PlonePAS Presentation

J. Cameron Cooper

Who am I?

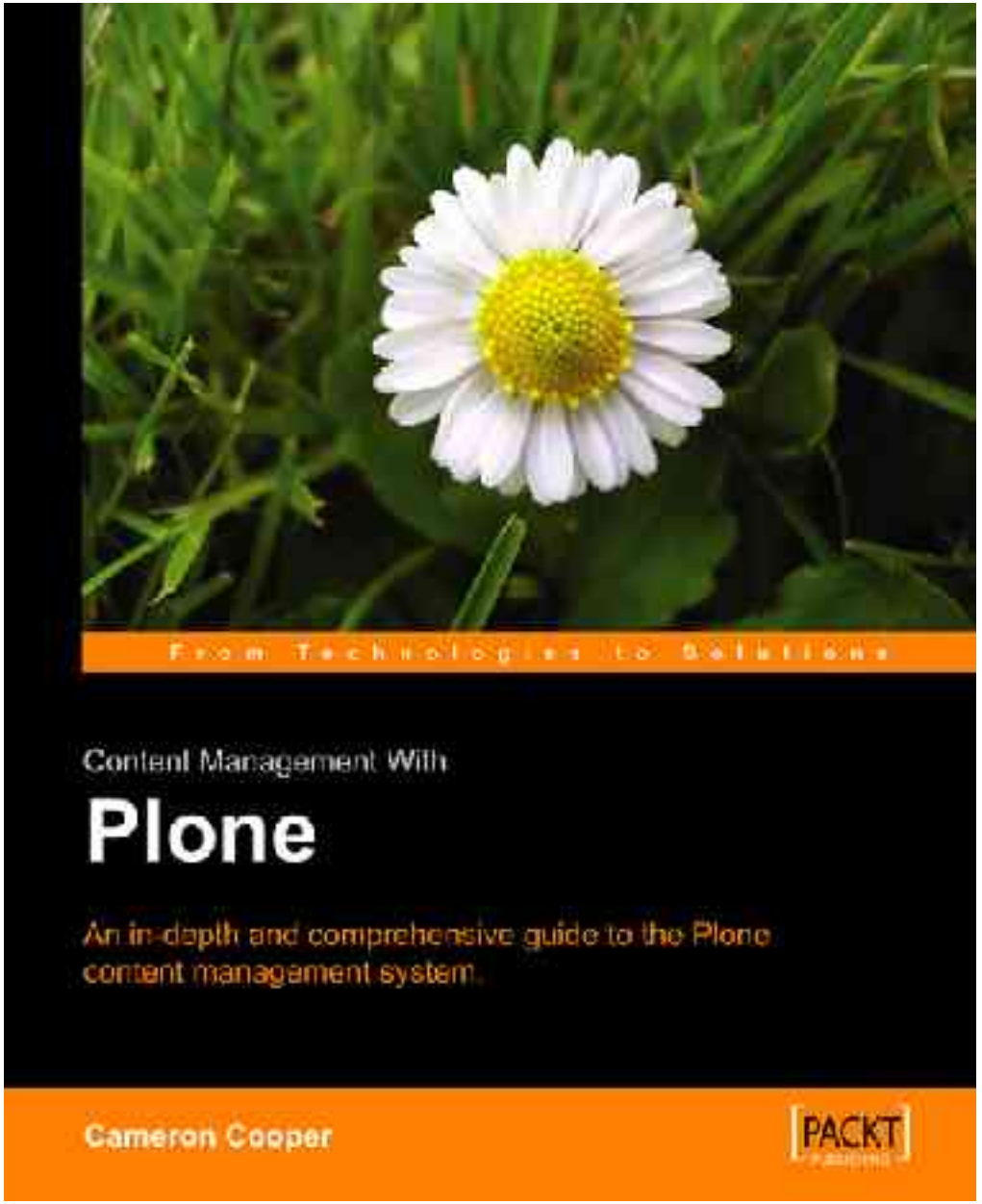
enfoldsystems

Connexions Project

Plone: Groups tools, etc.

Enfold Systems

enfoldsystems



enfoldsystems

What is PAS?

- PluggableAuthService
- UserFolder
- Comes from Zope Corp.

enfoldsystems

What does it do?

- Most things a regular UserFolder does, except through plugins
- All UserFolder functionality is pluggable
  - everything, and I do mean everything, is a plugin
  - provides some default plugins
  - you can find or write plugins to make it behave differently

enfoldsystems

Advantages

- Flexible administration
- Simple, well-defined extension
- Less magic
- Growing plugin population
  - At critical mass yet? I don't know.

enfoldsystems

Disadvantages

- Fairly new
- Loose coupling makes certain things tricky
- Some code paths inside PAS itself are tricky

Default Plugins Interfaces

- Extraction
- Authentication
- Challenge
- Update Credentials
- Reset Credentials
- User Factory
- Anon. User Factory
- Properties
- Groups
- Roles
- Update
- Validation
- User Enumeration
- Role Enumeration
- Role Assigner
- et al.

Zooming in...

- Let's look at Properties plugin...

```
class IPropertiesPlugin( Interface ):

    """ Return a property set for a user.
    """

    def getPropertiesForUser( user, request=None ):

        """ user -> {}

        o User will implement IPropertiedUser.

        o Plugin may scribble on the user, if needed (but must still
          return a mapping, even if empty).

        o May assign properties based on values in the REQUEST object, if
          present
        """
```

enfoldsystems

Usual Plugins

- ZODB based storage
  - Users
  - Groups
  - Roles
  - Properties
- Auth
  - Cookie
  - HTTP Basic
- Nothing special...

enfoldsystems

Other Plugins

- LDAPMultiPlugins
- Zope.org LDAP PAS
- GRUFMultiPlugin (PlonePAS)
- Delegating MultiPlugin

enfoldsystems

This is a Plone event!

- What about Plone?
- PAS is not drop-in for Plone
  - not your usual UserFolder
  - some missing capabilities, methods
  - GRUF API
  - etc.
- Sounds like trouble...
  - but now we have PlonePAS

enfoldsystems

PlonePAS History

- Started at CIGNEX San Jose Sprint
  - design/architecture
  - base code
- Enfold for the last few months
  - who? Me.
  - web interface, migration, etc.

enfoldsystems

What's in PlonePAS?

- Read/write ZODB plugins, and others
  - some new plugin types
- Migration of some standard UF setups
- Groups as objects
- Methods missing in PAS: GRUF, etc
- Mods to Plone UI
  - capabilities!
- Some experimental stuff: GRUFMultiPlugin

What's not in PlonePAS?

- Migration of your crazy UF setup.
- Non-basic plugins
  - Find elsewhere, or write your own
- Cleartext passwords
  - PasswordResetTool needed
- Bugs!
  - just kidding.

enfoldsystems

(Plone)PAS Catches

- Dependencies
- Caching
- Memberdata schema changes
- No fallback to Basic Auth
- May not be everything GRUF does
- Migration, of course
- Also, see docs

enfoldsystems

Versions

- Plone PAS is only tested on
  - Plone 2.0.5 (though 2.1 shouldn't be too hard)
  - GRUF 2.1 or GRUF 3.2
  - PluggableAuthService 1.0.4 (unreleased)
    - Plugin Registry
  - Latest LDAPMultiPlugin (possibly unreleased)
    - id mangling has changed in PAS

Installation

- Download from plone.org, Enfold, or Sourceforge.
- Make a backup!
- No, seriously, make a backup.
  - And don't do this on your production machine!
- Follow the usual procedure
  - does anyone not know this?
  - the README deals with this

Migration

- Automatic when using QuickInstaller
  - see install log for details
  - but only for certain setups
- Migratable setups:
  - Default
  - LDAPUserFolder as user source
  - other UFs might work...
    - but with no special features

Manual Migration

- Find/write plugins for your external source
  - configure the same as original
  - this is how it works with LDAPUF
- For actual migrations:
  - run script to store data
  - delete existing UF
  - install PlonePAS; set up plugins if needed
  - run script to add users
  - this is how ZODB user migration works.

Future of Migration

- PlonePAS may eventually have better hooks for migrations.
- Probably write pre and post scripts, and register somehow.

Now what?

- Configure, if needed
  - add plugins
  - turn off, rearrange plugins
  - write plugins, even
- Use like a regular UF
  - though the interface is different than you may be used to

PAS ZMI Interface

- Go to acl_users, of course
- You will see all the plugins installed
  - all types are available in pull-down
- Click plugin object for activation and config
  - Some have more config than others
- Default plugins let you define users/groups, assign roles, et al.
- The *plugins* object is special...

ZMI User Management

- source_users: create/delete users, set pass.
- source_groups: same, with groups
- portal_role_manager: set global roles to manage, assign to users/groups
- credentials_basic_auth, credentials_cookie_auth: auth sources
- mutable_properties: contain user props

enfoldsystems

Plugin Management

- The *plugins* object lists all available interfaces
  - drill down to see all plugins implementing those interfaces
- A plugin can implement one or many
  - if it does most, it might be called a multi-plugin
- In each interface, all available plugins can be added, removed, and ordered
- ...so a plugin can be installed, but not used

Plugin Management Example

- Say we don't want cookie auth anymore
- PAS provides an auth plugin using cookies
  - no more CookieCrumbler
- There's also an HTTP Basic Auth plugin
  - make sure there's an instance of this
- In the auth plugin detail screen:
  - remove cookie auth plugin from the active list
  - add the basic auth plugin

enfoldsystems

The Plone Interface

- PAS provides for user management that looks little different from standard Plone
- Plone Setup, Users/Groups Management
- You'll see little difference (except perhaps that global roles can be set on users)
  - but this is an adaptation to fit PAS into the existing code
- ...but when you start changing things up, you'll notice a difference.

enfoldsystems

Capabilities

- If you get users, groups, roles, properties, etc from a read-only source, the control is read-only.
- I call this capabilities checking. It avoids providing a false appearance of control.
- This can be a problem in regular Plone, but is especially accute in PAS.
- Plugins must implement certain interfaces to declare themselves non-readonly.

Setting up LDAPMultiPlugin

- No LDAP around, unfortunately, but we can see the steps.
  - Trust me, it works.
- Get LDAPUserFolder and LDAPMultiPlugins
  - The ADSI versions will also work
- Create a plugin (LDAP or AD) in PAS folder, and configure it like your LDAP
- If you already have an LDAPUF, PlonePAS can migrate it automatically.

LDAP Config

- Let's look at the LDAP plugins...
  - **Activate** tab shows lots of interfaces
    - we'll remove groups
  - **Contents** tab shows an LDAFUF
  - **Properties** tab has some config
- Plugin config
  - in plugins object > properties plugin
  - add LDAP MP
  - order to top

enfoldsystems

Looking forward

- Should see more plugins
  - if you write 'em, release 'em
- Better migration framework
- Some upstream insertion
  - Plone should carry groups tool base
- PLIP 102 to use PAS as main UF
  - replaces GRUF... but GRUF won't go away anytime soon
  - based on PlonePAS

enfoldsystems

Remaining questions

- UI is problematic, even with capabilities
  - it assumes monolithic UF
    - PAS causes problems in this paradigm
  - Replace existing Plone UI with more pluginish UI
    - But what does this really look like?
- Is PAS really fast/stable/clear enough?
  - It's fairly new, we don't really know.
- Can installed base cope?
  - how long do we want GRUF-compat?

That's all, folks...

- Any questions?