



# Mac OS X

## Directory Services

# Agenda

- Open Directory
  - Mac OS X client access
  - Directory services in Mac OS X Server
  - Redundancy and replication
- Mac OS X access to other directory services
- Active Directory support





# Open Directory

# Mac OS X Server Strategy

**Industry standard**



**Open source**



**Innovation**



# What is Open Directory?

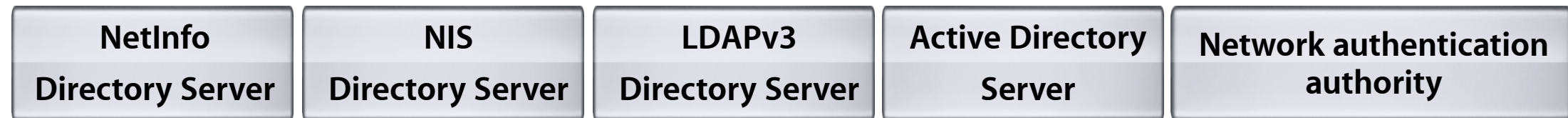
- Open Directory is a technology name
  - Includes both Client Access Technologies and Server Technologies
  - Adopts and promotes Industry Standard technologies
- Open Directory is built into Mac OS X and Mac OS X Server
  - Part of the OS since 10.0
- Open Sourced as part of Darwin
  - <http://developer.apple.com/darwin/projects/opendirectory/>



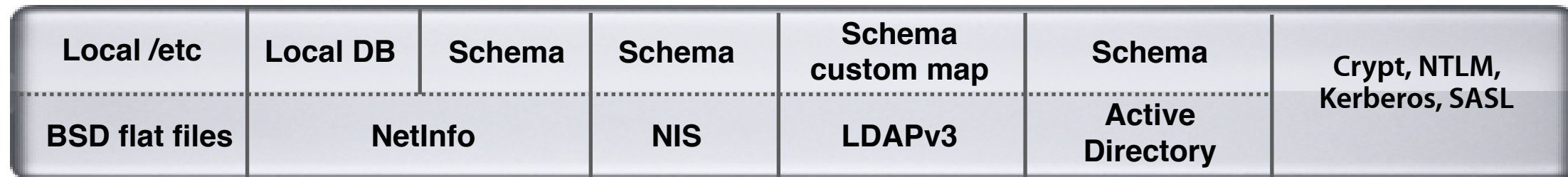
# Open Directory Architecture

Built into every Mac

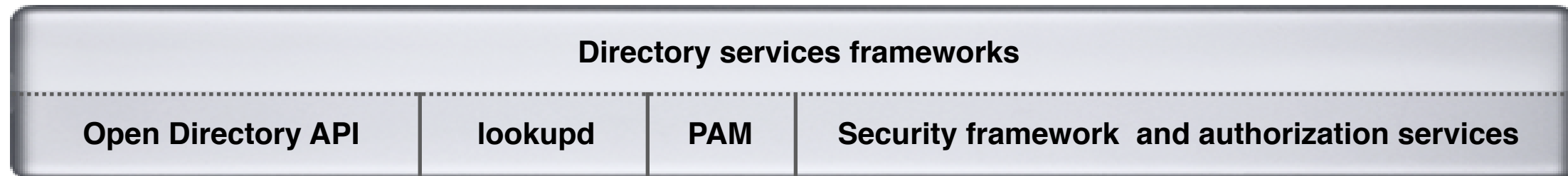
Network directory and authentication servers



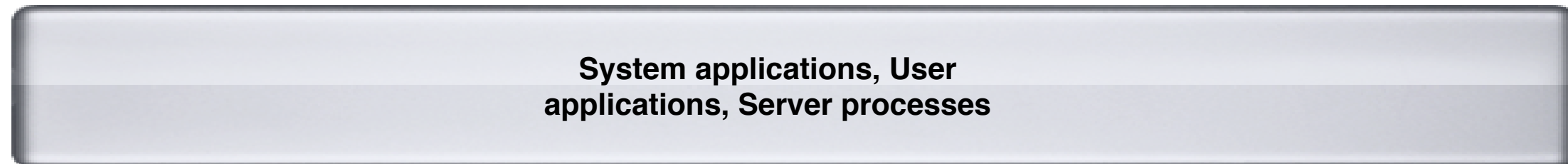
Directory access and authentication



Directory and authentication frameworks

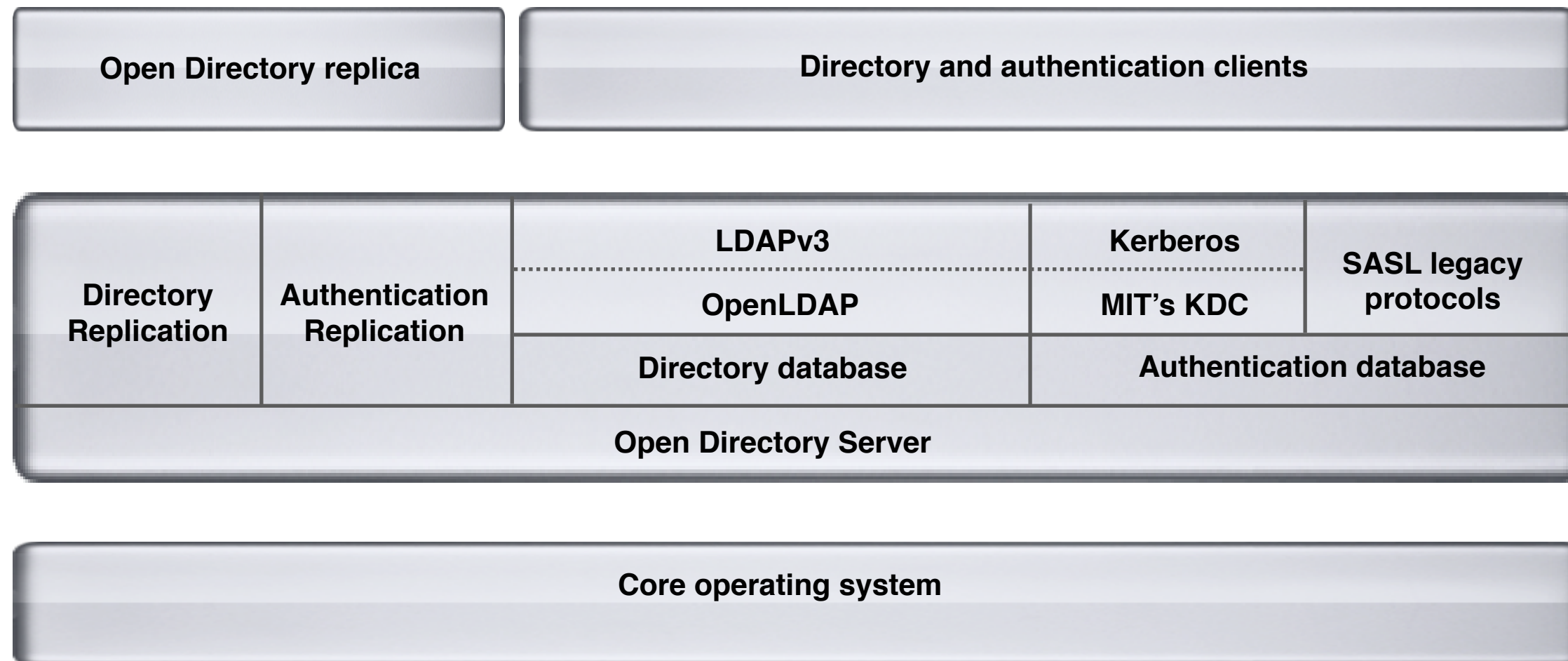


User applications, system functions, and server processes



# Open Directory

## Built into Mac OS X Server



# Directory Services



---

## **LDAP directory services**

Scalable, open standards-based directory infrastructure based on OpenLDAP and Berkeley DB.

---

## **Kerberos authentication**

Secure network-based authentication with support for single sign-on. SASL provides support for legacy password protocols.

---

## **Enterprise integration**

Integration with existing infrastructures using LDAP directory solutions such as eDirectory and Active Directory.

---

## **Legacy support**

Compatibility with NetInfo, NIS, and /etc directory services.

---







**To the Client**  
***“It Just Works”***



# **Automatically Configure Directory Services**

# Open Directory Zero-Conf

- Client sends DHCP request
- DHCP server provides IP information
- DHCP server also provides Option 95 data with location of assigned LDAP server
- Client checks the LDAP server and finds a Directory Access template (schema mapping)
- Client connects to the directory



# Automagic continued...

- Even before a user logs in, the workstation has access to important data it needs
- List of directory replicas
- List of sharepoints that host home directories, fonts, etc.
- Location of Password Server/KDC
- Pre-configured plist for the Kerberos client
- Managed client settings for the computer



# When the User Logs In

- The user is given a Kerberos TGT and allowed access to the computer
- Uses the TGT to get an AFP service ticket
- Uses the service ticket to mount the sharepoint hosting the user's home directory
- The user's managed client settings (individual & group) are enforced in addition to the computer settings loaded earlier
- Any login scripts or startup items are launched





# Open Directory at the Server

# Powerful Management Team

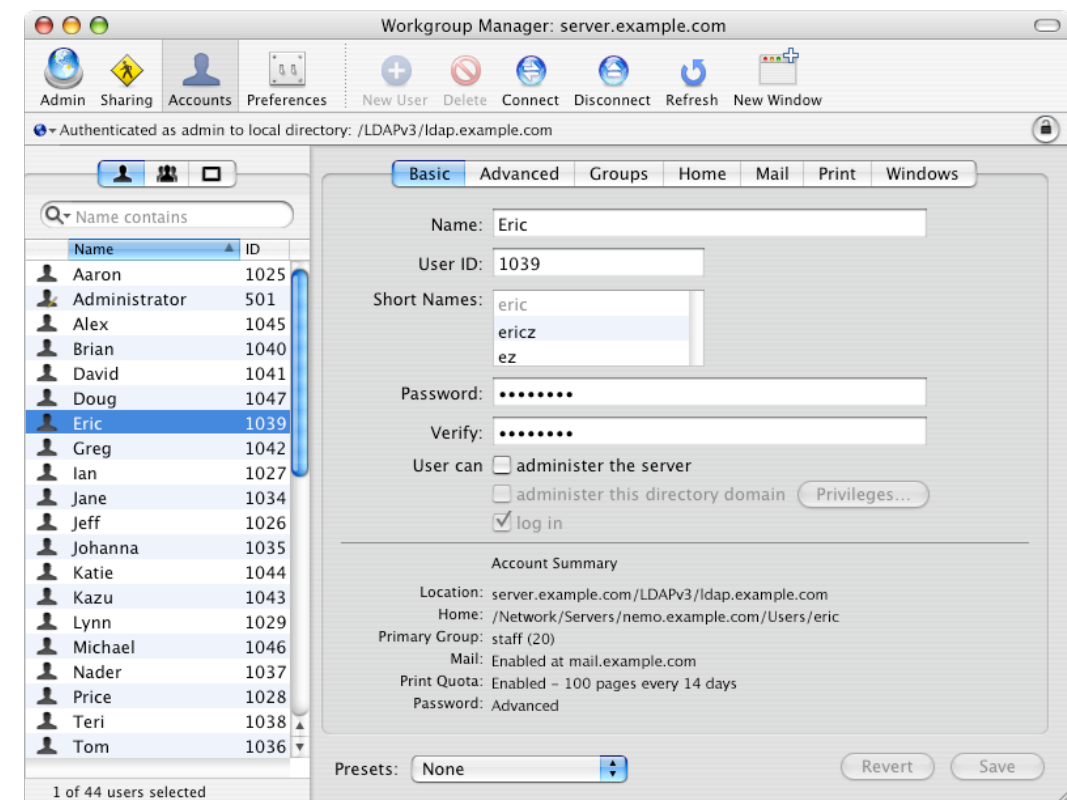


## Server Admin

Configure network services

Manage services on one or many servers

Monitor real-time logs and graphs



## Workgroup Manager

Set up user and group accounts

Define share points and access privileges

Manage settings for Mac computers



# What's in the Directory?

- User accounts
- Groups and group memberships
- Computer lists
- Sharepoints
- Printers





# HomeDirectory & NFSHomeDirectory

- Every Mac OS X client must have a Home Directory ...somewhere
- It can be:
  - Network-based on a server volume
  - Local on the client, /Users
  - Local on the client with but with access to a server for document storage
  - Temporary and local on the client



# User Home Directory Attributes

- NFSHomeDirectory stores the path to user homedir
  - Examples:
    - /Users/username*
    - /Network/Servers/server.apple.edu/Sharepoint/username*
- HomeDirectory stores the AFP url for the sharepoint in XML
  - Example:
    - `<home_dir><url>afp://server.apple.edu/Sharepoint</url><path>username</path></home_dir>`
  - If using NFS the HomeDirectory is either empty or stores the same data as NFSHomeDirectory



# Automounted Sharepoints

- RecordName or cn stores the server and sharepoint names
  - Example: server.apple.edu/Sharepoint
- VFSLinkDir stores the location in the file system where the server will be mounted
  - Example: /Networks/Servers/
- VFSType stores the protocol used to mount the sharepoint
  - Example: afp or nfs
- VFSOpts stores how to locate the sharepoint
  - For NFS it is a simple single-valued property
  - Example: net

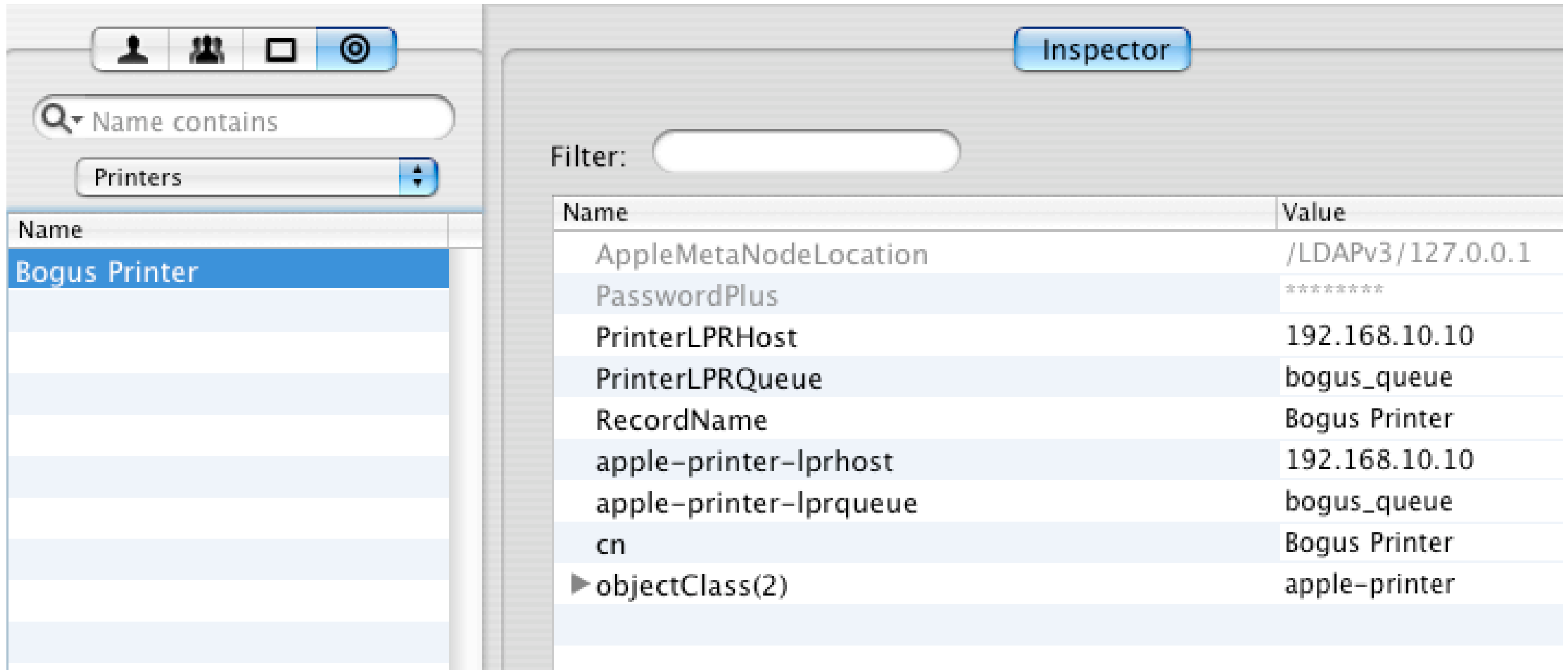


# Automounted Sharepoints

- VFSSOpts stores how to locate the sharepoint
  - AFP automounts have a more complex multi-valued property for this attribute. The first value is the same as NFS. The additional value provides the AFP url for the sharepoint and signals that no user authentication is required to connect to the automount
  - Example:  
net  
url==afp://;NO%20USER%20AUTHENT;server.apple.edu/Sharepoint



# Printers in Open Directory



The screenshot shows the Open Directory Admin tool interface. On the left, a search bar contains "Name contains" and a dropdown menu is set to "Printers". Below this, a list of entries is shown, with "Bogus Printer" selected. On the right, the "Inspector" pane displays the LDAP attributes for the selected entry in a table format.

Name	Value
AppleMetaNodeLocation	/LDAPv3/127.0.0.1
PasswordPlus	*****
PrinterLPRHost	192.168.10.10
PrinterLPRQueue	bogus_queue
RecordName	Bogus Printer
apple-printer-lprhost	192.168.10.10
apple-printer-lprqueue	bogus_queue
cn	Bogus Printer
▶ objectClass(2)	apple-printer



# Managed Client Settings

- Stored in the directory
- Can be assigned to individuals, groups or computer lists
- Some MCX settings are additive (what apps can be run)
- Other settings follow the hierarch: User, Computer, Group
- MCX settings are independent of the users home directory or the preference files store there
  
- Examples of MCX controls





# **Other Features of Workgroup Manager**



# Authentication



# Authentication Authority Matrix

Authentication Authority	Example	Comment
<code>;basic;</code>	<code>;basic;</code> or left empty	Crypt-based passwords stored in the user record in the password field
<code>;ApplePasswordServer;</code>	<code>;ApplePasswordServer;</code> <code>[ID][PublicKey]</code> <code>[Network Address]</code>	Used to indicate password server based user—no readable password hash
<code>;ShadowHash;</code>	<code>;ShadowHash;</code>	Local user account—password hash stored in <code>/var/db/shadow</code>
<code>;Kerberos;</code>	<code>;Kerberos;</code>	Used to indicated a Kerberos based user account



# Panther Shadow Hash

- Panther stores all passwords in `/var/db/shadow/`
  - Secure file system directory
  - Uses Sha-1, supports greater than 8 characters
  - System is no longer subject to “hash” download attacks
- Crypt is still supported, but primarily for legacy support
- Panther command line tools updated to use Directory Service API's for changing passwords





# Supporting Windows Clients

# Support Windows Clients

- Most services provided by Mac OS X Server are industry and internet standards like (POP, IMAP, HTTP, FTP, LPR, etc)
- Any Mac OS X Server can also provide SMB/CIFS file and print services
- Mac OS X Server can also join an NT authentication domain and provide services to users in that domain
- Additionally, Mac OS X Server can act as the PDC providing login authentication for Windows clients and user authentication for other services





# Open Directory Relationships

# Mac OS X Server can...

- Be independent with a local, stand-alone directory domain
- Bind/connect to a parent directory service
- Be a replica of an existing OD Master
- Host a shared directory as an OD Master



# Connect to an OD Master

- Servers can bind/connect to an OD Master and allow it to provide the directory and authentication services
- Those child servers offer file, print, web, mail, and other services to users in the directory
- Those servers will honor the Kerberos tickets
- The administrators of these servers can even create locally administered groups made up of users from the OD Master
- They can not change or modify the users themselves, unless they also have domain admin privileges



# Replicating Open Directory

- The LDAP directory is a single master structure
- Changes are made at the master and forwarded to replicas
- If the master is available, it appears that changes can be made at the replica--but they are actually forwarded to the master
- The Password Server & KDC use a multi-master structure, allowing users to change their passwords on any PWS/KDC
- Replication can be set to occur as changes are made, every X hours or immediately on demand.





# Failover in a Replicated Environment

- Each client caches a list of replicas whenever then connect
- If the assigned directory server does not respond, the client will request access from known replicas and work with the first to respond



# Managing Failover

- The default failover technique is designed to be simple and transparent to the user
- You can modify that behavior with a few basic steps
  - Use firewall to limit the clients/IP addresses a given server will support
  - Edit the replicas list to remove servers (like the Master)
  - Remove all servers from the replica list and rely on other techniques like DNS round robin.





# Some Notes on the Password Server

# Password Server

- Most authentication types require the Password Server to store on the hash (encrypted password).
  - CRAM-MD5, DHX, Digest-MD5, MS-CHAPv2, SMB-NT, SMB-LAN Manager
  - Also supported are APOP & WebDAV-Digest which require access to a clear text password
- Password Server Database stores:
  - 128 bit ID
  - Password hash (or clear text if required)
  - Users' short name (for messages/logs)
  - Password policy settings



# Open Directory Replication

- Panther Server can replicate:
  - LDAP Directory
  - Password Server
  - KDC
- Replication can:
  - Divvy the authentication load
  - Provide redundancy (fail-over)
  - Move directory closer to the clients, removing chances of delays/outages due to a WAN link



# Password Server/KDC Replication

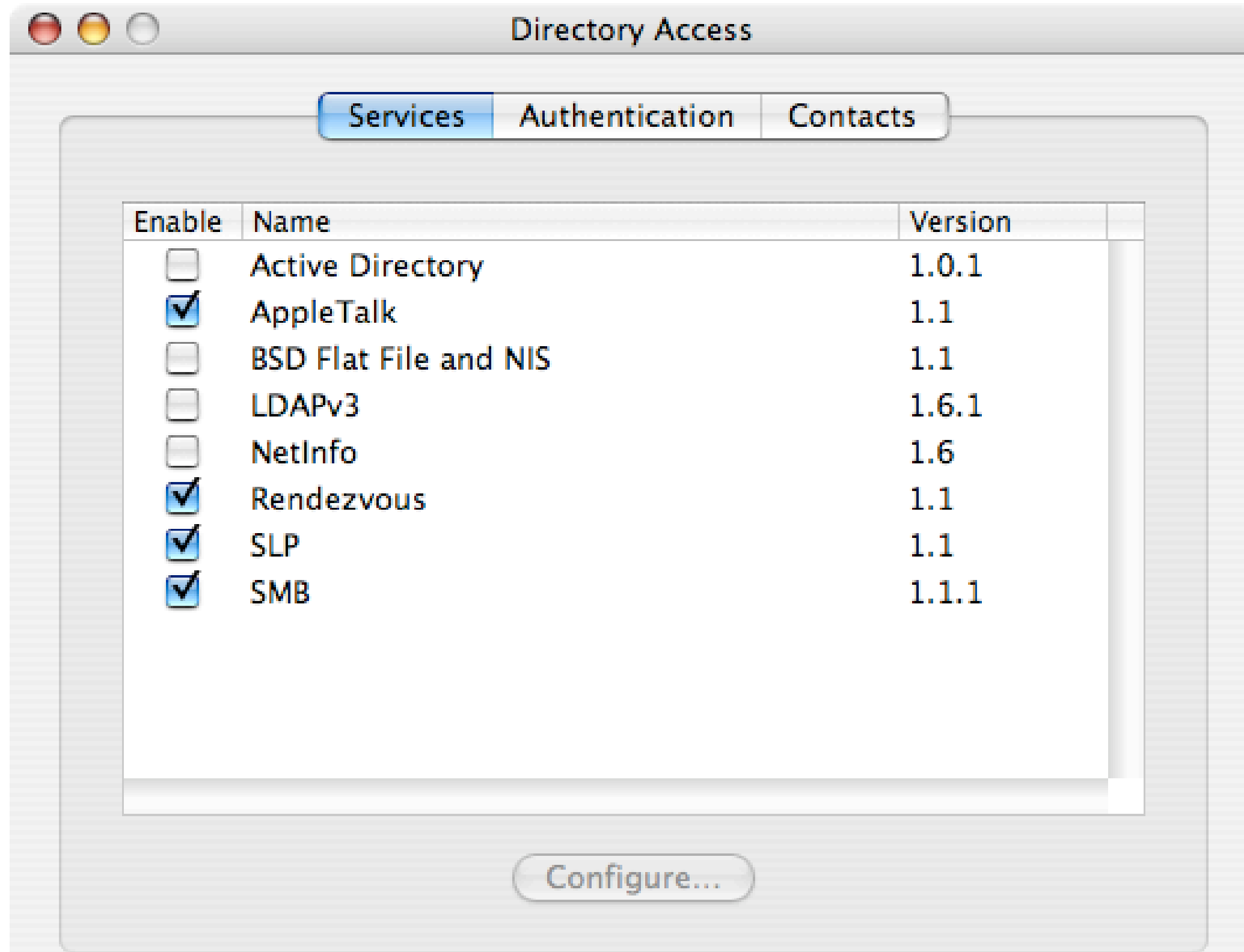
- Password Server leverages public-key/private-key encryption to build a secure system
  - Replication occurs on change or periodically (Administrator's choice)
  - All replicas can accept user password changes
  - Data is merged between two replicas
- All replication sessions are encrypted end-to-end
  - 128-bit key—RC5 encryption
- Client-side replica discovery is quite exhaustive





# **Working with Non-Apple Directories**

# Using Directory Access





# Directory Access

- Clients and server can connect to multiple types of directory services simultaneously.
- Can connect to multiple LDAP directories simultaneously
- Directories can be used for authentication or for contacts information





# **Integrating Mac OS X with Other Directories**

# Open Directory Plugins

- Apple plugins included in OS
  - Active Directory
  - NIS/BSD flat file
- Commercial plugins
  - Thursby's ADmitMac plugin
- Custom or in-house plugins
  - Organizations can develop customized plugins
  - SDK and example plugins is posted
  - Source code for shipping plugins is posted



# Existing Data & Schema

- Top priority: use attributes as intended
- Existing user data should be reviewed
- Attributes matching Mac OS X data requirements should be used
- User name, short name, password are present in most directories, but other values may also be available



# Recycling Unused Attributes

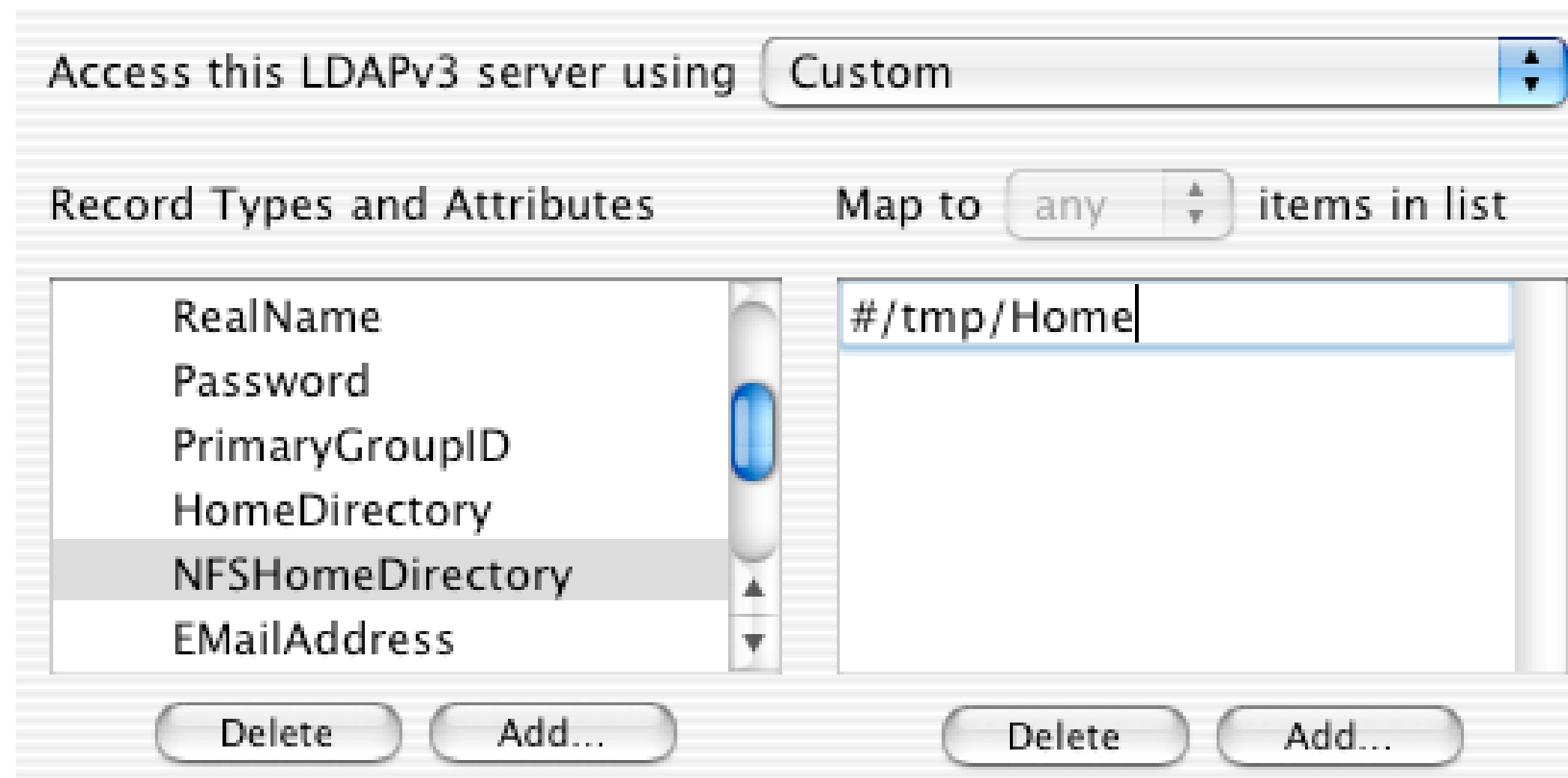
## Unused attributes in a directory can be repurposed

- The admin must carefully track and document the attributes being repurposed
- Server updates and new services must be carefully tested before being deployed
- Some risk for future conflict



# Static Assignment

- Mac OS X allows static assignments of attributes
- Instead of mapping to a field in LDAP, values can be assigned within the plugin's configuration
  - Type “#” and then the value



# Variable Assignment

- Mac OS X allows static assignments to include other directory attributes
- Instead of mapping to a field in LDAP, values can be assigned within the plugin's configuration
  - Type “#” to enable the static & variable assignments
  - Nest the variable with \$
  - Example: **#/tmp/\$uid\$**



# Secondary Directory

- User attributes **CANNOT** be split across directories, but...
- Mounts, Groups, and other classes of objects can be delivered from a separate directory
- These attributes could be stored in:
  - local Netinfo domain
  - NetInfo parent domain on Mac OS X Server
  - LDAP directory on Mac OS X Server

Enable	Configuration Name	Server Name or IP Address	LDAP Mappings	SSL
<input checked="" type="checkbox"/>	Active Directory	192.168.0.5	Custom	<input type="checkbox"/>
<input checked="" type="checkbox"/>	OpenDirectory	192.168.0.10	Open Directory Server	<input type="checkbox"/>





# Extend the LDAP Schema

- All directories allow for this, although some are less forgiving than others
- Can manually edit for simple changes
- Preferred method is to create and thoroughly test LDIF files to modify the schema
- Full schema extension provides one of the best Mac OS X user experiences
- Mac OS X client and server updates should be reviewed for new schema elements or Apple's KnowledgeBase should be consulted when updates are released



# Deploy a Meta-Directory or VDE

- Allows clients to use their native directory
- The Meta automates the synchronization between various directories
- A virtual directory engine provides real-time joining of multiple directories into a single “virtual” directory system
- Reduces problems due to server upgrades or changes in schema requirements





# Active Directory

# Active Directory Plugin

Active Directory Forest:

Active Directory Domain:

Computer ID:

[Unbind...](#)

Hide Advanced Options

Cache last user logon for offline operation

Authenticate in multiple domains

Prefer this domain server:

This domain server will be used when available

Map UID to attribute:

This attribute will be used as the Unique ID

Allow administration by:

Enter group names separated by commas. All members of these groups will have administrator privileges on this computer.

# Client Access to Active Directory

- Users from AD can authenticate at the Mac OS X login screen
- A consistent, but unique UNIX-style UID is generated
- SMB home directory is mounted on the desktop
- Given a local home directory in /Users
- Address Book can search users in Active Directory
- The user's login credentials can be cached for offline access



# Extending the AD Plugin

- Command line tool: dsconfigad
  - Can configure the AD plugin
  - Allows support for server-based home directories
  - Both AFP & SMB can be used



# Mac OS X Server Access to AD

- Access to Active Directory users and groups
- Consistent UID, so access privileges can be set
- Provide all services to users from AD
- Local groups can be populated with AD users
- Kerberized Samba in addition to NTLMv1





TM and © 2003 Apple Computer, Inc. All rights reserved.





**More Information**

# Reference Library

## Documentation, Open Source, References

- Mac OS X Server Documentation
  - <http://www.apple.com/server/macosx/resources.html#g>
  - [http://www.apple.com/server/desktop\\_management.html](http://www.apple.com/server/desktop_management.html)
- Mac OS X LDAP schema
  - Open LDAP Configuration files
- Mac OS X Security APIs
  - <http://developer.apple.com/darwin/projects/cdsa/>



# Reference Library

## Documentation, Open Source, References

- Open LDAP
  - <http://www.openldap.org/>
- SASL
  - <http://asg.web.cmu.edu/sasl/>
- PAM
  - <http://www.kernel.org/pub/linux/libs/pam/>
- Kerberos
  - <http://web.mit.edu/kerberos/www/>



# Reference Library

## Documentation, Open Source, References

- Directory Services API Documentation
  - [http://developer.apple.com/techpubs/macosx/Networking/Open\\_Directory/index.html](http://developer.apple.com/techpubs/macosx/Networking/Open_Directory/index.html)
  - [http://developer.apple.com/techpubs/macosx/Networking/Open\\_Dir\\_Plugin/index.html](http://developer.apple.com/techpubs/macosx/Networking/Open_Dir_Plugin/index.html)
- Darwin Open Directory
  - <http://developer.apple.com/darwin/projects/opendirectory/>
- Open Directory SDK
  - [ftp://ftp.apple.com/developer/Development\\_Kits/DirectoryServicesSDKv1.0.dmg.bin](ftp://ftp.apple.com/developer/Development_Kits/DirectoryServicesSDKv1.0.dmg.bin)





**Q&A**