
- Indexation / recherche
- Base de données objets
- Accès à des bases SQL
- Utilisation d'ORM
- Sécurité / Authentification modulable
- Formulaire webs (génération / validation)
- AJAX (kss, jquery)
- Framework de test
- Framework de dépréciation
- Framework de migration de la ZODB
- Workflow
- i18n, unicode
- ...

Méthodologie de développement

→ dirigé par les tests et la documentation

- Objets de base
  - Fonctionnalités
  - Interfaces / schémas
  - Organisation des objets
  - Doctest d'aide à la conception
  - Classes, implémentation
- Interface utilisateur
  - Skin perso
  - Vues / templates /formulaires
  - Doctests fonctionnels (zope.testbrowser)
- Tests unitaires pour chaque bug

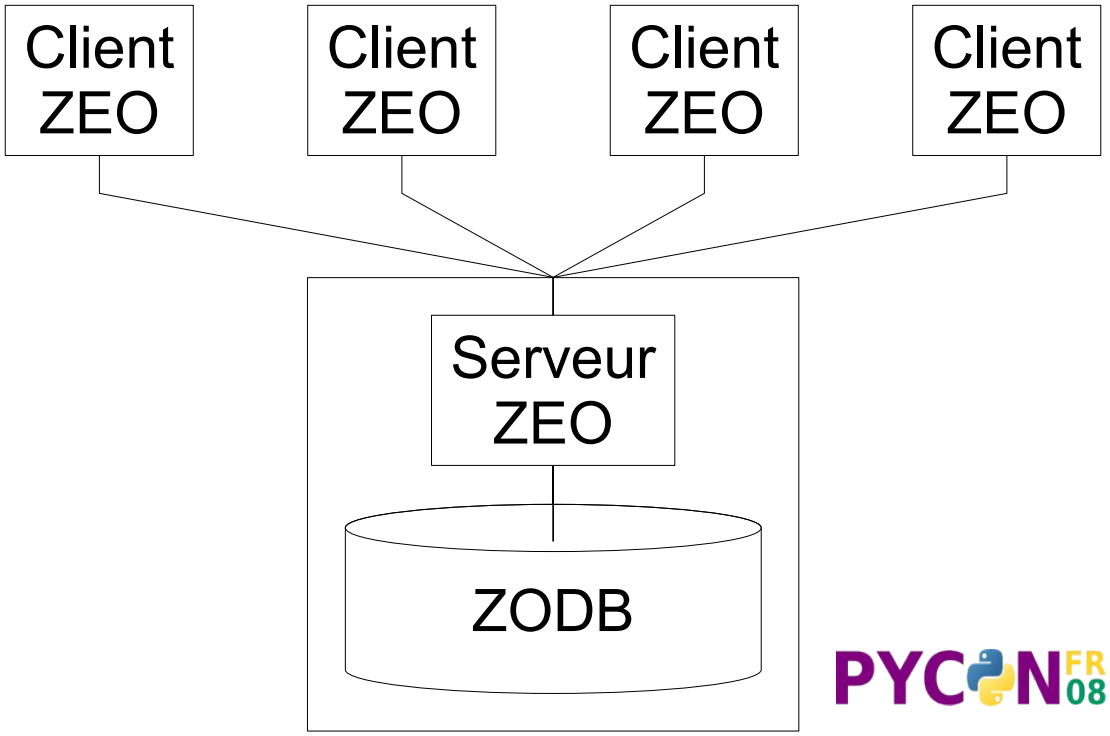
ZODB

- Objet conteneur racine (root)
- Hiérarchie d'objets Python
- http://localhost/objet1/../objetN/@@vue.html
- Root
  - objet1
    - Objet2
- Traversing modifiable

ZODB v3.8

- Persistance
- BTrees
- Transactions
- Historique
- Blobs
- Plusieurs méthodes de stockage
  - Filestorage
  - DirectoryStorage
  - RelStorage
- ZEO
- ZODB ou pas ?

```
from persistent import Persistent

MyObject(Persistent):
    pass
```



Component Architecture

- 2 paquets clés :
  - zope.interface
  - zope.component

Objet

```
class Blog( ):

    posts = [ ]

    def post(self, message):
        ####################
        ####################
        ##implémentation##
        ####################
```

Interface

```
from zope.interface import Interface, Attribute

class IBlog(Interface):

    posts = Attribute('posts')

    def post( ):
        'make a new post'
```

Objet avec interface

```
from zope.interface import implements

class IBlog(Interface):

    posts = Attribute('posts')

    def post( ):
        'make a new post'

class Blog( ):

    implements(IBlog)

    posts = [ ]

    def post(self, message):
        'make a new post'
```

ZOPE 3

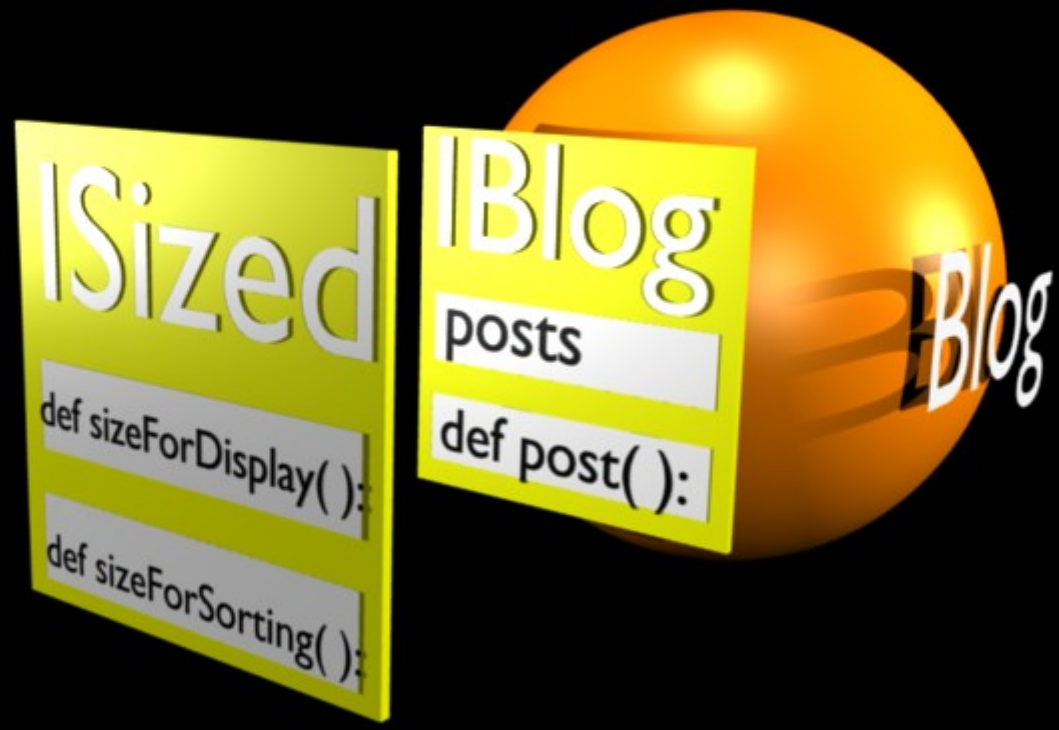
PYCON FR 08

API de zope.interface

- implements(Interface)
- IBlog.providedBy(blog)
- IBlog.implementedBy(Blog)
- alsoProvides(blog, Interface2)
- verifyObject(IBlog, blog)
- verifyClass(IBlog, Blog)
- Interface.setTaggedValue(tag, value)

Taille du Blog

from zope.size.interfaces import ISized



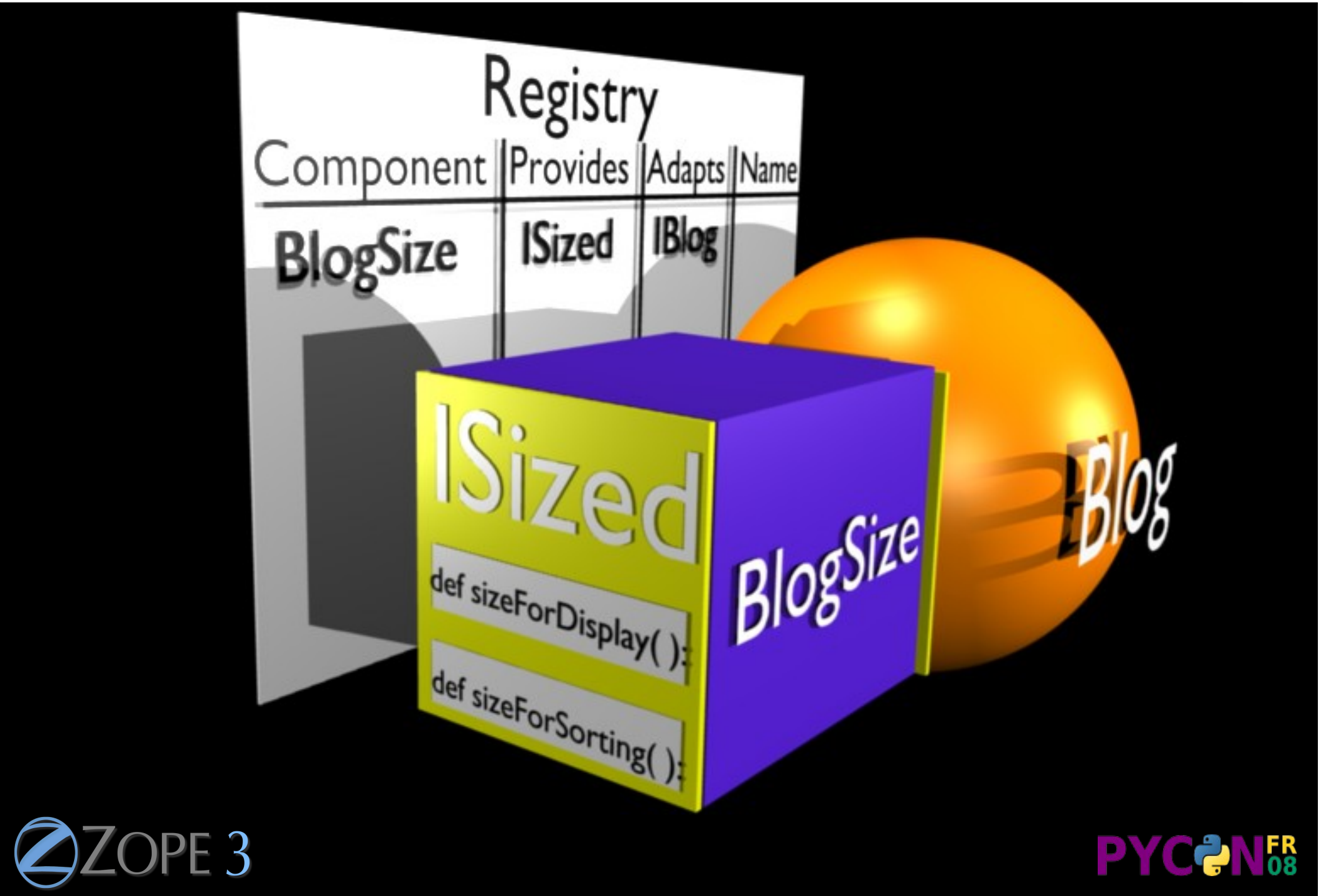
Adapter

```
from zope.component import adapts

class BlogSize( ):

    implements(ISized)
    adapts(IBlog)

    def __init__(self, context):
        self.context = context

    def sizeForSorting(self):
        return len(self.context.posts)

    def sizeForDisplay(self):
        return unicode(self.sizeForSorting())
```

Récupération de l'*adapter*

- Appel direct : nécessite un import explicite
  - blog_size = BlogSize(blog)
- Component Architecture
  - blog_size = zope.component.getAdapter(blog, ISized)
- Écriture simplifiée
  - blog_size = ISized(blog)

Registre global de composants



Inscription dans le registre

- Python :
  zope.component.provideAdapter(BlogSize)

- Langage XML de configuration : ZCML
  <adapter factory="BlogSize" />

- Grok :
  class BlogSize(grok.Adapter):
      grok.context(Blog)
      grok.provides(ISized)
      ...

L'*adapter* fait ce qu'il veut

```
class Adapter( ):

    implements(IInterface)
    adapts(IMyObject)

    def __init__(self, context):

        context.foobar = 12
```

Interface marqueur

Registry

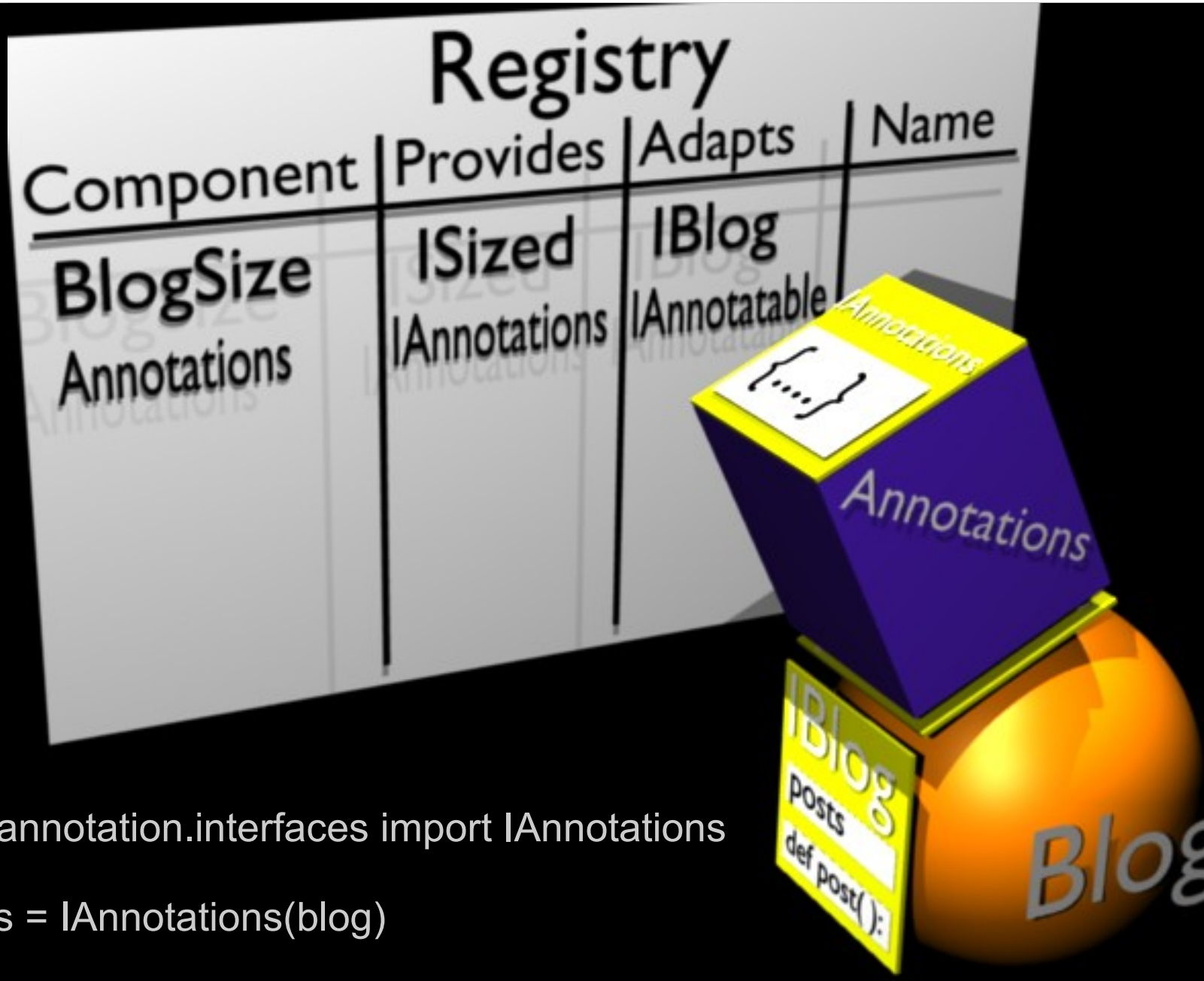
Component	Provides	Adapts	Name
BlogSize	ISized	IBlog	
Annotations	IAnnotations	IAnnotatable	


```
<class  class="Blog">
     <implements  interface="IAnnotatable" />
</class>
```

ZOPE 3

PYCONFR 08

Annotations

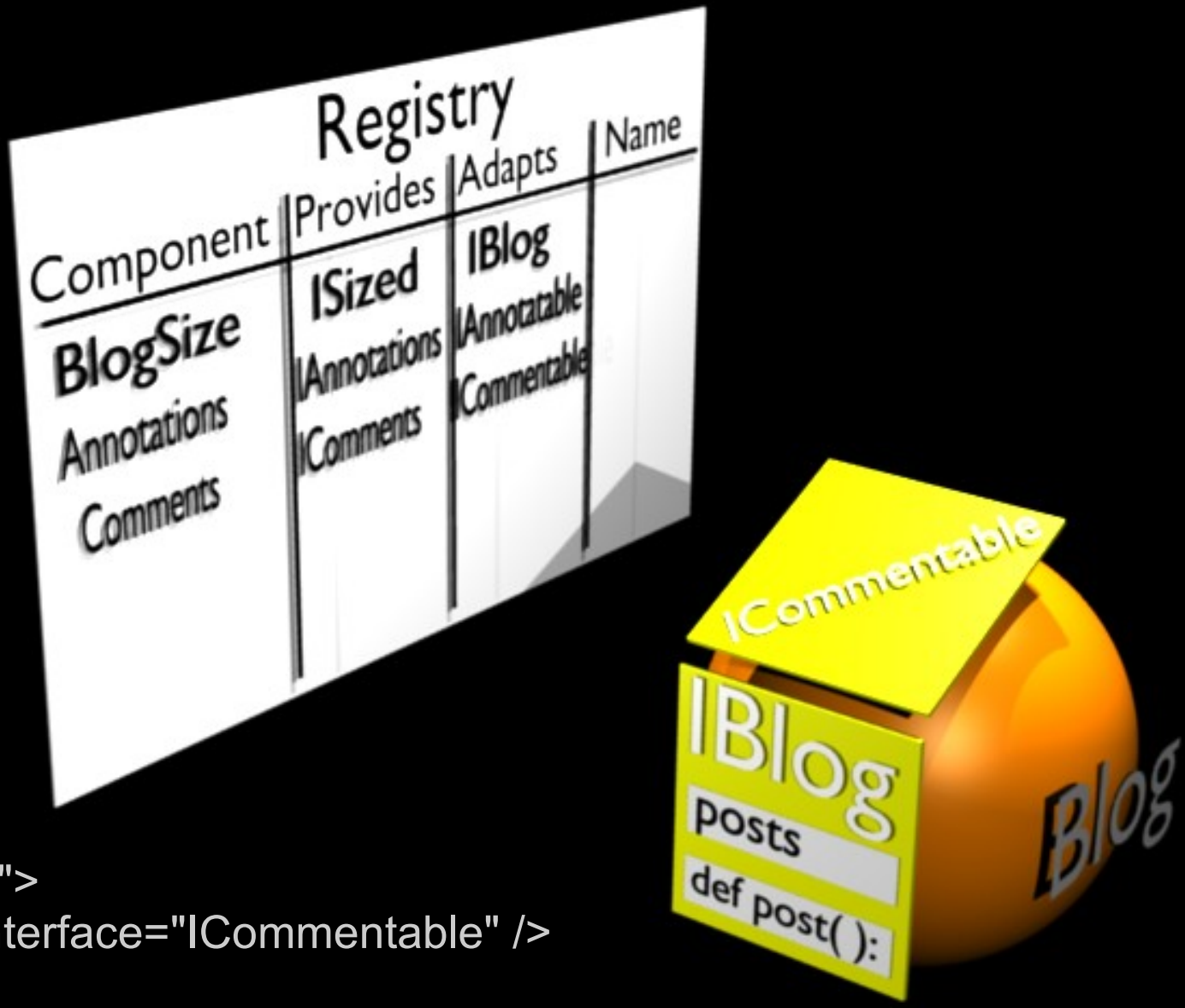



```
from zope.annotation.interfaces import IAnnotations

annotations = IAnnotations(blog)
```

ZOPE 3

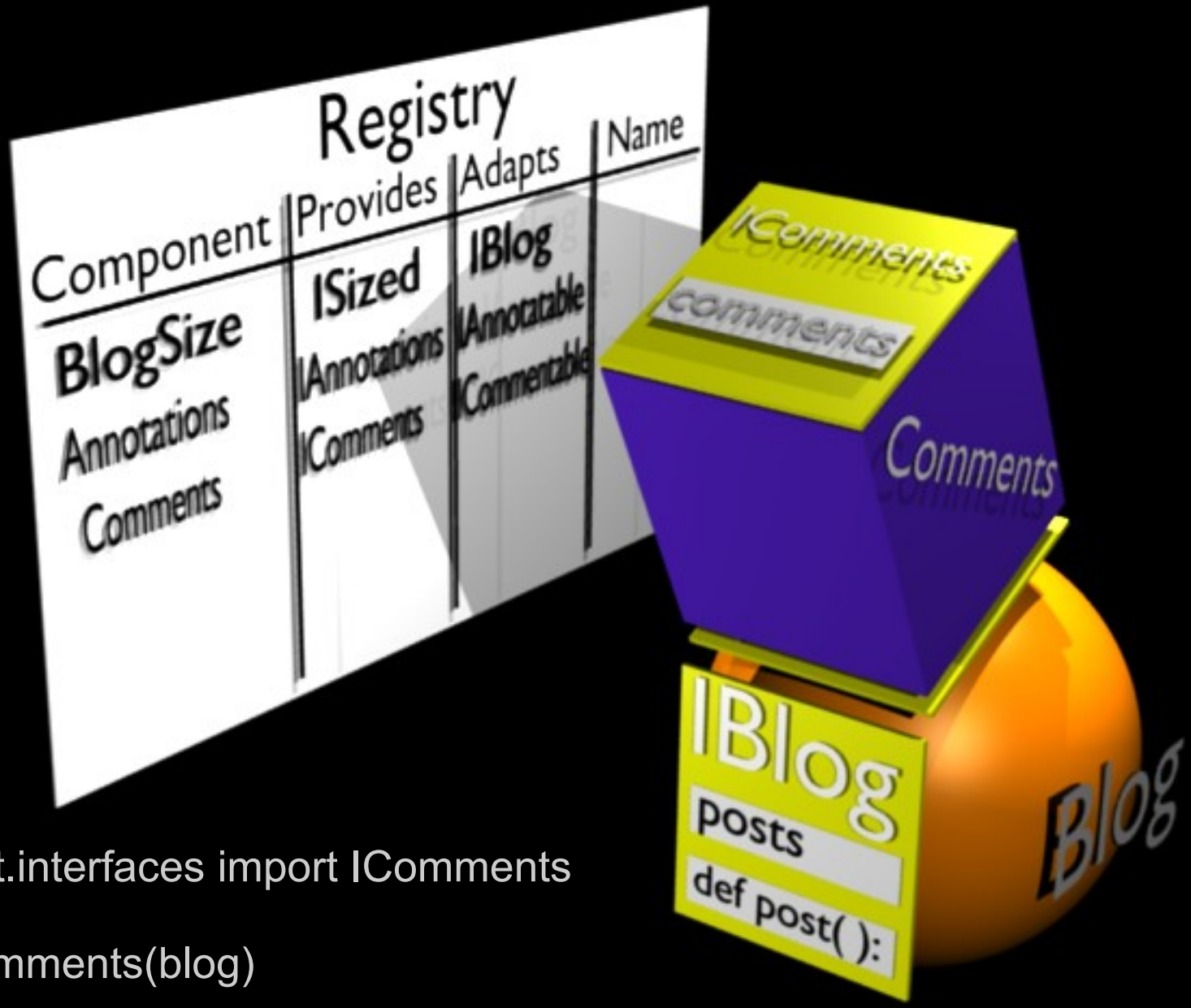
PYCONFR 08

Gestion des commentaires



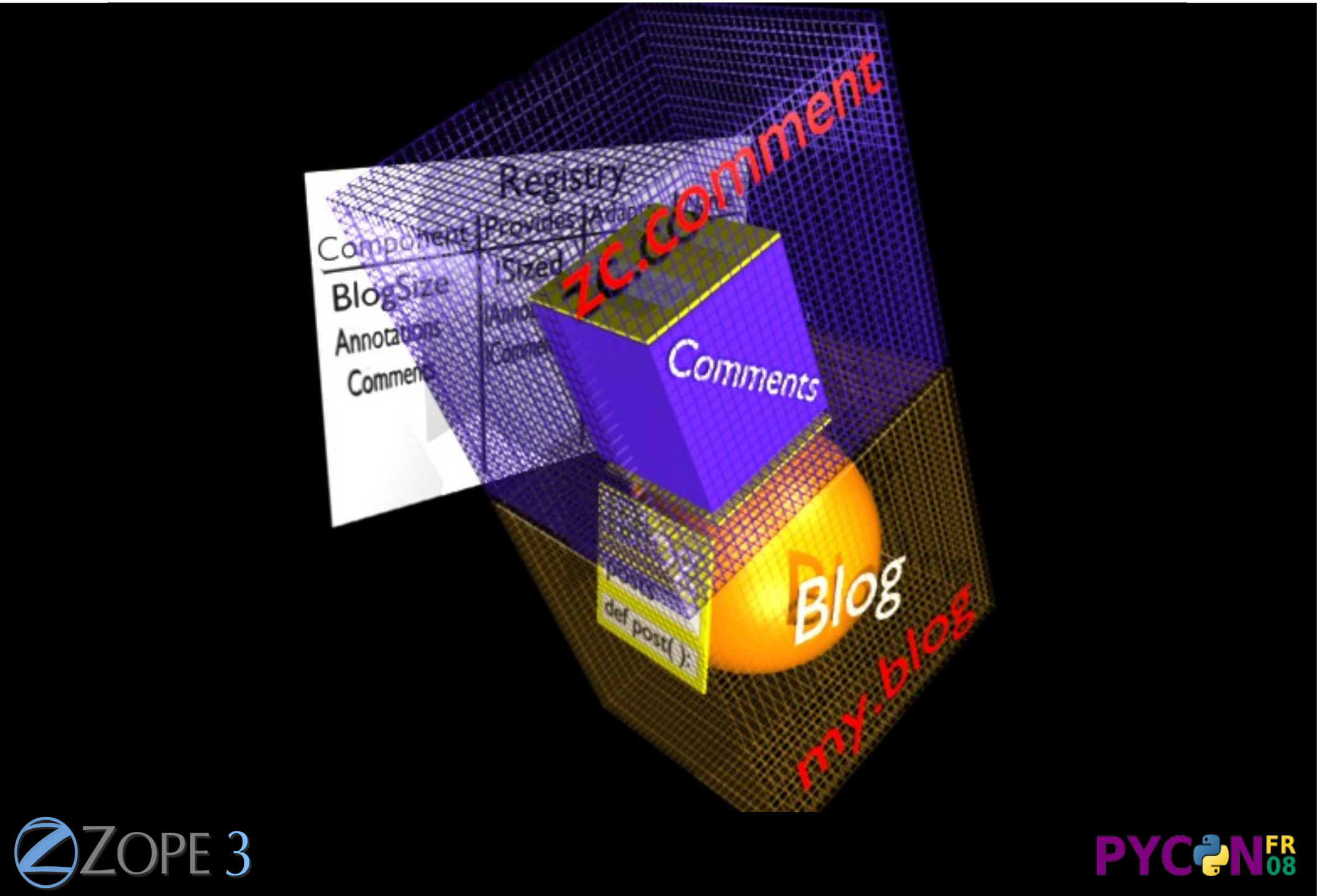

```
<class  class="Blog">
     <implements  interface="ICommentable" />
</class>
```

Ex : gestion des commentaires




```
from zc.comment.interfaces import IComments

comments = IComments(blog)
```

Commentaires



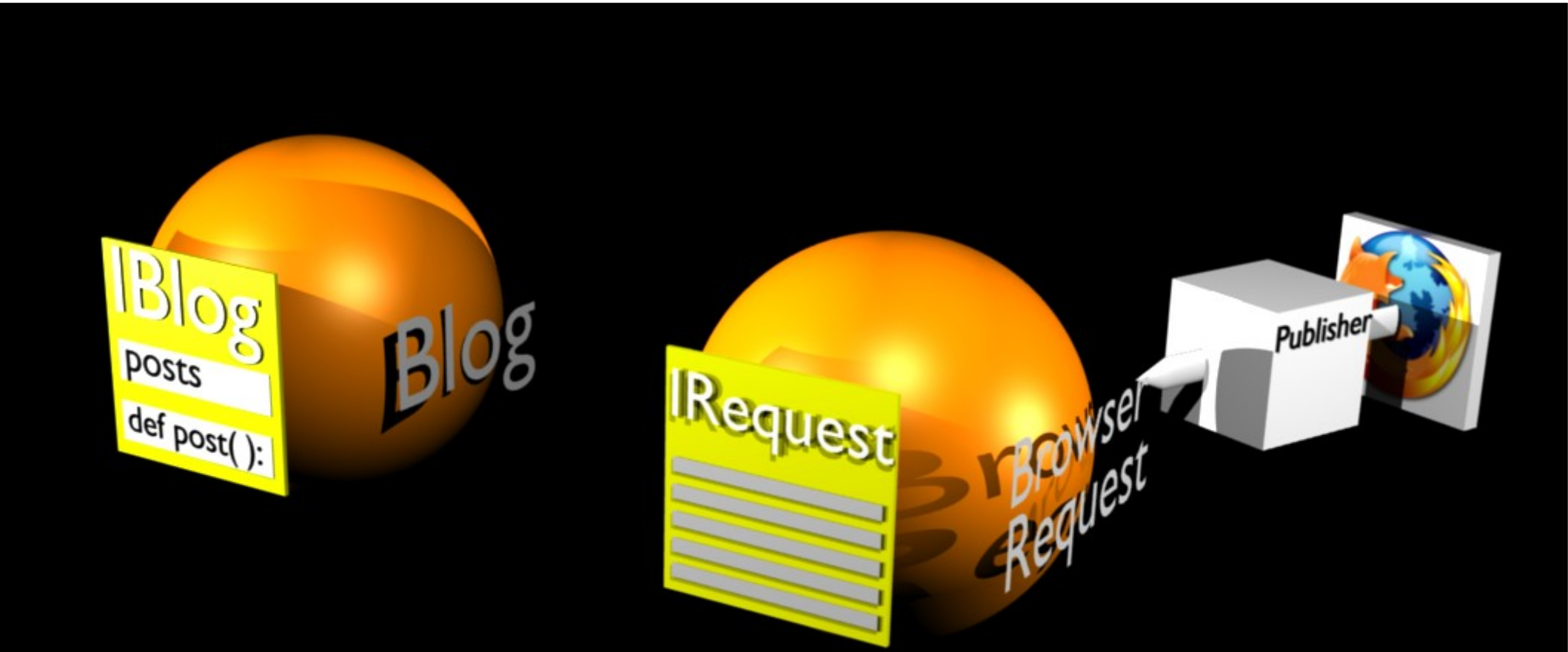
Cas d'utilisation des adapters

- Métadonnées (date de création, créateur, tags)
- Commentaires
- URL d'un objet
- Traversing
- Taille
- Catégorie
- Texte à indexer
- Vignette d'aperçu

Affichage du blog

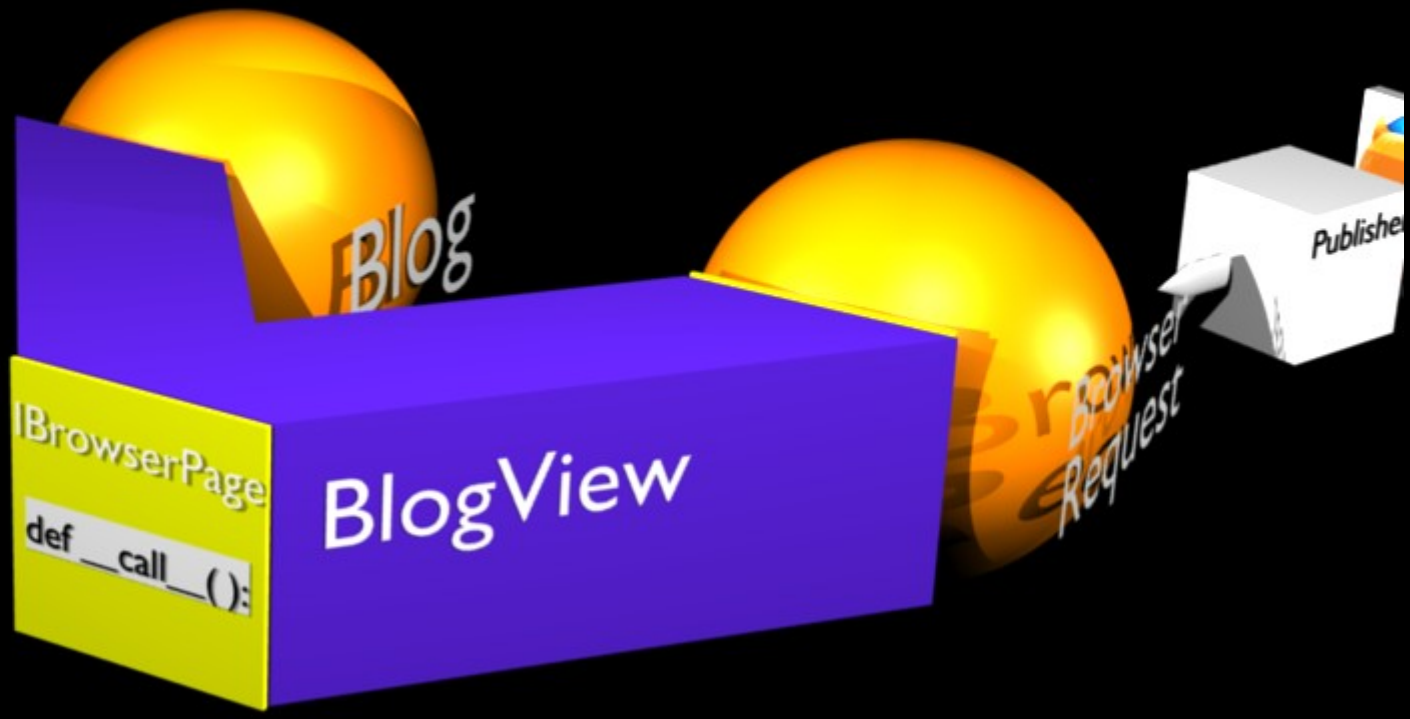
Affichage du blog

Affichage du blog



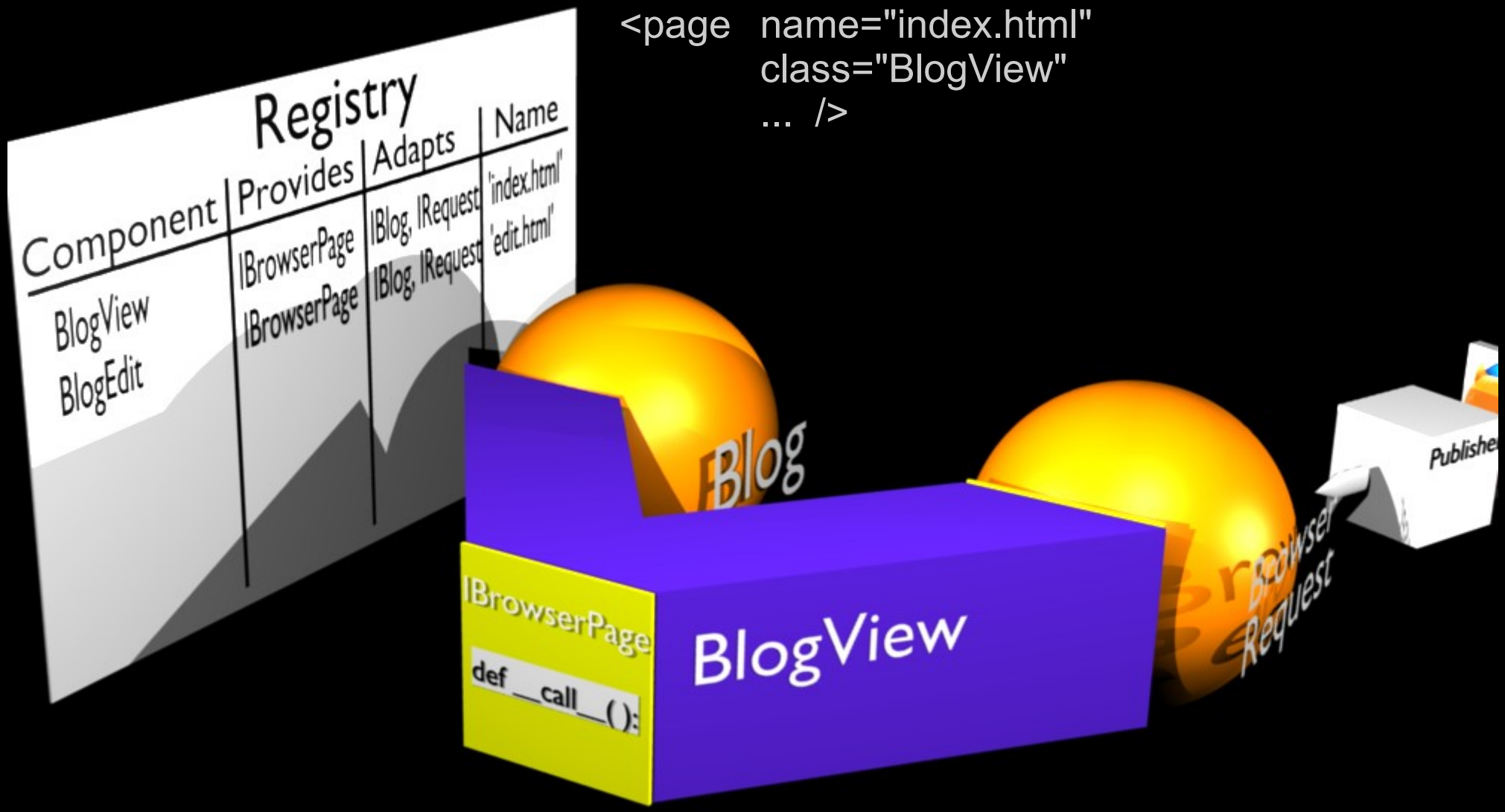
Affichage du blog

```
from zope.publisher.browser import BrowserPage

class BlogView(BrowserPage):

    def __call__(self):
        # return HTML
```

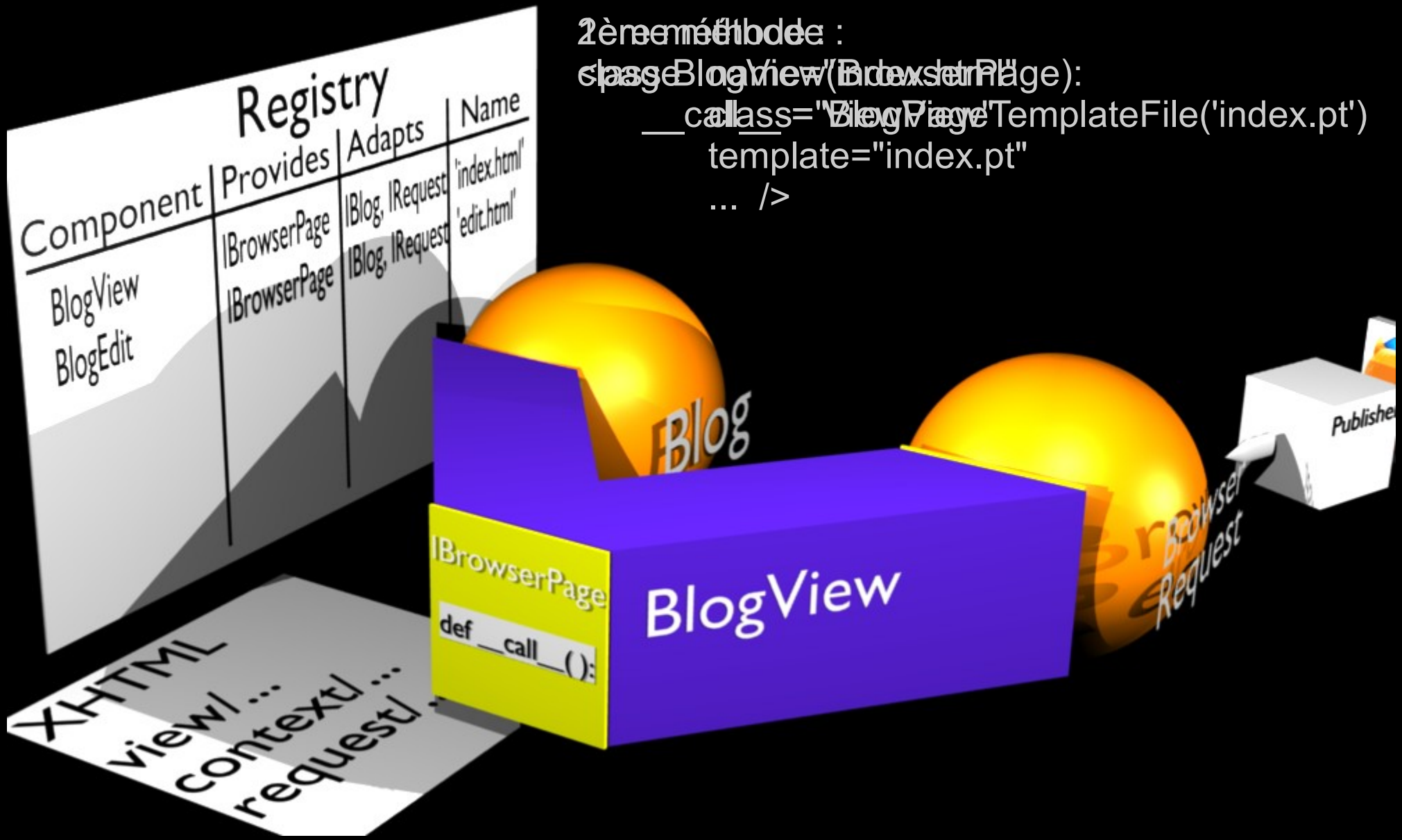


Inscription de la vue

```
<page   name="index.html"
        class="BlogView"
        ... />
```



Template pour la vue



2ème méthode :

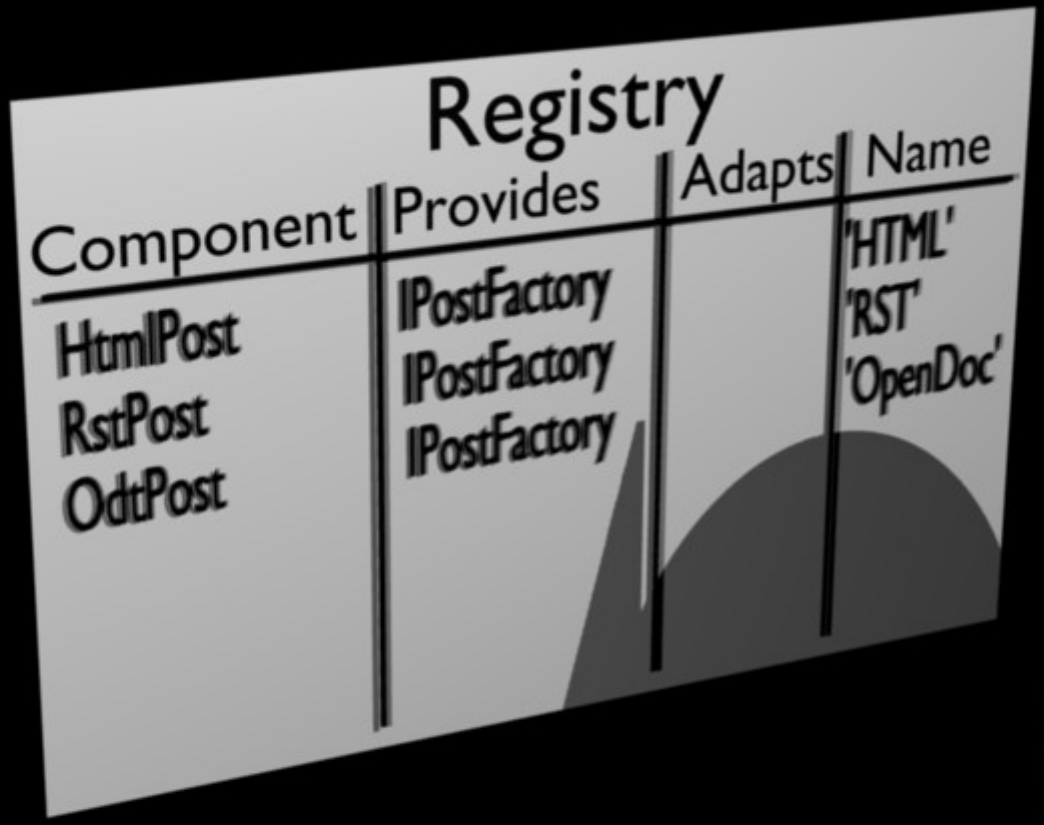
```
class BlogView(BrowserPage):
    __call__ = ViewPageTemplateFile('index.pt')
```

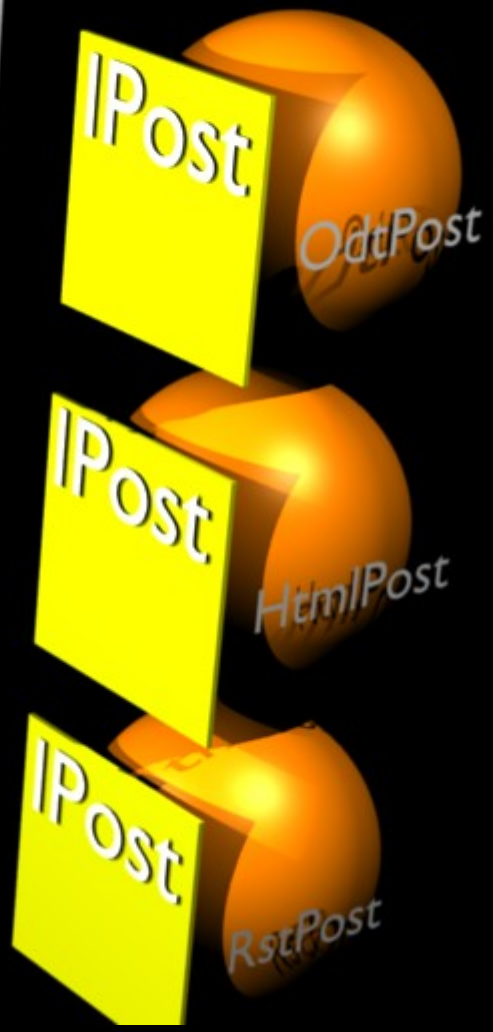
```
<browser:page
    class="BlogView"
    template="index.pt"
    ... />
```

Adaptateurs / Adapters / Utilities

- *Utility :* 0 objet adapté
- *Adapter* : 1 objet adapté
- Double *adapter* : 2 objets adaptés
  - Vues
- Triple *adapter* : 3 objets adaptés
  - Content Provider (context, request, view)
  - Viewlet Manager (context, request, view)
- Quadruple *adapter* : 4 objets adaptés
  - Viewlet (context, request, view, manager)

Utilities






```
from zope.component import getUtility

postfactory = getUtility(IPostFactory, 'RST')
newpost = postfactory()
```

Utilities

- Factories
- Composants
- Catalog
- Authentification
- Mailer
- Roles
- Integer Id
- Encodeur
- Page par défaut (IDefaultPage)
- ...

Une démo !

- zopeproject / grokproject
- ZMI / Rotterdam
- APIDOC
- Debug WSGI
- ++debug++

ZOPE 3

Utilisateur : Manager [Déconnexion]

Emplacement : [racine] /

Navigation

[racine]

Ajouter :

- Catalogue d'index
- Identificateur par cookie
- Fournisseur de préférences par défaut
- Utilitaire d'enregistrement des erreurs
- Site Eztranet
- Fichier
- Dossier
- Image
- Permission
- Conteneur de sessions persistantes
- Utilitaire d'authentification
- Utilitaire d'annotation des Utilisateurs
- Conteneur de sessions en RAM
- Rôle
- Dossier de Gestionnaire de Site
- Domaine de traduction
- Générateur d'identifiants uniques

Contenu | Aperçu | Métadonnées | Introspection | Inscription

Gérer le Site | Gérer le processus | Ajouter | Chercher | Autoriser | Erreurs | Aide

	Nom	Titre	Taille	Créé	Modifié
☐	eztranet		1 élément	01/05/08 15:52	01/05/08 15:52

Renommer | Couper | Copier | Supprimer

Navigation

[racine]

Ajouter :

Catalogue d'index
Identificateur par cookie
Fournisseur de préférences par défaut
Utilitaire d'enregistrement des erreurs
Site Eztranet
Fichier
Dossier
Image
Permission
Conteneur de sessions persistantes
Utilitaire d'authentification
Utilitaire d'annotation des Utilisateurs
Conteneur de sessions en RAM
Rôle
Dossier de Gestionnaire de Site
Domaine de traduction
Générateur

Contenu | Aperçu | Métadonnées | Introspection | Inscription

Gérer le Site | Gérer le processus | Ajouter | Chercher | Autoriser | Erreurs | Aide

Introspection de l'objet : zope.app.folder.folder.Folder (*<aucun nom>*)

Interfaces fournies directement

- zope.location.interfaces.ISite
- zope.app.folder.interfaces.IRootFolder

Interfaces fournies

- zope.app.folder.interfaces.IFolder
- persistent.interfaces.IPersistent
- zope.location.interfaces.IPossibleSite
- zope.app.container.interfaces.IContained

Classes de base

- persistent.Persistent
- zope.app.component.site.SiteManagerContainer
- zope.app.container.contained.Contained

Attributs/Propriétés

- **`data`** (type : `OOBTree`)
  Valeur : `<BTrees.OOBTree.OOBTree object at 0xb700ddac>`
  Permissions : non disponible (lecture), non disponible (écriture)

Navigation

[racine]

Ajouter :

Catalogue d'index
Identificateur par cookie
Fournisseur de préférences par défaut
Utilitaire d'enregistrement des erreurs
Site Eztranet
Fichier
Dossier
Image
Permission
Conteneur de sessions persistantes
Utilitaire d'authentification
Utilitaire d'annotation des Utilisateurs
Conteneur de sessions en RAM
Rôle
Dossier de Gestionnaire de Site
Domaine de traduction
Générateur

Inscription d'un objet « Folder »

*Interface fournie: zope.location.interfaces.ISite

Nom d'inscription:

Commentaire:

Inscrire

Deprecation API

When we started working on Zope 3.1, we noticed that the hardest part of the development process was to ensure backward-compatibility and correctly mark deprecated modules, classes, functions, methods and properties. This module provides a simple function called *deprecated(names, reason)* to deprecate the previously mentioned Python objects.

Deprecating objects inside a module

Let's start with a demonstration of deprecating any name inside a module. To demonstrate the functionality, I have placed the following code inside the *tests.py* file of this package:

> from zope.deprecation import deprecated demo1 = 1 deprecated('demo1', 'demo1 is no more.')
>
> demo2 = 2 deprecated('demo2', 'demo2 is no more.')
>
> demo3 = 3 deprecated('demo3', 'demo3 is no more.')

The first argument to the *deprecated()* function is a list of names that should be declared deprecated. If the first argument is a string, it is interpreted as one name. The second argument is the reason the particular name has been deprecated. It is good practice to also list the version in which the name will be removed completely.

Let's now see how the deprecation warnings are displayed.

```
>>> from zope.deprecation import tests
>>> tests.demo1
From tests.py's showwarning():
...README.txt:1: DeprecationWarning: demo1: demo1 is no more.
...
1

>>> import zope.deprecation.tests
>>> zope.deprecation.tests.demo2
From tests.py's showwarning():
...README.txt:1: DeprecationWarning: demo2: demo2 is no more.
...
2
```

Navigateur de code Zope 3

[racine]

Zope 3 root.

- BTrees
- RestrictedPython
- ZConfig
- ZODB
- eztranet
- persistent
- transaction
- zc
- zdaemon
- zope

Zope 3 powered

- Launchpad (Canonical / Ubuntu)
- Schooltool
- CMS
  - Hivurt
  - z3ext
  - (Plone 3.x)
- Sites web
  - kelpi.com
  - relief.fr
- Applications
  - epidemio SOS Médecins

Documentation / ressources

- Livre
  - Web Component Development with Zope 3 (Ph. von Weitershausen)
  - Zope Component Architecture (Baiju M)
- Web
  - Nouveau site zope.org (bientôt ?)
- Communauté francophone
  - Liste AFPY : zope3-french-user
  - IRC : #zope3-fr sur freenode
- Communauté internationale
  - Liste zope3-users@zope.org
  - Liste zope-dev@zope.org
  - IRC : #zope3-dev


Merci !