Samba Management Console
Agenda

- **Introduction**
  - Overview
  - Goals

- **Architecture**
  - Design
  - Components
  - Functions

- **Demo (I hope)**
  - Interface
  - System Integration

- **Roadmap**
  - Status
  - Next Step
Introduction: Samba usage

**Small env**
- Domestic Storage NAS
- All in one Appliance

**XL env**
- No i386 hw
- Heterogeneous env
- High performance FS

**Medium env**
- Fanatic
- Economic
Why middle and large companies don’t want to use Samba?

- **What is Samba?**
- **I use only MS products**
- **Samba is too complicated**
- **Samba is only for N...**
- **I have a big discount on MS products...**
- **My applications need MS**
- **Total cost of ownership**
- **I need ADS**
- **Where can I buy support?**
- **I’m not a programmer**
- ...
Introduction: Problems

- TCO
- Samba is too complicated
- Samba is only for Hackers

Management
Introduction: Idea

Samba Management Console!
Introduction: Project Goals

Simple
- Graphical user interface
- Handle Samba configuration
- Handle Samba Status (Process, session management)
- Handle Samba Users/Groups
- Setup

Open
- Process Integration/Automation with other systems
- Work with samba PDC/AD
- Work with windows AD

Global view
- Control all servers from a central point
- Handle groups of servers as one
- Collect information from different servers
Architecture: Design

Rich Client

Node Collector
Scheduler
Web Int
Cache

Node Controller
Proxy
Local
Architecture: Design

**Simple**
- Web Ajax
- Embedded Server
- Embedded DB

**Open**
- REST
- ExtJS
- Python
- Sqllite

**Global view**
- Centralized GUI
- Collector
- Proxy

- Samba Libs
- Cmd
- Swig
Representational State Transfer - REST

The most important concept in Rest concerns the RESOURCES (source of specific information), each of which is referenced with a global identifier

e.g. http://www.boeing.com/aircraft/747

REST Web Services Characteristics

- Client-Server: a pull-based interaction style: consuming components pulls representations.
- Stateless: each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server.
- Cache: to improve network efficiency responses must be capable of being labeled as cacheable or non-cacheable.
- Uniform interface: all resources are accessed with a generic interface (e.g., HTTP GET, POST, PUT, DELETE).
- Named resources - the system is comprised of resources which are named using a URL.
- Interconnected resource representations - the representations of the resources are interconnected using URLs, thereby enabling a client to progress from one state to another.
- Layered components - intermediaries, such as proxy servers, cache servers, gateways, etc, can be inserted between clients and resources to support performance, security, etc.

REST – is An Architectural Style, Not a Standard!
Why Python?

- see Jelmer Vernooij sambaxp 2008
- highly scalable,
- suitable for large projects as well as small ones
- rapid development
- portable cross-platform
- embeddable easily
- extensible object-oriented
- elegant, stable and mature
- powerful standard libs wealth of 3rd party packages
ExtJS is a cross-browser JavaScript library for building rich internet applications.

- High performance
- Customizable UI widgets
- Extensible Component model
- Leader in Javascript interface
- Commercial and Open Source licenses available
Model-View-Controller

- Independent on how retrieve informations
  - RPC
  - Command Line
  - JSON/REST
  - Cache
  - Python

- Different output format and Extensible
  - HTML
  - JSON
  - CSV
Demo : functions

Servers
- Global configuration
- Process Control
- Backup/Restore
- Logs

Share
- Shares
- Printers

Identity
- Users
- Groups
- Workstations

Status
- Sessions
- Files
- CPU
- Queue
Architecture : setup

- **Install**
  - Satisfy requirements (python, samba)
  - Uncompress the tarball

- **Configuration**
  - Define controlled server
  - Define samba bin dir
  - Define SMC users
  - Define http port

- **Run**
  - Smcd -f configuration.file

- **Use**
  - Point your browser to the server

```python
[global]
port=8080
smb_dir=/opt/samba

[users]
users=admin, pippo

[servers]
servers=mysrv

[user_admin]
password=admin
role=admin

[server_mysrv]
ip=127.0.0.1
name=myserver
admin=Administrator
password=domain
domain=domain
os=linux
samba=3.2

Python smcd.py -f smcd.conf -d
```
Proof of concept
Demo: Integration

URL /rest/type/resource/id

- **Type**
  - Global, Identity, Status

- **Resources**
  - Server, share, user, ...

- **Id (optional)**
  - Resource Identification

- **Operations**
  - **Get**: list elements/attributes in resource
  - **Post**: create new resource
  - **Put**: update resource
  - **Delete**: remove resource
The Rest interface gives all the functions and information present in the web client

- Retrieve information with simple url
- Interconnect to monitoring system
- User/group provisioning
- User Administration (password, lock..)
Demo: Integration
Beolink.org

Advantages

- Suitable for large env as well as small ones
- Portable cross-platform
- Embeddable easily
- Simple integration and install
- Total control with simple dashboard
- Interface for Automation process
- GPL
Status

Requirements

- > samba 3.2
- > python 2.6
- Registry for RW operations

More prototype than product

- Status section completed
- Users / groups readonly sections
- Most Configuration section is ro
- Centralized cache not implemented
Not clear yet but ..

- Better integration
  - smcncd → integrated in Swat
  - python Samba library (swig)

- Cache for speed performance in large env (sqlite)

- Django for smcclld

- Operations on all nodes, single view

- Better support for different samba releases

- Cluster ctdb

- Upgrade software (deploy)

- Adobe Air

- User integration

- Web Configuration
I look forward to meeting you…

XVII European AFS meeting 2010
PILSEN - CZECH REPUBLIC
September 28-30

Who should attend:
- Everyone interested in deploying a globally accessible file system
- Everyone interested in learning more about real world usage of Kerberos authentication in single realm and federated single sign-on environments
- Everyone who wants to share their knowledge and experience with other members of the AFS and Kerberos communities
- Everyone who wants to find out the latest developments affecting AFS and Kerberos

Thank you

manfred@freemails.ch
manfred@zeropiù.it

www.beolink.org/smc