Delegating Samba Administration

Jeremy Allison
Novell, Inc.

May 5, 2006

samba

Novell®

Why Samba needs to change

- Windows allows delegation of Administration tasks by allowing groups to be assigned “capabilities”.
  - Example : “Share Directories” capability is assigned to “Administrators” and “Server Operators” by default.
- Windows administrators are used to being able to create shares by navigating to a directory and selecting “share this folder” from a menu.
- Each capability is independently assignable to a user or group.
  - The “root” concept is effectively split into multiple roles.

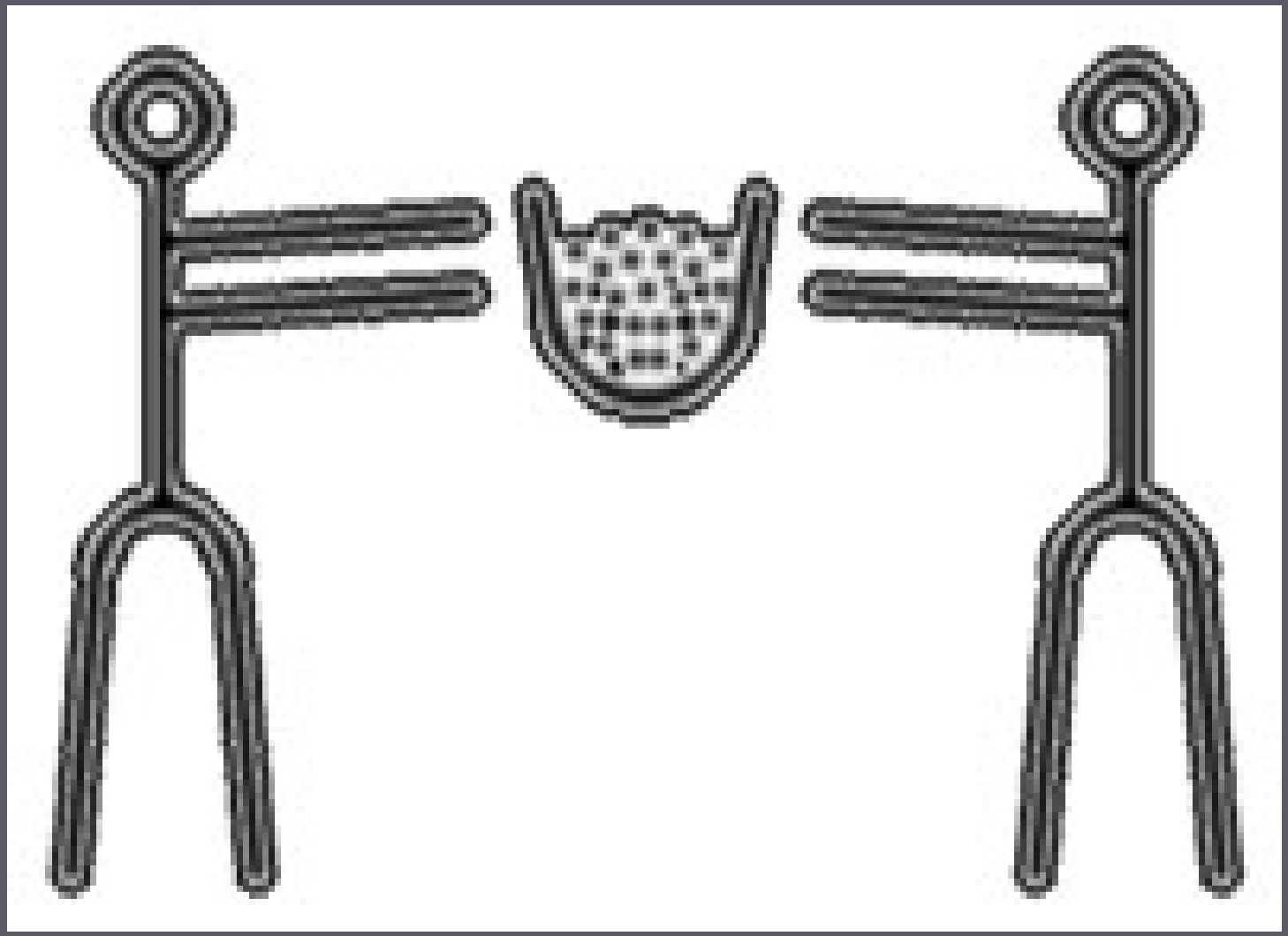
How Samba needs to change

- Samba needs to become easy for Windows trained Administrators to configure and use.
- One way to do this is to improve the RPC support on the server so Windows administration tools will configure Samba servers.
- This work is currently in progress, however it requires some considerable set-up on the server side.
  - Groups need to be mapped into the Samba group database.
  - Privileges need to be assigned on a group basis.
  - Complex scripts are needed to manipulate smb.conf and the underlying UNIX databases.

How Samba needs to change (continued)

- One underlying problem is that any process that can write to smb.conf can become root on the system.
- In addition, only a file/directory owner (or root of course) can change permissions in the filesystem.
  - Windows allows group ownership of directories without a user owner, so gets around this problem.
  - This simple change allows a very powerful feature – delegation of permissions administration to a non-administrative group.
- We need to enable the “delegate and forget” administration model.
  - Give a group the rights to create shares and administer the permissions within them.

Presenting “usershares”



What are usershares ?

- A usershare is a small text (*) file that exists in a special directory (usually within the Samba config tree) that defines a share.
  - As with all good UNIX ideas, it can be edited with “vi” ☺
- Deliberately designed not to be parsed as part of smb.conf.
  - Very truncated set of verbs – no “force user” allowed.
- Designed to be created locally by non-root users.
  - smbd must defend itself from malicious file links and malicious file contents.
- Allows creation of shares to be delegated to a UNIX group.

A sample usershare file

```
#VERSION 1
path=/home/jeremy
comment=This is a usershare
usershare_acl=S-1-1-0:R
```

A sample usershare file

- The first line allows file revisions (we'll never make *that* mistake again).
- The “path” and “comment” lines are identical to the way they're used in the standard smb.conf.
  - The “path” parameter must be absolute (ie. start with “/”)
- The “usershare_acl” entry specifies the ACL associated with the share.
  - Permitted values are “R” for read-only, and “F” for full access.
  - SID is used instead of UNIX uid/gid so smbd doesn't have to do uid -> SID mapping on parsing the file.
- Should we allow “guest ok = yes” ?

But “is it safe” ?

Protecting smbd

- In order for smbd to read a user-writable directory for configuration information it has to take many precautions.
- The directory containing the usershare files could have been filled with millions of bogus usershare files, leading to a DOS attack.
- The usershare files might by fifo's or device files that would cause smbd to hang on reading them.
- The usershare files might by symlinks pointing at such things elsewhere on the filesystem.
- A usershare file might be gigabytes in size, leading to a memory DOS.

The smb.conf usershare parameters

- “usershare path” points at the directory on the system containing the usershare files. It must have special characteristics.
  - Must be owned by “root”, have the “t” bit set on its permissions, and not be writable by “other”.
  - The “root” ownership means only root can have arbitrary access to it.
  - The “t” bit prevents different users overwriting others usershare files (only the creator can modify or delete them).
  - Only allowing root and the group owner to write into the directory allows administrator control of who can create shares (only members of the owning group).

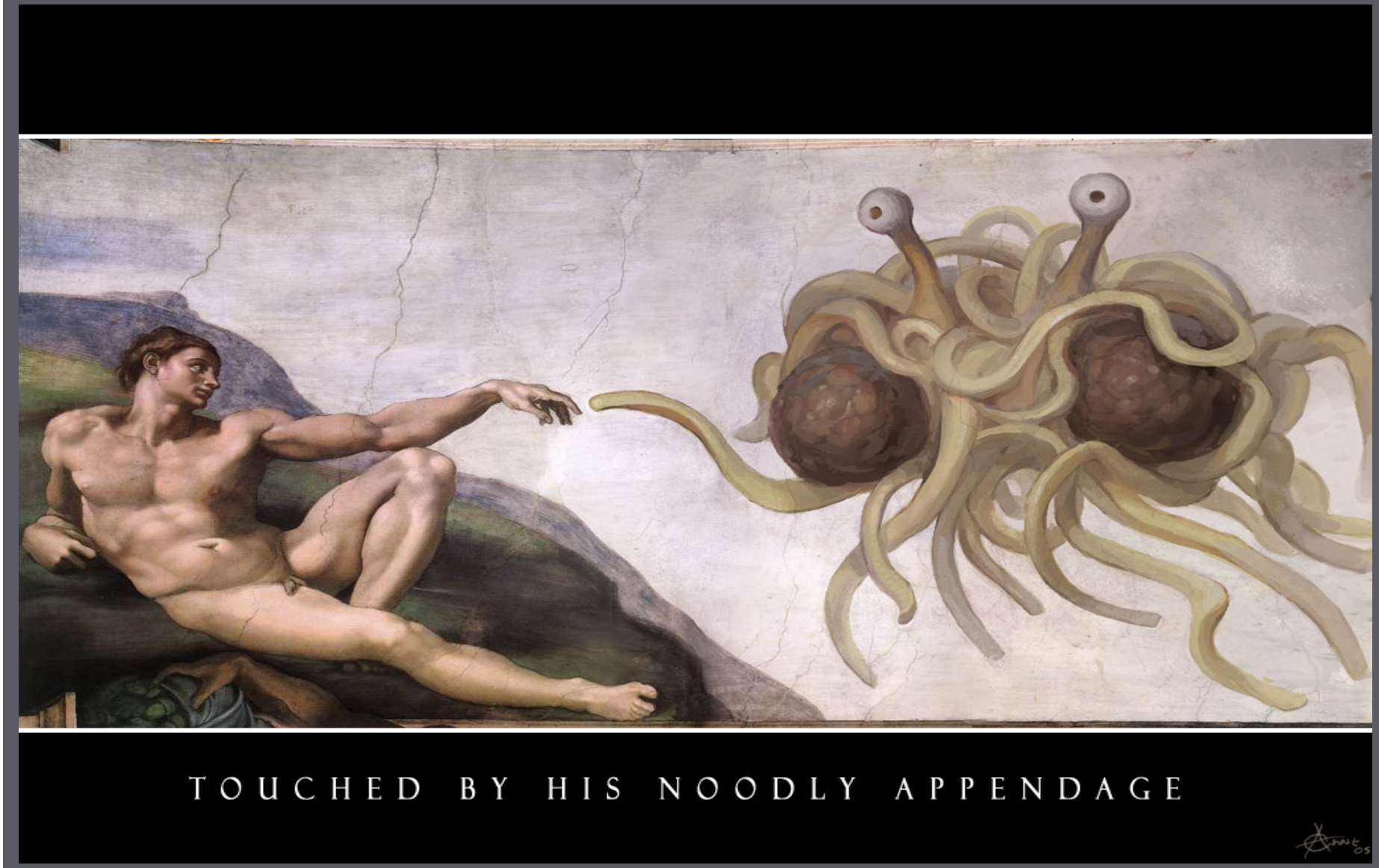
More smb.conf usershare parameters

- “usershare max shares” is smbd's attempt to protect itself from a “too many files in the directory” DOS.
  - Until you fire the person who did it ☺
- smbd allows 20% of “usershare max shares” entries in the usershare directory to be invalid, and after that stops processing the directory.
  - Set “usershare max shares” to the number of usershares you want to allow and don't worry about the DOS attacks until users complain.
- “usershare owner only” if set causes smbd to check if the directory being shared is the same as the creator of the usershare file, and ignore it if not.
  - Allows owner based restrictions on what can be shared.

The final parameters

- "usershare prefix [allow|deny] list" are a standard smb.conf list parameter which restrict the paths being shared in a usershare to either start with, or to be prohibited from starting with, the given absolute path.
  - Phew – too hard to explain – here's an example :
  - “usershare prefix deny list = /etc,/dev,/tmp”
  - “usershare prefix allow list = /home, /data, /space”
- “usershare template share” causes all created usershares to become copies of the given share (already defined in smb.conf).
  - Allows extra parameters to be set (eg. “guest ok = yes”).
  - If you don't want the template to be a real share mark it as “-valid = False”.

How do I create a usershare ?



First set up your smb.conf

- Add “usershare path = /etc/samba/usershares”
- Create /etc/samba/usershares and change the group owner to the group who should have the ability to create usershares (eg. serverops).
- Set the permissions to be “01770”.
- Tell smbd how many usershares you will allow by adding “usershare max shares = 100”.
- Now you need some usershares to be created....

New “net” command options.

% bin/net usershare help

Usage:

net usershare add <sharename> <path> [<comment>] [<acl>] to add or change a user defined share.

net usershare delete <sharename> to delete a user defined share.

net usershare info [-l|--long] [wildcard sharename] to print info about a user defined share.

net usershare list [-l|--long] [wildcard sharename] to list user defined shares.

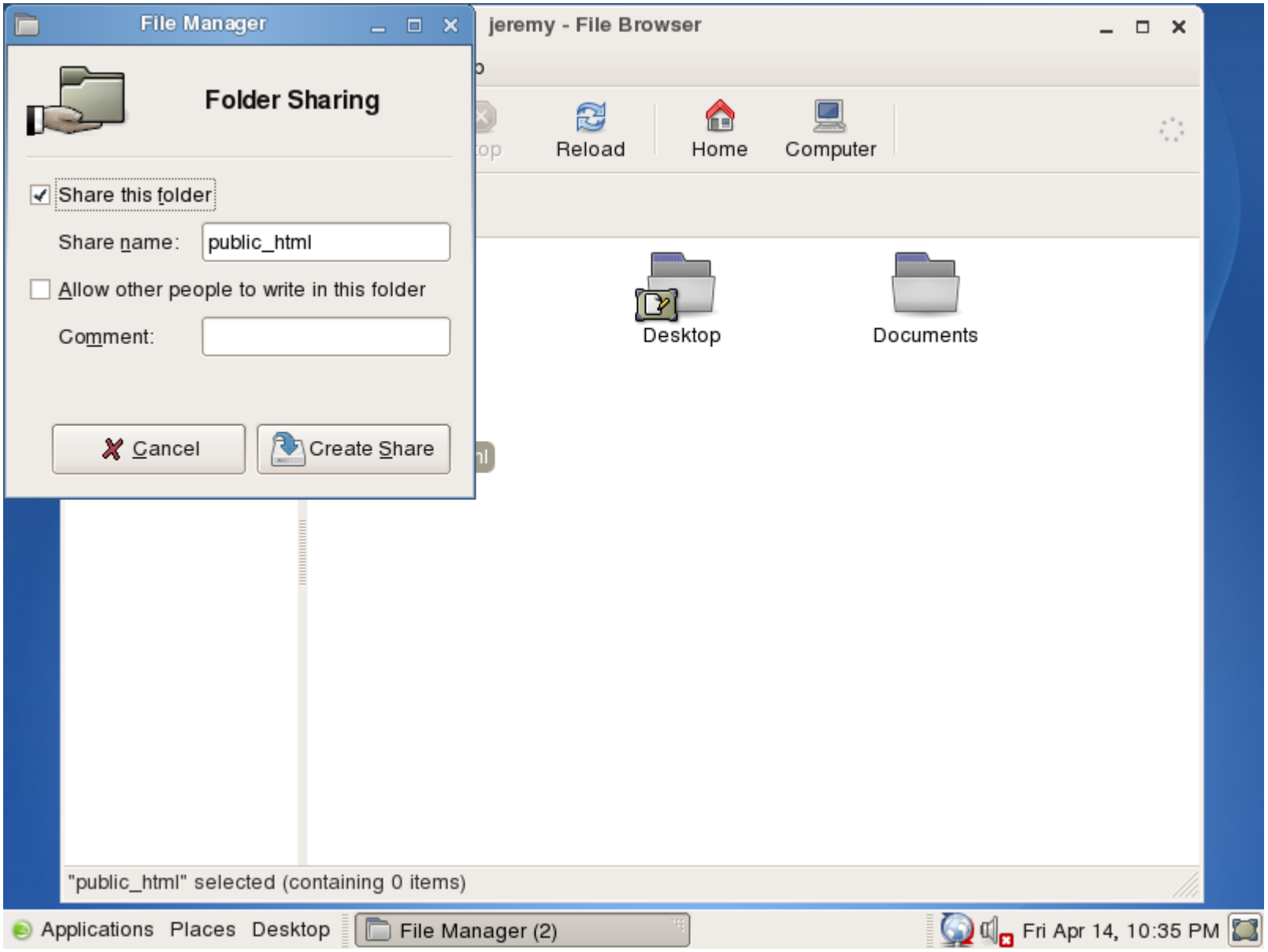
- net usershare help

A brief demo.....

% net usershare add jrahome /home/jra

Gnome and KDE Integration

- Support has been added to Nautilus (Gnome) and Konqueror (KDE) to right click on a directory and select “share this folder”.
  - Invokes “net usershare” command underneath the covers.
- NASTY BUG: net command is not internationalized.....
  - Samba user toolset needs gettext() support urgently.
- Windows “*Administrators”* will finally feel at home.
  - Well the ones that need GUI support anyway ☺.


File Manager
Folder Sharing
Share this folder
Share name:
public_html
Allow other people to write in this folder
Comment:
Cancel
Create Share
jeremy - File Browser
Reload
Home
Computer
Desktop
Documents
"public_html" selected (containing 0 items)
Applications
Places
Desktop
File Manager (2)
Fri Apr 14, 10:35 PM

Delegating group permission changes

- In Windows, a common use case is to create a share for a group of users, and make it owned by that group.
  - The inheritance model is that all files created within that share inherit the group ownership.
  - Becomes a “shared space” for that group.
  - No Administrator support needed to set ownership/permissions.
- Samba can do the same thing.
  - Not integrated with “share creation” dialog unfortunately.

Setting up a share with delegated permissions

- Create the directory (eg. /data/foo), change the group ownership to the allowed group (foo).
  - Set the directory permissions to : 2770 (setgid bit, owner and group all permissions).
  - Create the smb.conf stanza as follows :

```
[foo]
        path = /data/foo
        acl group control = yes
        dos filemode = yes
```

(acl group control is now deprecated).

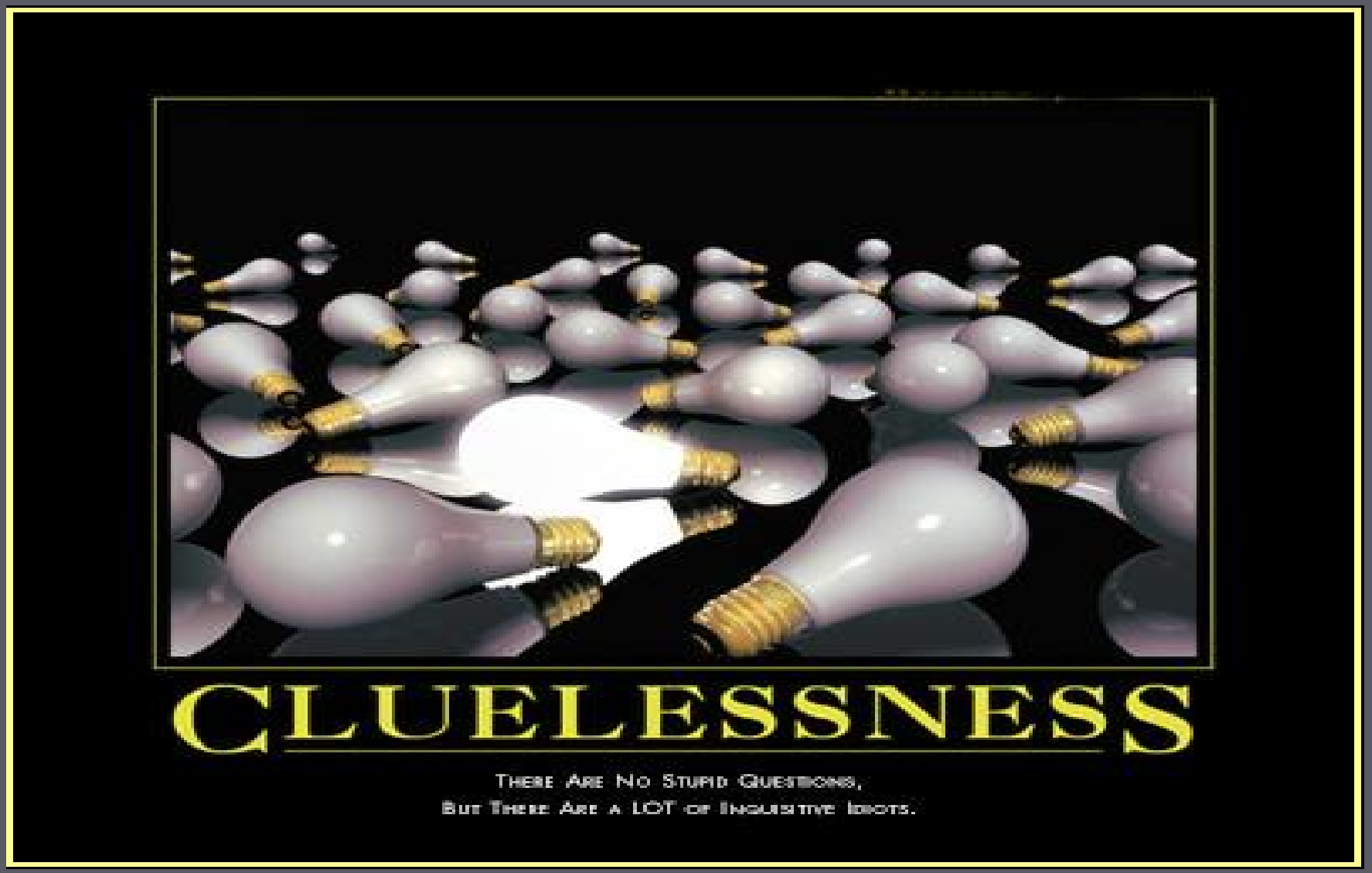
Setting up a share with delegated permissions (continued)

- The SETGID bit causes all files and directories created within that directory to have an owning group identical to the containing directory (so it propagetes down the tree).
  - Created directories also propagate the SETGID bit.
- The “acl group control” or “dos filemode” parameters allow smbd to override POSIX permissions to allow any connecting user in the owning group to modify the ACL permissions on the file or directory.

Conclusions

- We can get closer to an “ease of use” Windows style of administration for Samba.
  - Need more glue code to make everything seamless.
  - We need to do this to move Samba forward – we already have all the early adoptors.
- “usershares” could already be done via clever enough scripting of “add/delete share command”
  - No one has shipped something that automates this in a way as easy to use as usershares.
- More “wizard” like functionality needed (not necessarily from Samba) to make setting up shares easier.

Questions and Comments ?




Novell®