UNIXとWindowsとの共存

～Sambaを中心としたUNIX-Windows連携の実際～

たかはしもとのぶ

monyo@home.monyo.com

(株)NTTデータ 高橋基信

講師紹介

- 1993 早稲田大学第一文学部卒業
- NTTデータ通信(株)入社
- 1997 頃より、UNIX / Windows NT を中心とした OS やネットワーク、インターネット関連の技術サポートを行い、現在に至る
- MCSE / LPI / SCNA / CCDA などを保有
- 日経Windows 2000のQ&Aコーナで回答を担当
- Sambaの開発に携わる(2.0.10日本語版/3.0)
- 著書「アンドキュメンテッドMicrosoftネットワーク」

チュートリアルの目的

- UNIXとWindowsとの共存におけるオプションとそのメリット、デメリットについて理解する
- Sambaの機能について理解する
- SFUの機能について理解する

チュートリアルの概要

- UNIXとWindows共存のシナリオ
- Sambaの基本機能
  - ファイルサーバ、プリンタサーバ、ネットワーク
- Sambaの設定応用編
  - PDC機能、LDAPによる認証統合、メンバサーバ機能、Winbindによる認証統合
- SFUの機能紹介

UNIXとWindowsとの共存のシナリオ

- 以下の三点で説明する
  - ファイル共有 / プリンタ共有 / 認証
- 基本的に、あまり相性はよくない

	ファイル共有	プリンタ共有	認証
UNIX	NFS	LPRなど	NIS/LDAP
Windows	SMB	SMB(LPRも可)	NTドメイン Active Directory

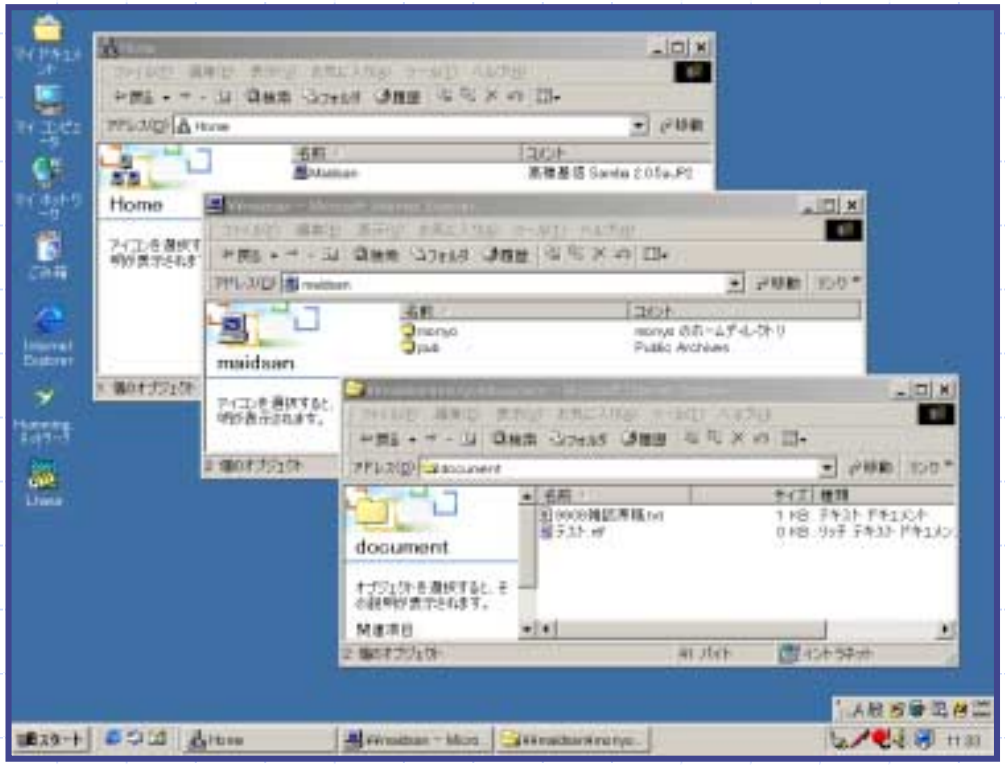
- おのおの標準としているプロトコルが違う

ファイル共有

- UNIX側はNFS、Windows側はSMBが基本
  - UNIX側にSMBを実装する
    - Samba (サーバ)
    - smbsh、smbfs、Sharityなど (クライアント)
  - Windows側にNFSを実装する
    - SFU (Microsoft Services for UNIX)(サーバ、クライアント)など
- その他の解
  - WebDAV
  - FTP

UNIXをSMBサーバにする

- Sambaのインストール
  - Windowsクライアント側からは、Windowsサーバと区別がつかない
  - ほとんどのLinux, 商用UNIXに添付
    - Solaris
    - hp-ux
    - AIXなど



UNIXをSMBサーバにする(2)

- Sambaのメリット
  - 初期導入コストが安い
    - クライアントマシンへのソフトウェア導入は不要
  - 運用管理面
- Sambaのデメリット
  - 互換性、新製品への追従
  - 日本語(完全)対応
    - Samba日本語版、Samba 3.0では対応…のはず
  - 性能面

WindowsをNFSクライアントにする

- 何らかの製品を導入する(SFUなど)
  - 各クライアントマシンにNFSクライアントのインストールが必要
    - 費用や運用が大変
  - UNIXサーバとのファイルパーミッションの問題
  - ユーザ名マッピング
  - 日本語ファイル名の問題

WindowsをNFSサーバにする

- 何らかの製品を導入する(SFUなど)
- メリット
  - UNIXマシンからは通常のコマンドでファイル共有
- デメリット
  - 需要はあるのか?
  - UNIX側とのファイルパーミッションの問題
  - 日本語ファイル名の問題
  - UNIXクライアントからの接続にはCALが必要

UNIXをSMBクライアントにする

- smbfs (Linux)
  - Linuxカーネルに同梱されている機能

```
# mount -t smbfs -o username=ユーザ名,password=パスワード,¥
  codepage=cp932,iocharset=ja_JP.euc_JP ¥
  //サーバ名/共有名 /マウント先
```

- smbsh
  - Samba汎用、root権限不要だが……
    ややかなり)挙動不審?

```
% smbsh -W DomainName -U UserName
Password:
% ls /smb
```

UNIXをSMBクライアントにする(2)

- Sharity (http://www.obdev.at/products/sharity/index.html)
  - 様々なOSに対してsmbfsと同等の機能を実現する
  - 有償
- Sharity-Light (http://www.obdev.at/products/sharity-light/)
  - 無償(GPL)だが、機能も限定されており、不安定
    - 開発も数年前に停止
- smbclient(Samba)
  - FTPライク、ファイル転送

```
# shlight  //ntsrv/temp /mnt1 -U ntuser
  Password: ← パスワードを入力する
  Using port 1889 for NFS.
# ls /mnt1
```

UNIXをSMBクライアントにする(3)

- そもそも需要があるのか?
- 特徴
  - 廉価(に済む場合もある)
  - UNIXファイルシステムとして扱える
  - ライセンスの問題
    - WindowsサーバのCALが必要
  - アクセス権、ユーザ名マッピング
  - 日本語対応
    - 完全ではない

WebDAVによるファイル共有

- Windowsに付属のWebDAVクライアント機能
  - Webフォルダ
- メリット
  - HTTPが通過すれば（ほとんどは）使用可能
- デメリット
  - 実際に行われているのはファイル転送
  - セキュリティ
  - アクセス権、ユーザ名マッピング
  - 日本語まわりの問題
  - WindowsのCAL(IISをWebDAVサーバとして利用する場合）

WebDAVによるファイル共有(2)

- WebDAVサーバ/クライアントの設定例
  http://www.atmarkit.co.jp/flinux/special/webdav/webdav02a.html

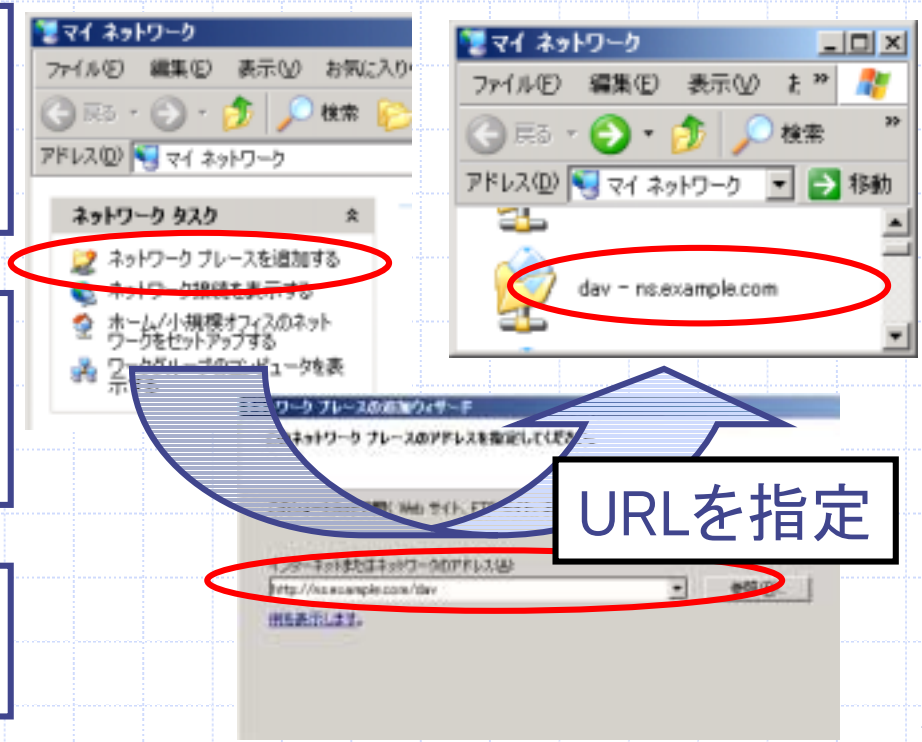
UNIXサーバ

1.Apacheのモジュール構成確認
mod_davが導入されているか？
導入されていない場合は要導入

2.Apacheの設定
WebDAVで公開するURLやアクセス制限定義
アクセス制限にはLDAPなども利用可能

3.Apache再起動
 # apachectl restart

Windowsクライアント側



FTPによるファイル共有

- Windows、UNIXともにFTPサーバ、クライアント機能がある
- メリット
  - 両OSの標準機能だけで実装可能
- デメリット
  - ファイル転送であって、直接ファイルの共有ができるわけではない
  - セキュリティ
  - WindowsのCAL

ファイル共有のまとめ

- WindowsからUNIX上への書き込みの場合、Sambaが無難
  - 基本的に無償
  - Windows側からシームレス
- UNIXからWindows上への書き込みの場合は、ケースバイケース
  - 基本的にOA環境ではあまりありえないケース?
  - 日本語ファイル名の問題
  - LinuxならsmbfsもしくはNFS（もしくはFTP）

UNIXをSMBプリンタサーバにする

- SambaでWindowsプリンタサーバを構成
  - プリンタドライバの自動ダウンロードも可能
- 特徴
  - Windowsクライアント側からシームレス
  - Windowsの固有機能が使えない
    - プリンタのステータス情報取得など
  - 作りこめば凝ったことが可能
    - Sambaの場合、実際には印刷以外の任意のコマンドを実行できる(FAXプリンタ、PDFライタなど)

プリンタ共有

- UNIX側はlpr、Windows側はSMBが基本
  - UNIX側にSMBを実装する
    - Samba (サーバ)
    - smbclient (クライアント)
  - Windows側にlprを実装する
    - UNIX用印刷サービス
    - Windows 9x系OSには標準では機能がない
  - ネットワークプリンタを使用する
    - マルチプロトコルで両OSに対応

WindowsをIprクライアントにする

- Windows NT系OSでのみ可能
  - 「UNIX用印刷サービス」サービスをインストールして構成する
  - Windows 9x系OSでは不可
    - 別途ツールをインストールするか、Windows NT系OSを経由する必要がある
- 特徴
  - Windows固有の機能が使えない

WindowsをIprサーバにする

- Windows NT/2000/2003をIprサーバにする
  - 「UNIX用印刷サービス」をインストール、構成する
  - UNIX側からは通常の印刷機構で接続する
- 特徴
  - UNIX側でプリンタの設定を行う必要がある
    - UNIXのプリンタサポートはWindowsと比べるとよくない
    - 設定が複雑になりがち
  - WindowsのCALが必要

UNIXをSMBプリンタクライアントにする

- SambaのsmbclientコマンドでWindowsの印刷共有に接続、印刷
  - バッチ処理も可能

```
% smbclient //server/printshare
smb:¥> print <FileName>
smb:¥>exit

% smbclient //server/printshare Password ¥
  -U UserName -c 'print <FileName>'
```

- 特徴
  - 認証情報を印刷スクリプト中で固定的に持つ必要がある
  - UNIX上の印刷機構から印刷させる設定は複雑
  - 印刷データをUNIX上で加工しておく必要がある

ネットワークプリンタの導入

- ネットワークプリンタを導入する
  - WindowsからはSMBプリンタ
  - UNIXからはlprのPSプリンタとして接続する
- 特徴
  - 機能上は、一番無難
  - 一般的に価格が高い

プリンタ共有のまとめ

- 可能であれば、ネットワークプリンタを導入するのが簡単
  - UNIX上でのプリンタ設定は一般的に難しい
  - Windowsをプリンタサーバにする場合はCALが必要になる
- 印刷は常にWindows上から行う
  - UNIX上で生成したファイル(PDFなど)をWindows側からファイル共有した上印刷する

認証

- Windows側はドメインが基本
- UNIX側はNIS、LDAPなど
  - ただし、標準といえるかどうかは微妙
- Windows側でUNIX側の認証機構を実装
  - Kerberosのレルムに接続(Windows 2000以降)
  - GINA.DLLの置き換え(Windows NT系OS)
  - SFUを導入してNISサーバとして構成
  - Kerberosサーバとして使用（Active Directory)

認証(2)

- UNIX側でWindows側の認証機構を実装
  - SambaのWindowsドメイン参加
    - Sambaのリソースの認証をWindowsドメインで行う
  - auth_ntlm(Samba)
    - コマンドからWindowsドメインの認証機能を使用
  - Winbind(Samba)
  - pam_ntdom
  - pam_krb5
    - PAM経由での認証をWindowsドメインで行う

WindowsをUNIXの認証サーバに(1)

- SFUのNISサーバ機能を使用する
  - Active Directory必須
  - NISのパスワード管理に注意が必要
  - SFU自体の動向がやや不透明
- 特徴
  - 別途SFUが必要
  - NISのセキュリティ面を無視できる場合はそれなりに有効か

WindowsをUNIXの認証サーバに(2)

- Kerberosサーバとして使用する
  - Microsoft社から多数の技術情報あり
    - Kerberos 5 (krb5 1.0) の相互運用性
      http://www.microsoft.com/japan/technet/prodtechnol/windows2000serv/deploy/confeat/kerbstep.asp
    - Windows 2000 Kerberos の相互運用性
      http://www.microsoft.com/japan/windows2000/techinfo/howitworks/security/kerbint.asp
    - Interoperability with Microsoft Windows 2000 Active Directory and Kerberos Services
      http://msdn.microsoft.com/library/en-us/dnactdir/html/kerberossamp.asp
- 特徴
  - そもそもUNIXで一般的な認証ではない?
  - 手順が煩雑

WindowsをUNIXの認証サーバに(3)

- LDAP認証サーバとして使用できない
  - Active DirectoryはLDAPディレクトリサービスを提供するが、認証自体はKerberos認証かNTLM認証
    - LDAP認証はできない
  - InetOrgPersonなどのスキーマ、UserPassword属性はあるが、Windows自身は使っていない
  - LDAPでのアカウント作成やパスワード変更は可能
    - 269190: [HOWTO] LDAP を介して Windows 2000 ユーザーのパスワードを変更する

外部認証機構でWindowsを認証(1)

- Kerberosのレルムに接続
  - Windows 2000以降で可能
  - リソースキットのksetupコマンドを使用
- 特徴
  - そもそもKerberos認証が一般的ではない?
  - 認証以外はWindows側で独自に管理するしか
    - アカウント自身やグループポリシーなど

外部認証機構でWindowsを認証(2)

- Kerberosのレルムに接続
  http://www.monyo.com/technical/windows/kerberos2.html

UNIX(Kerberos)サーバ側

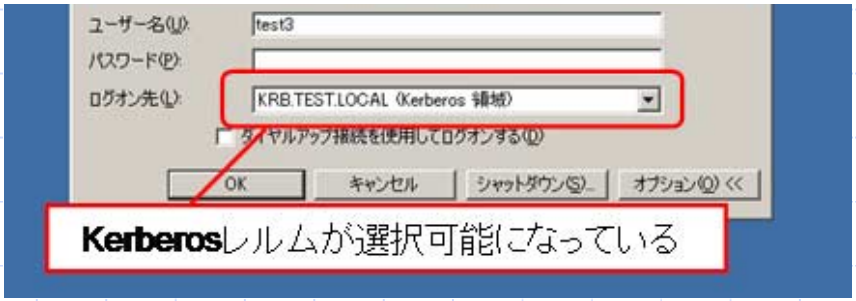
1.KDCの設定
DB初期化、管理アカウント作成
/etc/krb5.confの設定

2.DNSの設定
kerberos-admのSRVレコードを追加

3.プリンシパル登録
host/*host.realm.com@REALM.COM*
user@REALM.COM

Windowsクライアント側

```
C:¥>ksetup /setdomain KRB5.COM
 Setting Dns Domain
C:¥>ksetup /addkdc KRB5.COM
192.168.1.1
C:¥>ksetup /setmachpassword pass
 Setting computer password
C:¥>ksetup /mapuser * *
```



外部認証機構でWindowsを認証(3)

- GINA.DLLを改造する
  - 認証に使用するDLLを改造、追加する
- 特徴
  - なんでもあり
    - LDAPで認証させることなども可能
    - NDS認証や認証VLANなどは、この機能を使っている
  - 高度なプログラミングが必要
    - 通常はこの方法は無理

UNIXをWindows認証サーバに

- Sambaのドメイン機能
  - Windows NT互換のPDC、BDCとして機能
    - ただし、BDC機能には制限がある
    - Active DirectoryのDCとして機能することはできない
- 特徴
  - CAL不要
  - あくまでWindows NT互換

外部認証機構でUNIXを認証(1)

- Sambaリソースの認証
  - Sambaをドメインメンバとして構成
  - Windowsとの統合度は高い
- コマンドラインからWindowsの認証機構を使用
  - auth_ntlm(Samba)
  - 特徴
    - ユーザ名、パスワード認証のみ(squid用の認証コマンド)
    - Winbind(PAM)必須

```
% ntlm_auth --helper-protocol=squid-2.4-basic
UserName UserPass
ERR
```

外部認証機構でUNIXを認証(2)

- Winbind(Samba)
  - Sambaをドメインメンバとして構成した際のアドオン
  - UNIX側にアカウントを作成する必要がない
    - NISなどと同様
  - PAM経由でSamba以外のプロダクトも認証できる
- 特徴
  - 多機能、現在も開発継続中
  - やや不安定(特にLinux以外)
  - PAM,libnss機能が必要

外部認証機構でUNIXを認証(3)

- pam_ntdomによるPAM認証
  - NTLMベースで単純なユーザ名、パスワード認証
  - 現在では、Winbindを使ったほうがよい
- pam_krb5
  - Active DirectoryのKerberosサーバ機能を使用
- 特徴
  - WindowsのCALが必要な場合がある
  - PAMの使用が大前提

認証のまとめ

- Windowsの認証をUNIXに行わせるのは困難
  - どうしても必要なときはGINA.DLLの改造か、Kerberos認証
  - アプリケーションレベルで対応したほうが簡単
- UNIXの認証をWindowsに統合することは可能
  - 方法はいろいろあるが、定番といえるものはない
  - 状況により使い分ける必要がある

Sambaの基本機能

ファイルサーバ
プリンタサーバ
認証
ネットワーク
クライアント

Samba のファイルサーバ機能(1)

- ファイルシステムの差分吸収
  - 地味ではあるが、相互運用において重要
    - Windowsのファイル属性とUNIXのパーミッションの対応
      - map archive / map system / map hidden
        - 各ファイル属性をUNIXの実行権ビットに対応させる
    - DOS仕様のファイルアクセスの考え方
      - delete read only
        - 読み取り専用ファイルを削除する?
      - dos filetimes
        - 書き込み権があれば、時刻変更を可能にする?

Samba のファイルサーバ機能(2)

- DOS仕様のファイルアクセスの考え方(続)
  - dos filemode
    - 書き込み権があれば、アクセス権変更を可能にする?
- Windows側ファイルシステムの時刻管理(2秒単位)対応
  - dos file time resolution
    - 時刻が2秒単位に丸められる。FAT互換性維持のため
      http://support.microsoft.com/default.aspx?scid=kb;ja;402160
- Windows側のファイル作成時刻情報に対応
  - fake directory create time
    - UNIXには存在しないディレクトリ作成時刻の情報を設定
      http://www.samba.gr.jp/project/kb/J0/1/15.html
      デフォルトでは最終更新時刻

Samba のファイルサーバ機能(3)

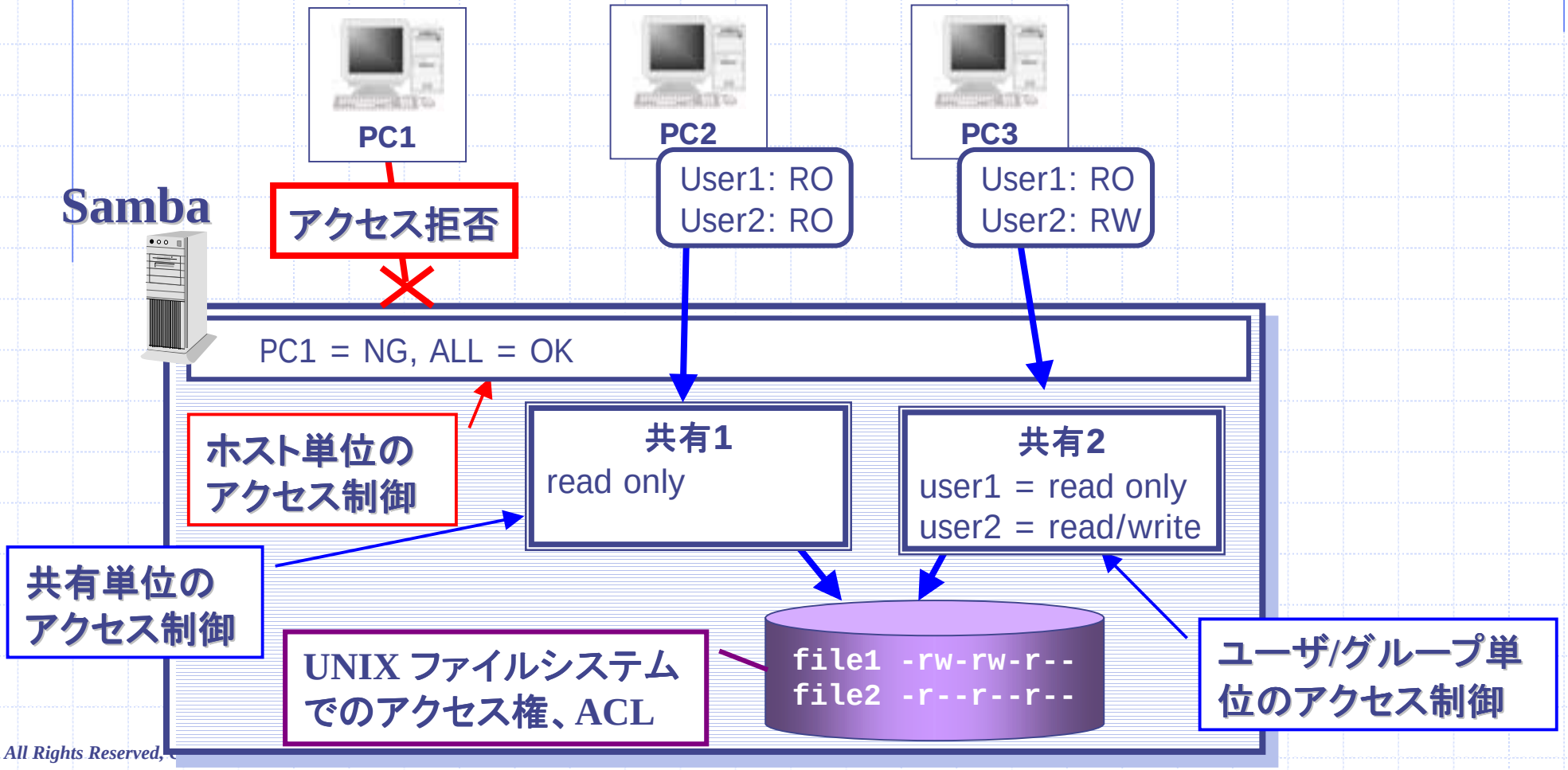
- 日本語ファイル名
  - シフトJISの機種依存文字や外字も完全サポート Samba日本語版やSamba 3.0で実現
    - シフトJIS正規化(日本語版)
      - Windows NT系と9x系OSの文字コードの違いを吸収
    - 外字、機種依存文字(日本語版)
      - マッピングテーブルを追加
    - 大文字、小文字変換ロジックの修正(日本語版,Samba 3.0)
      - 1バイトずつ大文字化することによりファイル名が化ける
    - 大文字、小文字同一視ロジックの修正(日本語版)
      - Windows NT系OSと完全互換

Samba のファイルサーバ機能(4)

- 日本語ファイル名(続)
  - 禁止文字の対処(日本語版、Samba 3.0)
  - 大文字、小文字のバイト長の差異に対応(Samba 3.0)
  - 短いファイル名の生成ロジックの対応(Samba 3.0)
  - 各種符号化形式に対応
    - EUC-JP / Shift_JIS / CAP / HEX / UTF-8 など
      - client code page / coding system
      - dos charset / unix charset
- その他のリソースの日本語対応
  - 共有名 / コンピュータ名 / ワークグループ名など

Samba のファイルサーバ機能(5)

- アクセス制御



Samba のファイルサーバ機能(6)

- アクセス制御(続)
  - ホスト単位でのアクセス制御
    - IPアドレスでのアクセス制御
      - `hosts allow, hosts deny`
  - 共有単位でのアクセス制御
    - ユーザ、グループ単位でのアクセス制御
      - `valid users, invalid users, admin users`
    - 共有レベルでのアクセス制御
      - `read only, write list, read list`
    - IPアドレスでのアクセス制御
  - ファイルシステムレベルでのアクセス制御
    - 伝統的なアクセス権、ACL(サポートされているOSのみ)

Samba のファイルサーバ機能(7)

- アクセス制御(続)
  - ホストベースのアクセス制御
    - ホスト単位、共有単位で設定可能
      - [global]セクションで設定するとホスト単位
  - 読書きレベルでのアクセス制御
    - デフォルトを指定後、例外のユーザやグループを個別に指定
  - ユーザ単位でのアクセス制御
    - 特定ユーザ、グループにroot権限の付与も可能(共有単位)

ホストベース

```
[global]
  ...
  hosts allow = 192.168.10.

[share]
  hosts allow = 192.168.1.
```

読み書きレベル

```
[share1]
  ...
  read only = yes
  write list = monyo,@admins
```

基本的にはread onlyだが、write listのユーザは書き込める設定

ユーザ単位

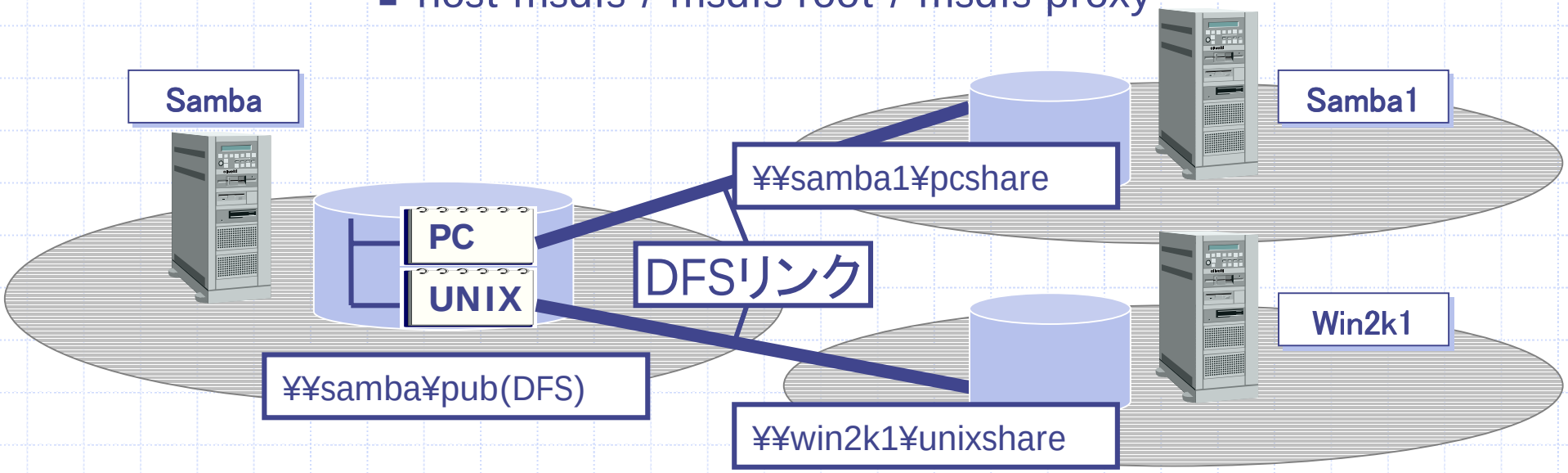
```
[share1]
  invalid users = @users
  valid users = monyo
  admin users = monyo
```

Samba のファイルサーバ機能(8)

- ACL(アクセスコントロールリスト)
  - 本格的なWindowsファイルサーバには必須
  - 共有のACL、ファイルACL共に対応
    - ただしファイルACLはOSがPOSIX ACL対応している場合のみ
    - ドメインに参加しているSambaサーバの場合、ドメインのアカウントにACLを付与できる
    - UNIX側ではsetfacl/getfaclコマンドなどで設定

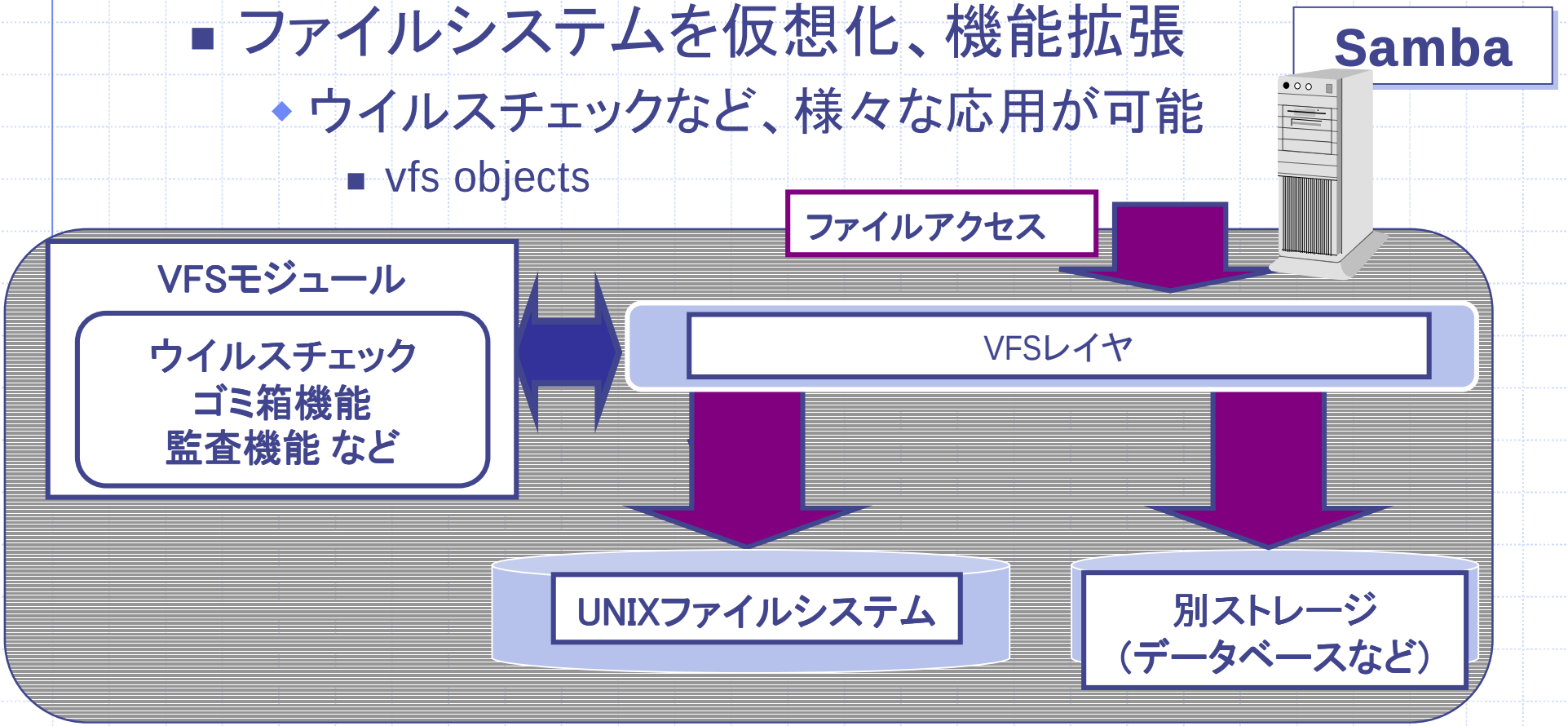
Samba のファイルサーバ機能(9)

- 仮想ファイルシステム(VFS)
  - 実際のUNCとは別の仮想的なUNCでファイルサーバにアクセスする機能
    - スタンドアロンDFSのみをサポート
      - host msdfs / msdfs root / msdfs proxy



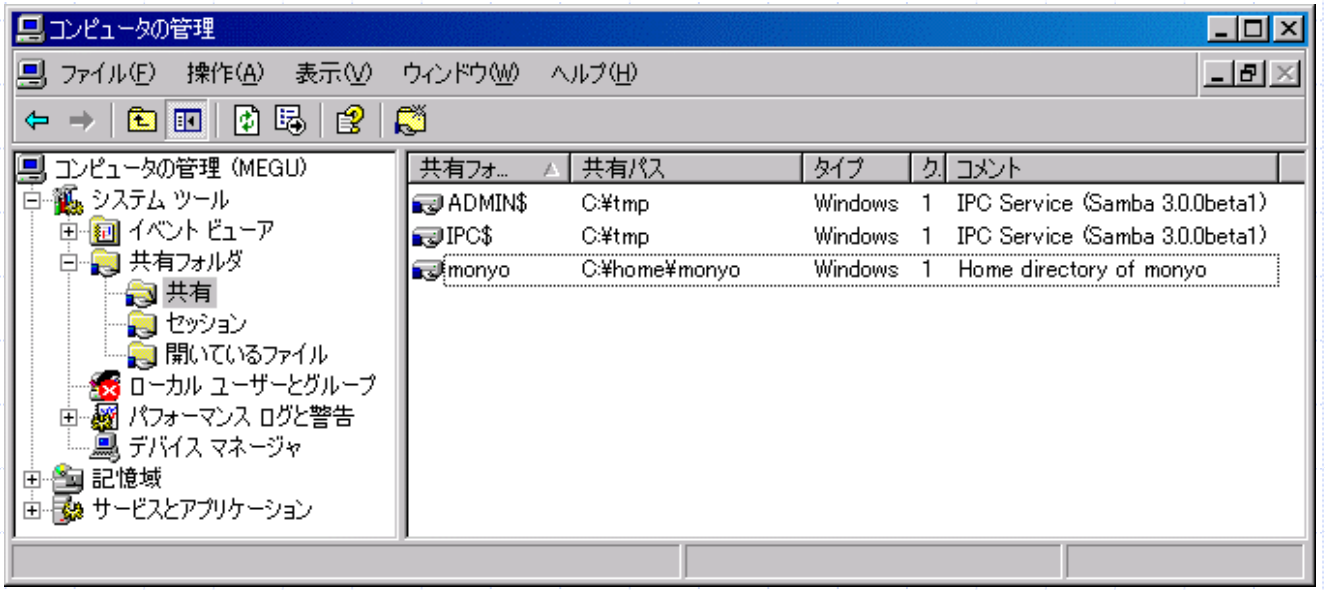
Samba のファイルサーバ機能(10)

- 分散ファイルシステム(DFS)
  - ファイルシステムを仮想化、機能拡張
    - ウイルスチェックなど、様々な応用が可能
      - vfs objects



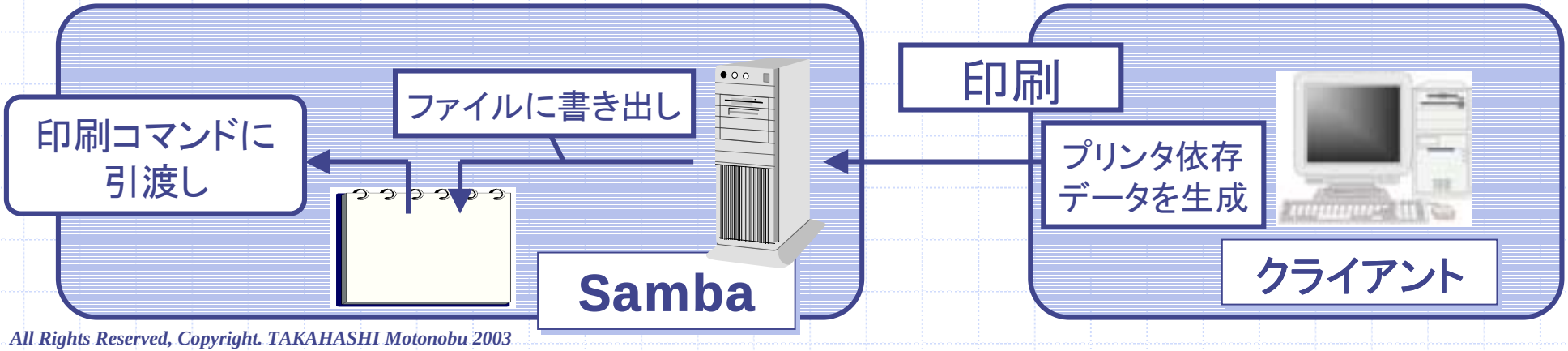
Samba のファイルサーバ機能(11)

- Windows GUIからの管理
  - Windowsマシンと同様にしてシームレスに管理
    - 共有の作成、修正、削除
      - add share command / change share command
        delete share command



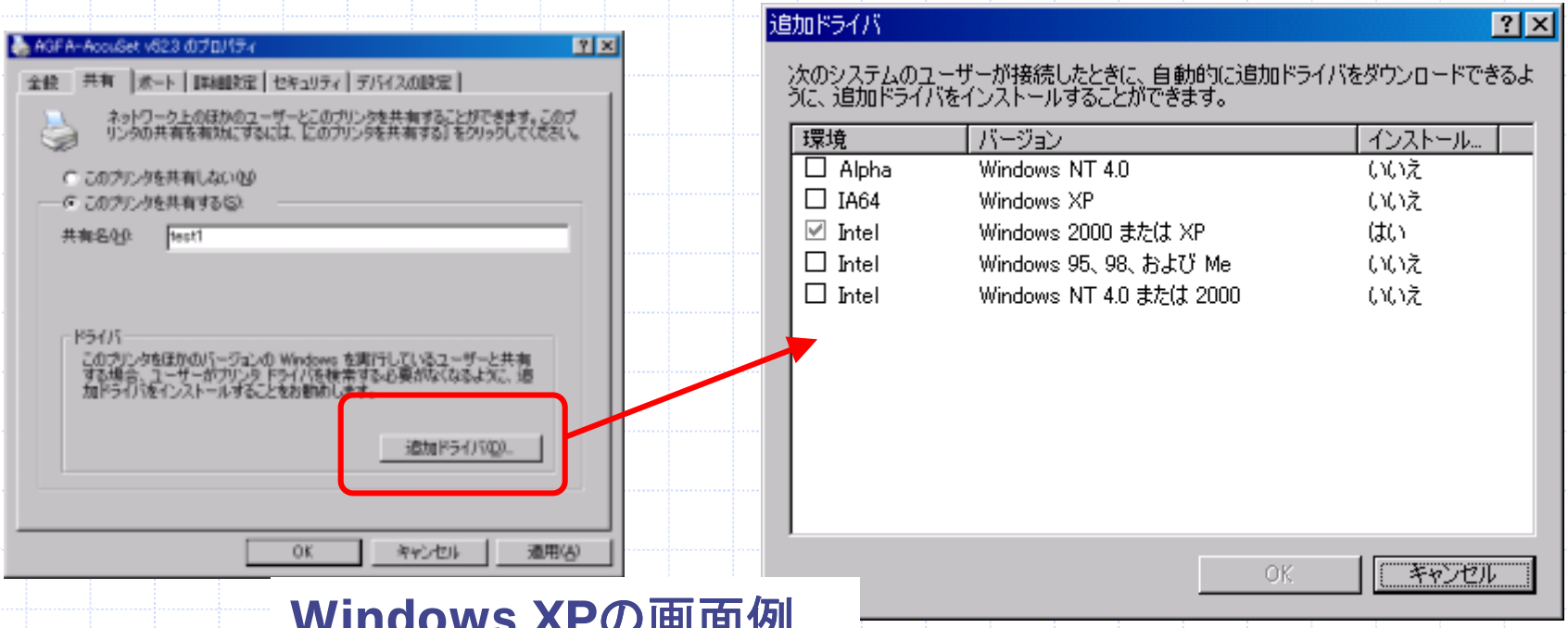
Sambaのプリンタサーバ機能(1)

- 基本的な印刷サポート
  - Windows 9xレベルのプリンタサーバ
    - printable
  - プリンタドライバはクライアント側で用意
  - Sambaは受信データをファイルに保存した上で、印刷コマンドを呼び出す(lprコマンドなど)
    - UNIX側でデータを加工しないこと



Samba のプリンタサーバ機能(2)

- プリンタドライバの自動ダウンロード機能
  - クライアント上でのプリンタドライバインストール不要
    - 個別のインストール作業を不要に



Windows XPの画面例

Samba のプリンタサーバ機能(3)

- プリンタドライバの自動ダウンロード機能(続)
  - Samba側の設定方法

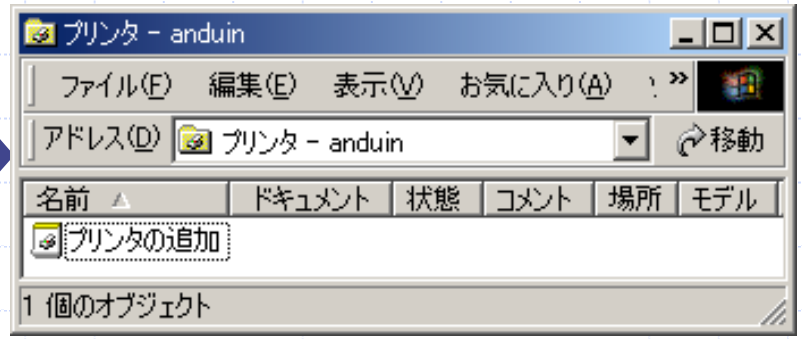
1.smb.confの設定

```
[print$]
     path = /path/to/somewhere
     guest ok = yes
     browseable = yes
     read only = yes
     write list = root
```

2.Sambaサーバ接続

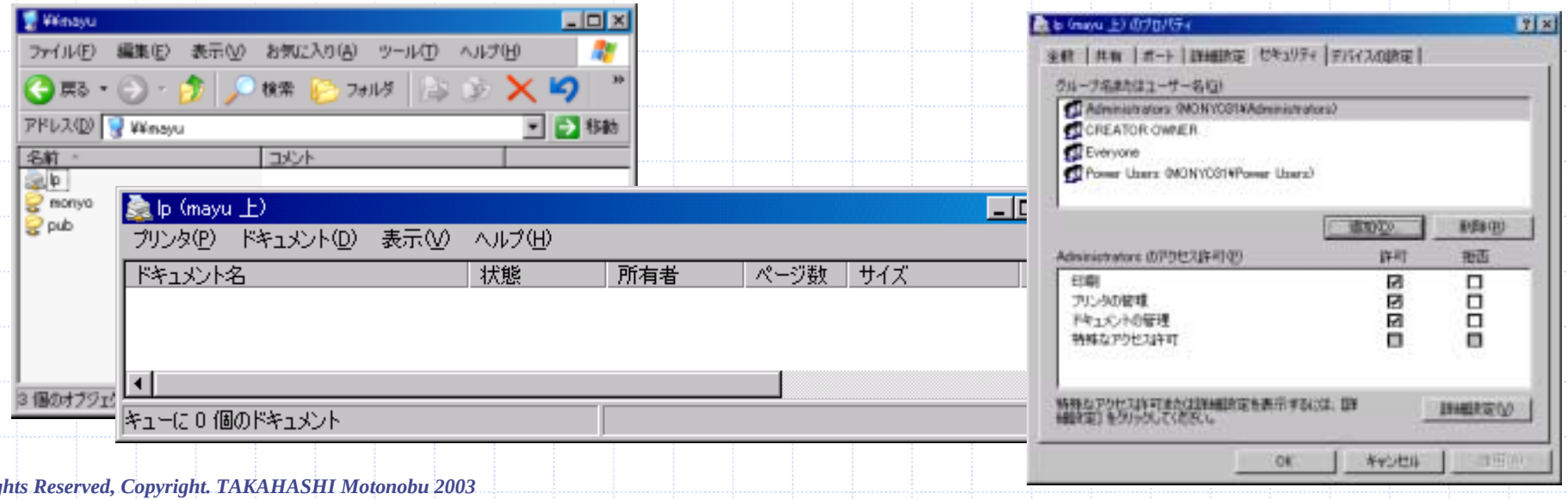
printer adminかrootで接続すること!

3.プリンタ追加ウィザード経由で、プリンタドライバをアップロード



Samba のプリンタサーバ機能(4)

- Windows GUIからの管理
  - Windowsマシンと同様にしてシームレスに管理
    - プリンタ共有へのACL設定
    - 印刷共有の作成、修正、削除
      - add printer command / delete share command



Samba のプリンタサーバ機能(5)

- 「印刷」コマンドの柔軟なカスタマイズ
  - UNIXコマンドで可能な処理であれば、何でも行わせることが可能
    - ファイルへ出力 / PDFライタ / FAXプリンタ
      - printing / lpr command / lpq command ...
- CUPSサポート
  - 最近のLinuxディストリビューションには実装
    - printing = cups / printcap name = cups

Sambaの認証機能(1)

SMBの各種セキュリティのサポート

機能	Windows NT/2000/XP	Samba 2.2	Samba 3.0
平文パスワード抑止	**EnablePlainTextPassword**	encrypt passwords	**encrypt passwords** **client plaintextauth**
LMレスポンス抑止	**LMCompatibilityLevel**	**lanman auth**	**lanman auth** **client lanman auth**
NTLMレスポンス抑止	**LMCompatibilityLevel**	不可能	**ntlm auth** **client ntlmv2auth**
NTLMv2対応	**LMCompatibilityLevel**(NT 4.0 SP4以降)	不可能	**ntlmv2 auth** **client ntlmv2auth**
SMB署名	Require(Enable)SecuritySignature (NT 4.0 SP3以降)	不可能	**server signing** **client signing**
セキュアチャネル 署名･暗号化	RequireSignOrSeal (NT 4.0 SP4以降)など	不可能	**server schannel** **client schannel**

Sambaの認証機能(2)

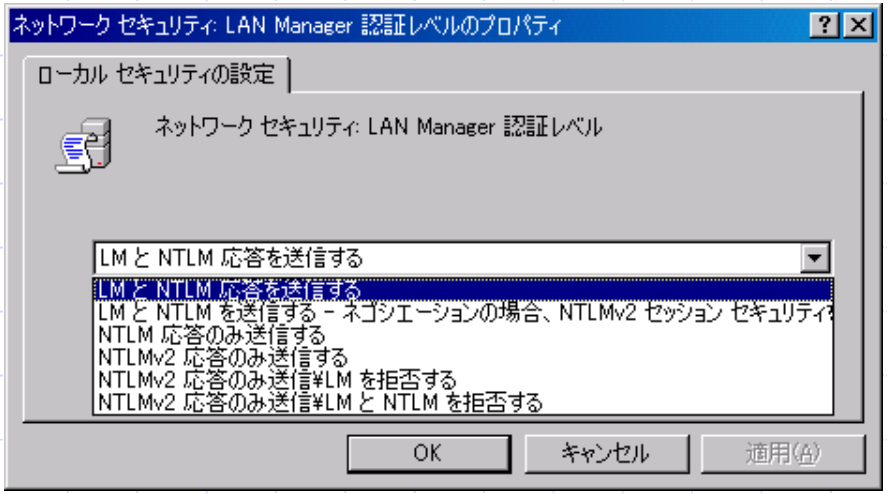
- SMBの各種セキュリティのサポート(続)
  - いわゆる暗号化パスワード
    - 平文パスワードに対応する概念(実際はハッシュ)
      - encrypt passwords
  - 暗号化(ハッシュ)の種類

LMレスポンス	Windows 9x以降	DES
NTLMレスポンス	Windows NT以降	MD4
NTLMv2レスポンス	Windows NT 4.0 SP4以降	HMAC-MD5

  - SMB署名
    - 伝送路での改ざん、リプレイ攻撃を抑止

Samba の認証機能(3)

- SMBの各種セキュリティのサポート(続)
  - Windows側の設定
    - LMcompatibilityLevelレジストリなどに対応
    - ローカルポリシーで設定可能(Windows 2000/XP)
      - LAN Manager認証レベル
      - ……通信にデジタル署名を行う
      - サードパーティ製のSMBサーバへのパスワードを暗号化せずに送信する



Sambaの認証機能(4)

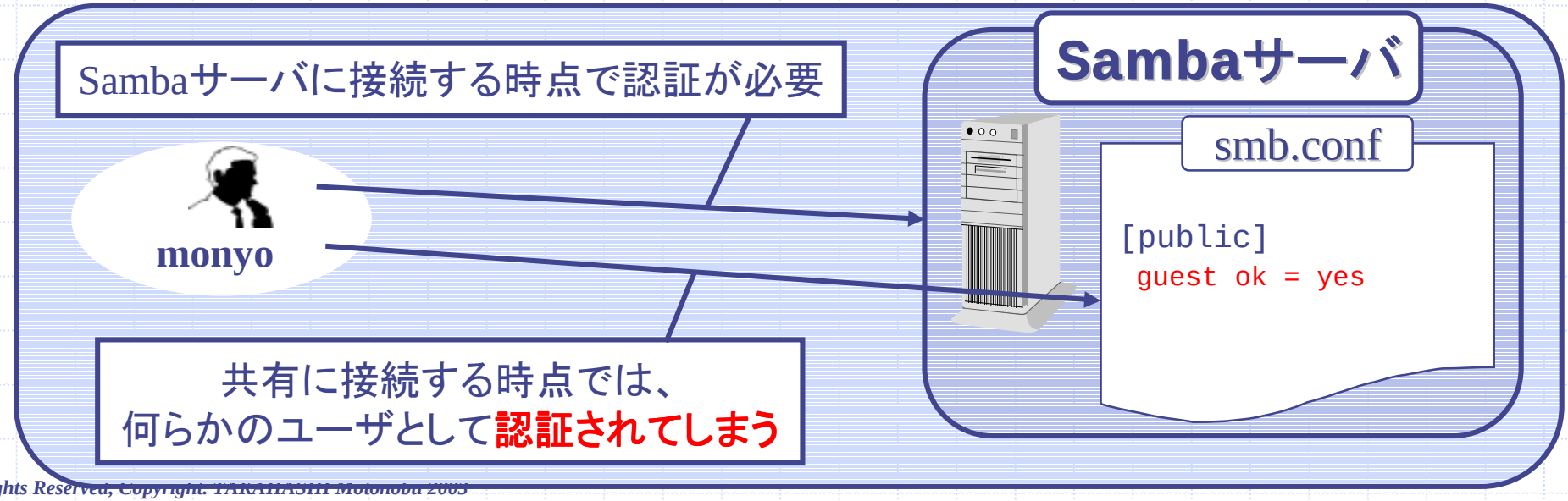
- Windowsドメインのサポート
  - NTドメインを完全に置き換え可能
    - PDC / BDC機能
      - domain logons
    - 認証データベースにLDAPを使用可能
  - ドメインメンバーとしても機能
    - NTドメイン、Active Directoryに参加可能
      - security = [domain|ads]

Samba の認証機能(5)

- 共有レベルのセキュリティのサポート
  - Windows 9x系OSと同等の認証機能
    - 簡易ではあるが、セキュリティは低い
      - security = share
- 別のSMBサーバへの認証委任
  - 非ドメイン、非LDAP環境での認証統合が可能
    - 今後はLDAP化がトレンド?
      - security = server

Sambaの認証機能(6)

- ゲスト認証のサポート
  - WindowsのGuestアカウント相当の機能を実装
    - 認証なしのアクセスを実現
      - map to guest = Bad User - 存在しないアカウントでのアクセスをゲストアクセスとみなす



Sambaの認証機能(7)

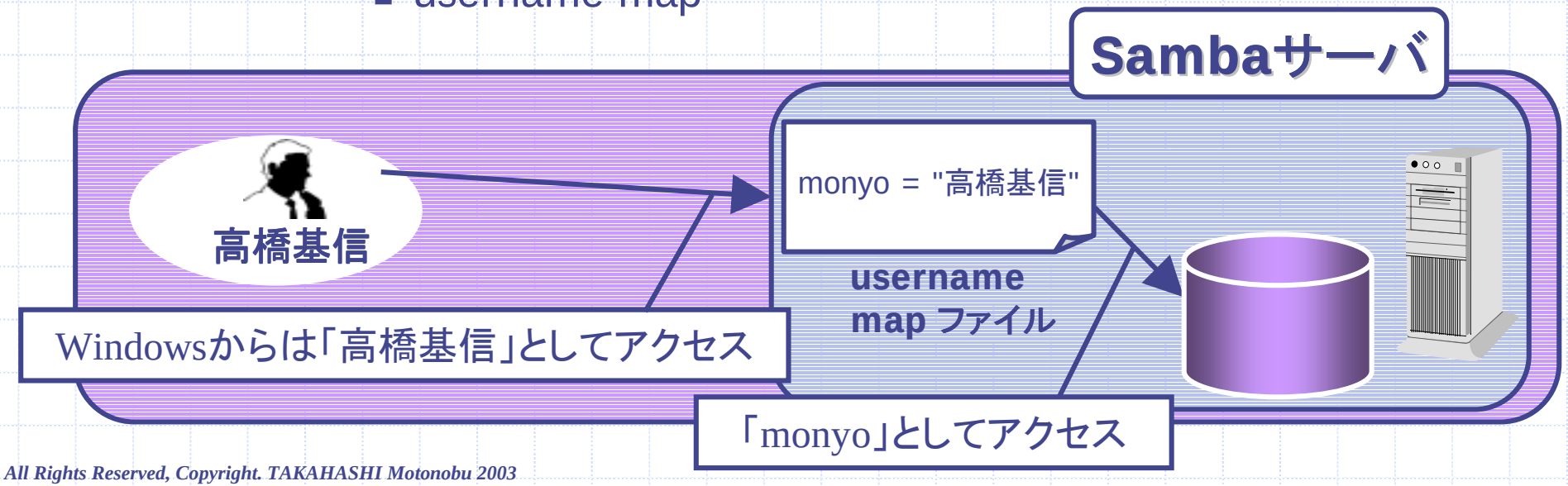
- 柔軟な認証データベース
  - バックエンドにLDAPやNIS+などを使用可能
  - Samba 3.0からは、複数の認証方式を組み合わせて使用できる
    - passdb backend

```
[global]
  passdb backend = ldapsam:ldap://ldap.monyo.com ¥
        smbpasswd
```

キーワード	認証方式
smbpasswd	デフォルト、従来からの方式(ファイルのパス名を指定)
tdbsam	TDB形式のデータベース(ファイルのパス名を指定)
ldapsam	LDAPサーバ(LDAPサーバのURLを指定)
nisplussam	NIS+サーバ(NIS+ドメイン名を指定)
mysql	mysqlのデータベース

Sambaの認証機能(8)

- ユーザ名マッピング
  - Windows側とUNIX側で異なるユーザ名を使用可能
    - 間接的に日本語ユーザ名をサポートできる
      Windows側「高橋基信」、UNIX側「monyo」など
      - username map



Samba のネットワーク機能(1)

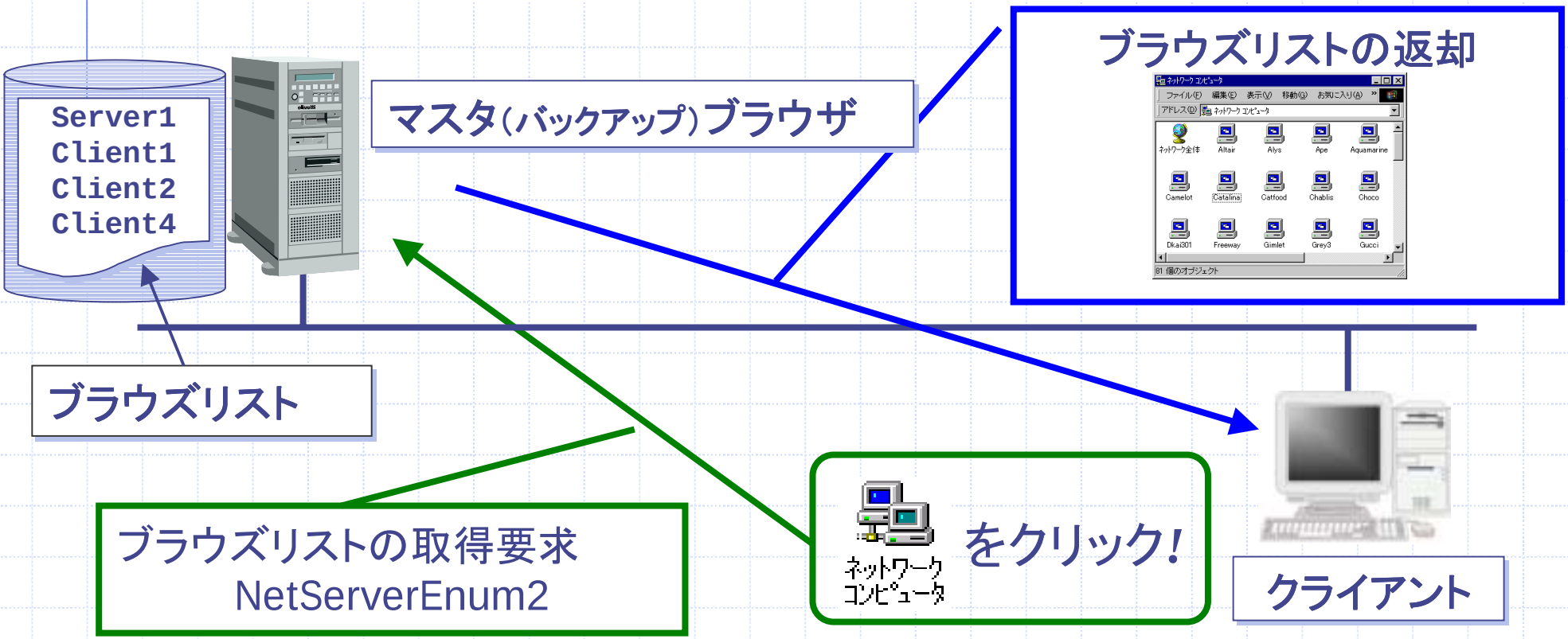
- 名前解決機能
  - Windows固有のNetBIOSの名前解決に対応
    - NetBIOSブロードキャストの名前解決に対応
      - nbtstatコマンドにも応答する
    - WINSクライアント機能
      - WINSサーバへの名前登録
        - wins server

Samba のネットワーク機能(2)

- 名前解決機能(続)
  - Windows固有のNetBIOSの名前解決に対応
    - WINSサーバ機能
      - wins support
      - 複製機能は未実装
      - Dynamic DNSなど外部プログラムとの連携も可能
        - wins hook
      - 静的エントリ

Samba のネットワーク機能(3)

- ブラウジング機能
  - 「ネットワークコンピュータ」を提供する機能



Samba のネットワーク機能(4)

◆ブラウジング機能上のマシンの役割

役割	主な機能
マスタブラウザ (ドメインマスタブラウザ) (ローカルマスタブラウザ)	◆ワークグループのブラウズリストのマスタを保持する ◆クライアントからの問い合わせに応答する
バックアップブラウザ	◆マスタブラウザから、ブラウズリストの複製を受け取る ◆クライアントからの問い合わせに応答する
ポテンシャルブラウザ	◆ブラウザになる能力を持っているが、現在はなっていない
ノンブラウザ	◆ブラウザになる能力がない

- マスタブラウザとバックアップブラウザを総称してブラウザと呼称する

Samba のネットワーク機能(5)

- ブラウジング機能(続)
  - Sambaはマスタブラウザ、ポテンシャルブラウザ、非ブラウザとして機能させられる
    - 優先マスタブラウザ
      - preffered master
    - ドメインマスタブラウザ
      - 複数IPサブネットのブラウズリストを統合
        - domain master
    - ブラウザ選定の際の優先度を指定
      - local master

Sambaのクライアント機能(1)

- smbclientコマンド
  - ftpライクなインタフェースでWindowsの共有に接続
  - smbfsなどが使えないケースでも最低限ファイル転送はできる

```
% smbclient //megumi/temp -U monyo
Password:
Domain=[HOME] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: ¥> dir
  .                                   DA        0  Wed Oct 22 01:34:01 2003
  ..                                  DA        0  Wed Oct 22 01:34:01 2003
  CHARTEST                             D        0  Sat Apr 26 01:09:40 2003
  rpcdump.exe                          A   667264  Wed Sep 17 16:23:56 2003

                39252 blocks of size 1048576. 27090 blocks available
smb: ¥> get rpcdump.exe
getting file rpcdump.exe of size 667264 as rpcdump.exe (823.8 kb/s) (average 823.8 kb/s)
```

Sambaのクライアント機能(2)

- netコマンド(Samba 3.0)
  - UNIX側からWindowsサーバの各種管理が可能
    - ユーザ管理、グループ管理、共有管理など

```
# net user ADD monyo -S server -U Administrator
Password:
Added user monyo
```

ユーザの追加

- nmblookupコマンド
  - nslookupライクにNetBIOS名の問い合わせが可能

Sambaの設定応用編

ドメインコントローラ機能

LDAPによる認証統合

メンバサーバ機能

Winbindによる認証統合

SambaのPDC機能(1)

- Windows NT Server 4.0相当のPDC機能を実装
  - 実装レベルは非常に高い
    - domain logons
  - NTドメインのリプレースが可能
    - PDC,BDCともにSambaで構築
    - CAL不要のドメイン構築が可能
  - 実装しているのは、あくまでNTドメインの機能
    - Active DirectoryのDC機能は未実装
  - SAMの同期をはじめ、実装していない機能もある

SambaのPDC機能(2)

- PDC機能を実装(続)
  - ユーザ認証
  - コンピュータの認証
    - Windows NT系マシンの「ドメイン参加」をサポート
    - セキュアチャネルの構築

SambaのPDC機能(3)

- PDCの機能を実装(続)
  - ユーザプロファイル
    - ユーザ環境を一元管理
      - logon drive / logon path - ホームディレクトリ
      - logon home
    - ログオンスクリプトサポート
      - logon script
  - システムポリシー
    - クライアント端末のレジストリを一元設定
      - NETLOGON共有にファイルを配置することでサポート

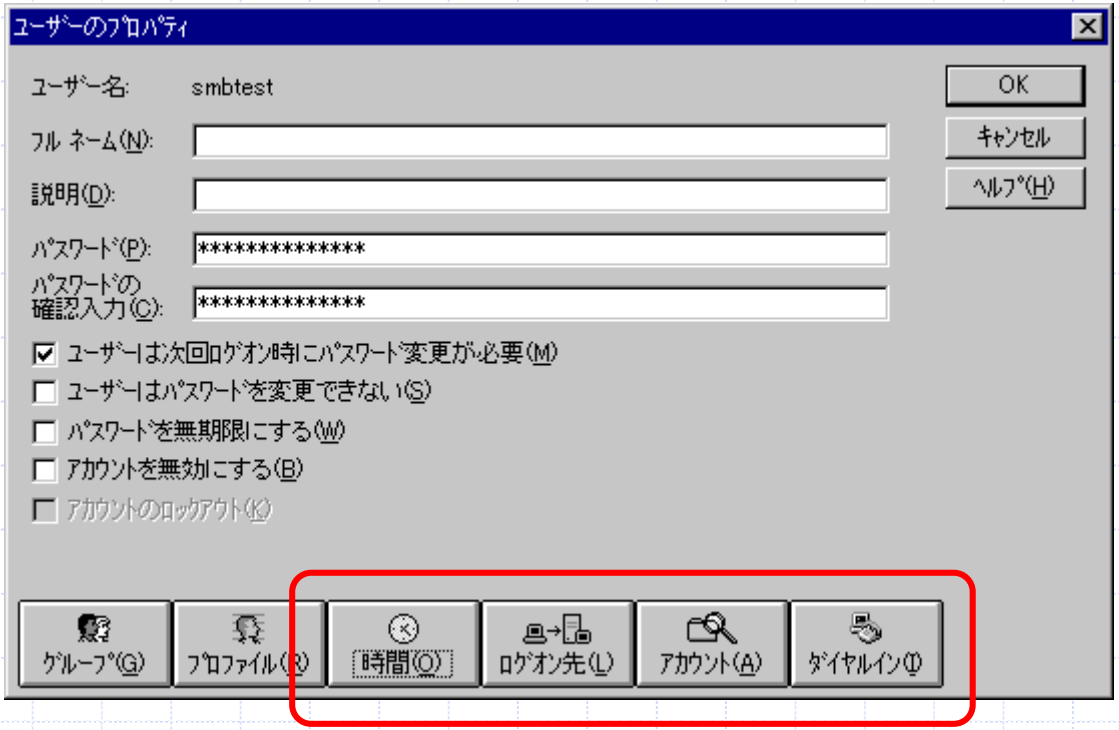
SambaのPDC機能(4)

- 実装していない機能
  - SAMの同期
    - Windows NTのドメインコントローラとのSAM同期機能
      - Windows NTのPDC＆SambaのBDCという構成はとれない
        逆も同様。net vampire コマンドで情報の吸い上げは可能
  - グローバルグループ
    - Samba 2.2系列では既定のグローバルグループのみ
      - domain admin group / domain guest group
    - Samba 3.0以降ではグローバルグループをサポート
      - net groupmap コマンド

SambaのPDC機能(5)

実装していない機能(続)

- 細かいユーザ属性の一部
  - 実質的には、あまり問題とはならない?
    - ログオン時間
    - ログオン先
    - ローカルアカウント
    - ダイヤルイン



SambaのPDC機能(6)

- 基本的な設定(必須)
  - PDCとして構成
  - ドメイン名
- プロファイル情報
  - ホームディレクトリ
  - プロファイルの場所
  - ログオンスクリプト

基本的な設定

```
[global]
  workgroup = DOMAIN
  domain logons = yes
  (domain master = yes)
  os level = 32

  (logon drive = h:)
  (logon home = ¥¥%N¥%U)
  (logon path = ¥
    ¥¥%N¥%U¥profile)
  (logon script = logon.cmd)

[netlogon]
  (profile acls =)
  (nt smb support = no)
```

SambaのPDC機能(7)

設定方法(続)

- マシンのドメインへの追加

手作業でUNIXアカウントを作成

```
# useradd -d /dev/null -s /bin/false w2kpro1$
```

手作業でSambaアカウントを作成

```
# smbpasswd -a -m w2kpro1
```

add user(machine) sciprt を設定

```
[global]
  add user(machine) script = ¥
  /usr/sbin/useradd -d /dev/null -s /bin/false %u
```

Windows側のGUIで追加

SambaマシンのRoot権限を持つuidのSambaアカウントとして認証して参加

(注)Samba 2.2系列のドメインにWindows XPが参加する場合は、RequireSignOrSealレジストリを変更する

SambaのPDC機能(8)

設定方法(続)

グローバルグループの追加

Samba 2.2の場合

パラメータで対応グループを設定
設定可能なグループは以下の2つ

```
[global]
  domain admin group = domadm
  domain guest group = domguest
```

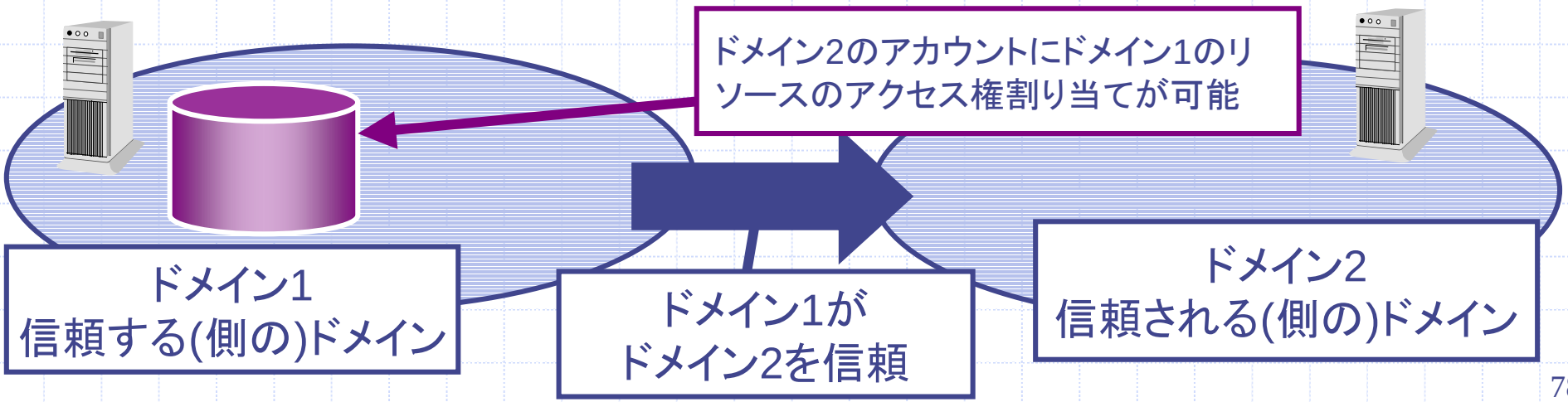
Samba 3.0の場合

net groupmap コマンドを使用
任意のグループを設定可能


```
#net groupmap modify NTgroup='Domain Admins' UNIXgroup=domadm
#net groupmap add rid=1000 NTgroup='Group 1' UNIXgroup=group1
#net groupmap list | grep "Group 1"
Group 1 (S-1-5-21-1108995562-3116817432-1375597819-1000) -> group1
```

SambaのPDC機能(9)

- 信頼関係の構築(Samba 3.0から)
  - 異なるWindowsドメインのアカウントに対してアクセス権を割り当てる機能
    - 大規模環境では必要なことも
  - 明示的な一方向の信頼関係のみサポート



SambaのPDC機能(10)

- 信頼関係の構築(続)
  - 信頼される側のドメインとして設定

```
root# smbpasswd -a -i rumba ←ドメイン名(ここではrumba)を入力
New SMB password: ← 信頼関係用のパスワード
Retype SMB password: ← 再度入力
Added user rumba$
```

  - 信頼する側のドメインとして設定

```
root# net rpc trustdom establish nt4dom2
Password: ←予めWindows側で設定された信頼関係締結用パスワード
Success!
```

SambaのBDC機能

- 設定自体は簡単
  - 以下の箇所以外はPDCと基本的に同一
    - domain master = Noにする
    - SIDについては、PDCのものを複製する

```
[global]
  domain master = No
  domain logons = Yes
```

```
# smbpasswd -S
# net rpc getsid
```

- 認証情報の同期は難しい
  - smbpasswdファイルを用いる場合はrsyncなど
  - LDAPによる認証の一元管理を推奨
  - WindowsのPDCとのSAMの同期はできない

LDAPによる認証統合(1)

- Sambaアカウントの情報をLDAPで管理
  - 複数Sambaサーバ間でSAM(相当)を共有
    - ドメイン構築時に有用
      - ただし、非ドメイン環境でも使用可能
    - 既存のLDAP認証基盤との統合が可能
      - ただし、パスワード情報自体の統合はできない
  - Samba 3.0では複数の認証方式の組み合せも可
    - Samba 2.2でLDAP認証を用いる場合、それ以外の方式との併用はできない
      - --with-ldapsamオプションでconfigure時に指定(Samba 2.2)

LDAPによる認証統合(2)

LDAP認証の設定

- Samba 3.0.0 & OpenLDAPを例に説明、パスなどは標準的なもの

UNIX(LDAP)サーバ側

1. Sambaスキーマの追加
/etc/ldap/slapd.confの設定

2. 管理用オブジェクトの作成

```
ou=Users,dc=samba,dc=local
ou=Groups,dc=samba,dc=local
ou=Computers,dc=samba,dc=local
```

3. Sambaが使用する管理用DNの設定
DN自身の作成と、必要な権限の割り当て

```
cn=admin,ou=users,dc=samba,dc=local
```

Sambaサーバ側

1. smb.confファイルの設定

```
[global]
  passdb backend = ldap://ldap.samba.local
  ldap admin dn = cn-admin,dc=samba,dc=local
  ldap suffix = dc=samba,dc=local
  ldap user suffix = ou=Users
  ldap group suffix = ou=Groups
  ldap machine suffix = ou=Computer
  ldap ssl = off
```

2. 管理用DNのパスワードを設定

```
smbpasswd -w
```

LDAPによる認証統合(3)

- LDAP認証の設定(続)
  - Samba 3.0.0 & OpenLDAPを例に説明、パスなどは標準的なもの
  - ユーザ、コンピュータの管理
    - pdbeditコマンドで追加、修正、削除
    - LDAP経由で直接編集も可能
  - 既存環境からの移行
    - パスワード認証時にそのパスワードをLDAPに格納
      - シームレスな移行が可能に
        - ldap passwd sync
    - smbldap-toolsの使用

```
# pdbedit -a -u monyo
# pdbedit -a -m w2kpro2
```

LDAPによる認証統合(4)

LDAP認証の設定(続)

- smbldap-toolsによる簡易設定
  - smbldap-tools-0.8.1 - http://samba.idealx.org/ Samba 3.0に対応
  - UNIXユーザ管理もLDAPで行うことが前提

```
smbldap-groupadd.pl    : to add a new group
smbldap-groupdel.pl    : to delete a group
smbldap-groupmod.pl    : to modify a group (mostly used to add user to a group)
smbldap-groupshow.pl   : to view a group
smbldap-useradd.pl     : to add a new user
smbldap-userdel.pl     : to delete a user
smbldap-usermod.pl     : to modify an user datas
smbldap-usershow.pl    : to view an user datas
smbldap-passwd.pl      : to sync passwd (Unix and Samba)
smbldap-populate.pl    : to add a builtin ldif to initialize your LDAP master for
                         smbldap use, or to add a specified ldif
```

Samba のメンバサーバ機能(1)

- Windowsドメインへの参加
  - ドメインのメンバサーバとして機能
    - ドメインのアカウント、グループによる認証
      - Windows側での認証の一元管理が可能
        - マシンアクセス時の認証
        - ACLなどによるファイル保護
      - セキュアチャネルによる保護
    - UNIX上に別途アカウントは必要
      - Winbindを用いることで不要に

Samba のメンバサーバ機能(2)

- Windowsドメインへの参加
  - WindowsのCALが必要
    - Active Directoryを構築している場合
      - Windows NTで構築したドメインでは不要
  - Samba 2.2はNT互換の実装
    - ドメイン参加にはNetBIOSが必須
    - Samba 3.0はWindows 2000互換の参加が可能
      - Kerberosプロトコルによる参加を実装

Samba のメンバサーバ機能(3)

- 設定方法
  - Windows NT互換の参加

1. smb.confの設定
2. smbd の停止

```
[global]
  workgroup = domain
  securiy = domain
```

Windows側のGUIで
手作業でマシンアカウントを作成

smbpasswdコマンドを実行

```
# smbpasswd -j DOMAIN -r PDCname
```

smbpasswd かnet コマンドを実行

```
# smbpasswd -j DOMAIN -r PDCname ¥
    -U Admin%Admin_password
```

```
# net rpc join -S PDCname ¥
    -U Admin%Admin_password
```

Samba のメンバサーバ機能(4)

設定方法(続)

Windows 2000互換の参加方法

1. smb.confの設定

```
[global]
  realm = <ADのドメイン名(大文字)>
  security = ADS
  encrypt passwords = yes
```

2. /etc/krb5.confの設定

```
[realms]
  W2K.HOME.MONYO.COM = {
 ↑Active Directoryのドメイン名
   kdc = miyu.w2k.home.monyo.com
}       ↑DCのFQDN名
```

3. kinitコマンドの実行

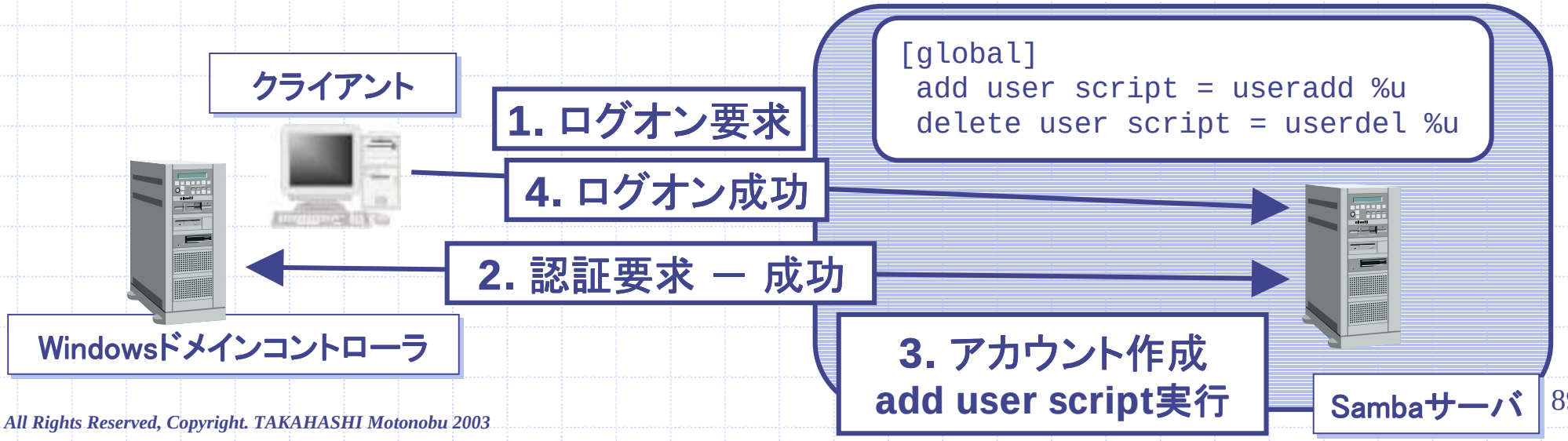
```
# kinit administrator
Password for administrator@W2K.HOME
.MONYO.COM:
```

4. netコマンドの実行

```
# net ads join
Joined 'MAPLE' to realm 'W2K.HOME.MONYO.COM'
```

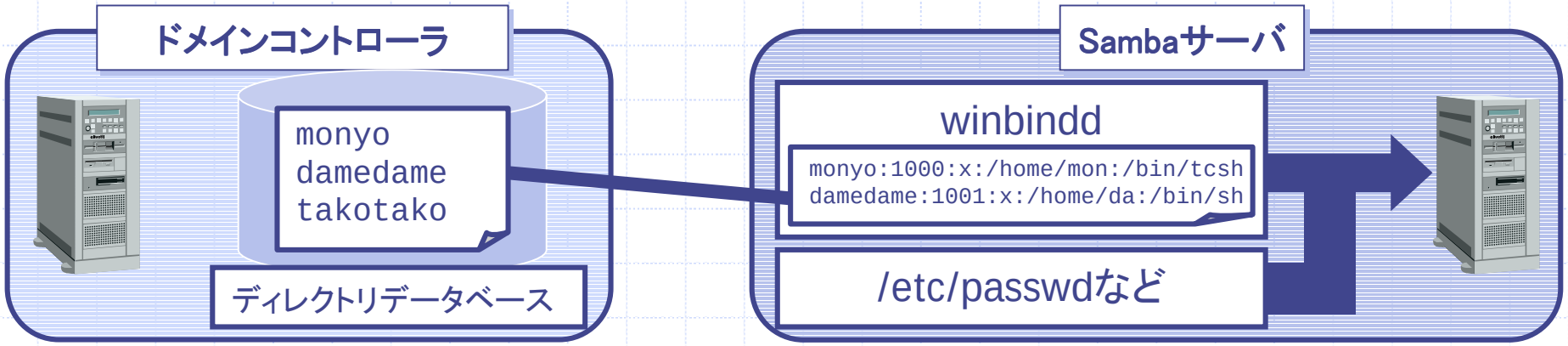
Samba のメンバサーバ機能(5)

- アカウントの自動作成、削除
  - Sambaサーバ側アカウントの自動管理が可能
    - Windowsドメイン側では認証のみ可能
      - アカウント情報自体の提供はできない
      - 認証したアカウントに対応するUNIXアカウントが必要



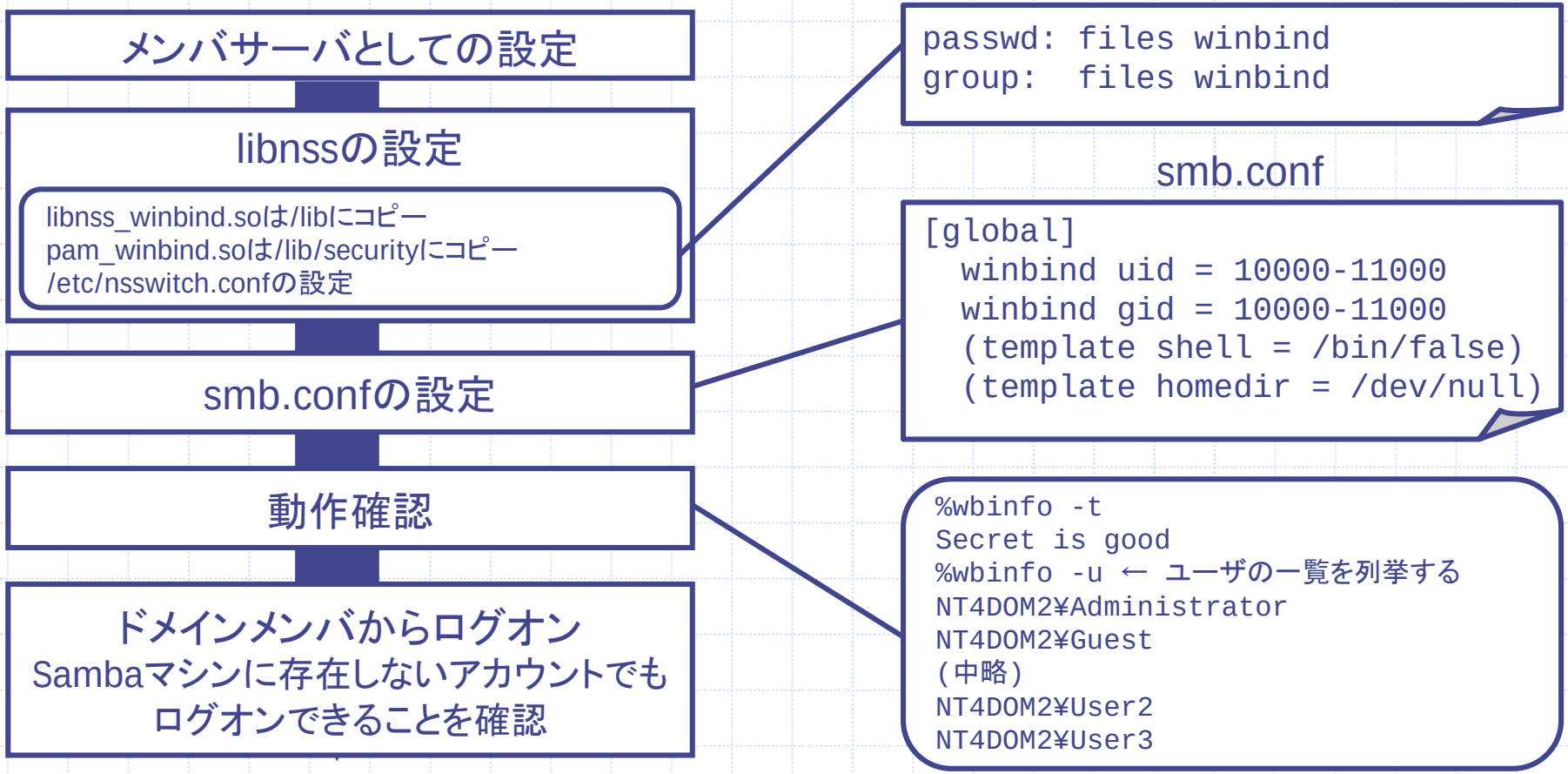
Winbindによる認証統合(1)

- UNIX上でのアカウント生成を不要に
  - メンバサーバ機能上に構築
  - Windowsドメイン上のアカウントに対応するUNIXアカウントの情報を提供する
    - libnss機能を使用
    - UNIX上でのアカウント作成が不要



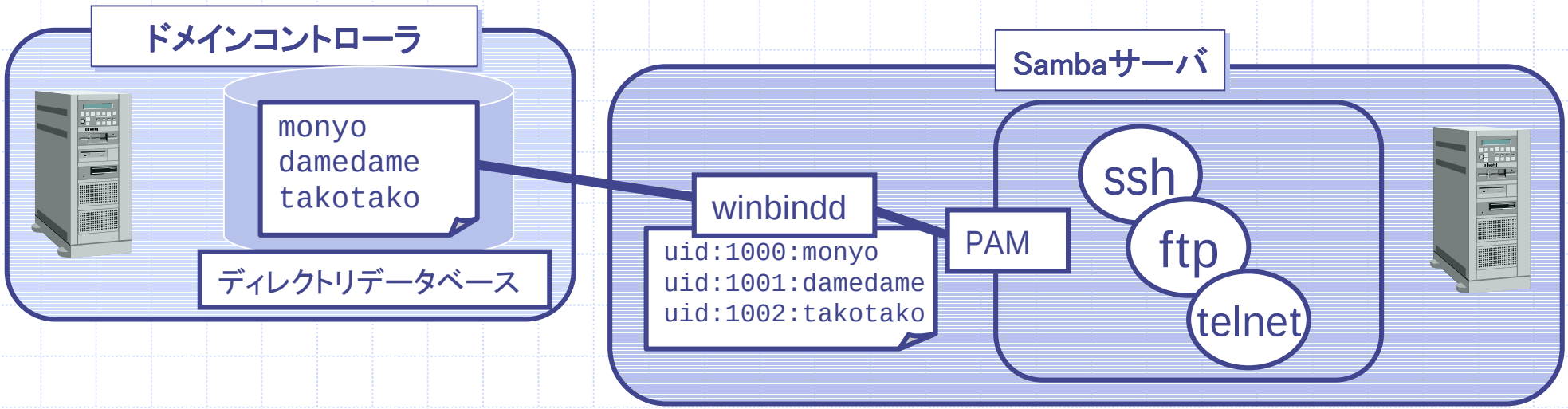
Winbindによる認証統合(2)

Winbindの設定



Winbindによる認証統合(3)

- Samba以外のプロダクトへの認証機能提供
  - PAMを経由することで、Samba以外のプロダクトに対する認証提供も可能
    - UNIX上の認証全体をWindows側に統合できる



Winbindによる認証統合(4)

- Samba以外のプロダクトからの使用
  - PAM経由で使用する
    - ホームディレクトリなどの作成は別途要考慮
      - template shellパラメータも要設定
        デフォルトシェルは/bin/false

Winbindの設定

PAMの設定
OSによって設定方法は多少異なる

telnetなどでログオン
Windowsドメインのパスワードで認証できることを確認

/etc/pam/system-auth
(Red Hat Linuxの場合)

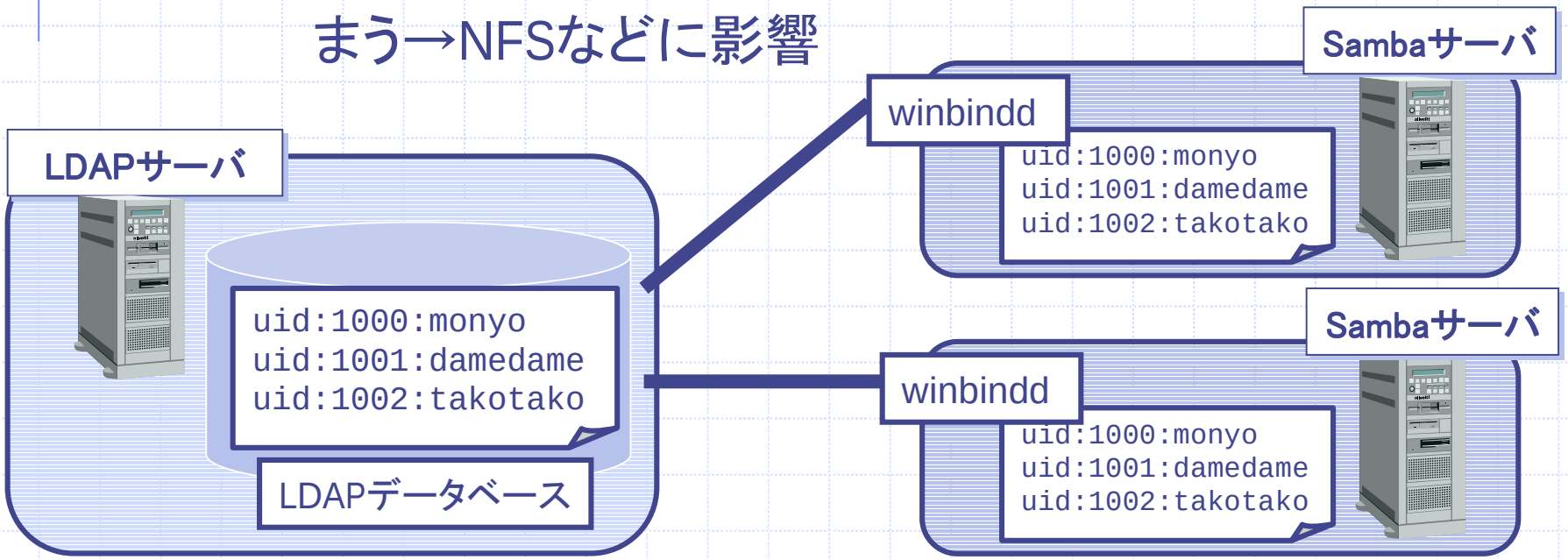
```
#%PAM-1.0
(省略)
auth     sufficient   /lib/security/pam_winbind.so ← 追加
auth     required     /lib/security/pam_deny.so
(省略)
account required     /lib/security/pam_unix.so
account sufficient   /lib/security/pam_winbind.so ← 追加

session required     /lib/security/pam_unix.so
session required     /lib/security/pam_mkhomedir.so ¥
      skel=/etc/skel umask=0022 ← 場合によって追加
```

```
% telnet linux1
login: nt4dom2¥user1 ← 大文字/小文字は関係ない
```

Winbindによる認証統合(5)

- idmapによるUID/GIDの統合
  - Windowsアカウントに対応するUNIXアカウントのUIDがマシン毎に異なる問題を解消
    - この機能を使わないとマッピングがマシン毎に異なってしまう→NFSなどに影響



Winbindによる認証統合(6)

idmapによるUID/GIDの統合(続)

- 設定方法(ldapsamと重複する部分は省略)

UNIX(LDAP)サーバ側

1. LDAPの設定

Sambaスキーマの追加
管理用オブジェクトの作成
管理用DNの設定

2. 管理用オブジェクトの作成

```
ou=idmap,dc=samba,dc=local
```

Sambaサーバ側

1. smb.confファイルの設定

```
[global]
  ...
  ldap admin dn = cn=admin,dc=samba,dc-local
  idmap backend = ldapsam:ldap://ldap.samba.local/
  ldap idmap suffix = ou=idmap,dc=samba,dc=local
  idmap uid = 40000-50000
  idmap gid = 40000-50000
```

2. 管理用DNのパスワードを設定

SFUの機能紹介

機能紹介

NISサーバ

パスワード同期

SFUの機能紹介(1)

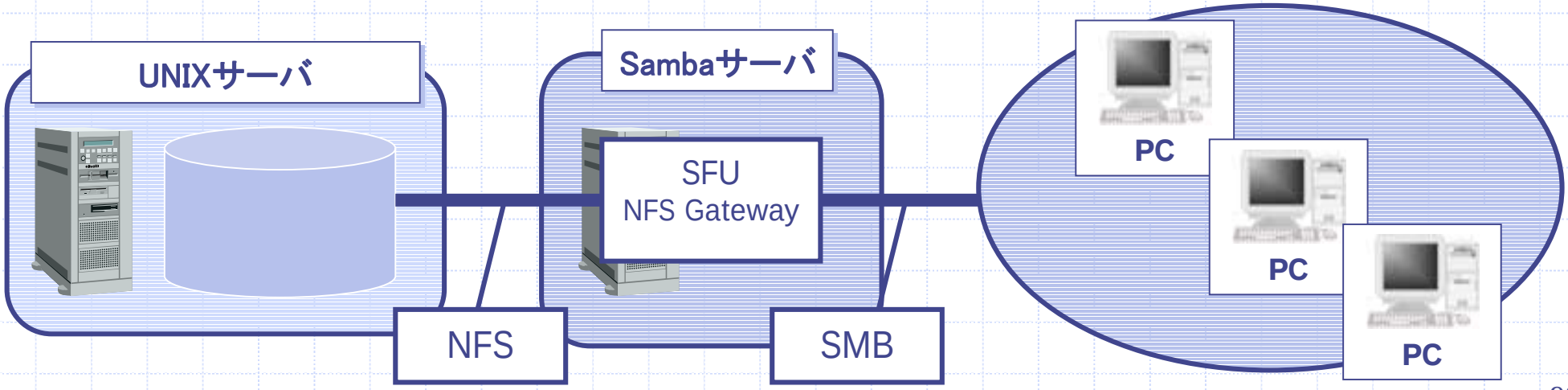
コンポーネント一覧

- インストール時に選択可能

分類	コンポーネント名	機能概要
NFS	NFSコンポーネント	NFSサーバ、クライアント機能、NFSゲートウェイ機能(NFSサーバ上のリソースをSMBプロトコルを用いてクライアントに公開する)を提供
NFS	NFS認証ツール	NFS認証サーバやpcnfsdなどを提供
認証	NISサーバ	Active DirectoryのDCをNISサーバとして動作させる機能を提供
認証	パスワード同期	Windows側とUNIX側とのパスワードを同期させる機能を提供
その他	リモート接続コンポーネント	telnetやrshのクライアントおよびサーバ機能を提供
その他	ActiveState Active Perl 5.6	Windows上でPerlの実行環境を提供
Interix	ユーティリティ	UNIXの基本的なコマンドやperlなどを提供
Interix	Interix GNUコンポーネント	GPLのツール群や開発環境などを提供
Interix	Interix SDK	X11R5の開発環境などを提供

SFUの機能紹介(2)

- NFSサーバ機能
  - Windows上の共有をNFS経由でマウント可能
- NFSゲートウェイ機能
  - NFS共有をSMBクライアントから利用できるように



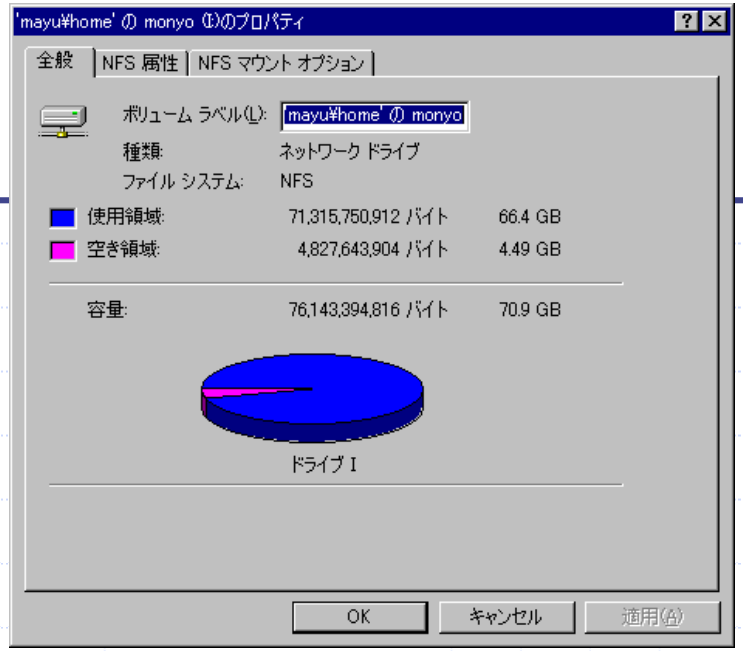
SFUの機能紹介(3)

NFSクライアント機能

- NFS共有をSMB共有のようにマウント可能

```
C:¥>mount -o lang=euc-jp -o pcnfs=shiori -u:monyo ¥¥mayu¥home¥monyo i:
Password:
i: は ¥¥mayu¥home¥monyo に正常に接続しました

コマンドは正常に終了しました。
```



- ユーザ認証は、pcnfsdかNISで行う
- 日本語ファイル名処理は一部不備がある

SFUの機能紹介(4)

- NISサーバ機能
  - Active Directoryのアカウントを NIS経由でUNIXから使用可能に
    - Active Directoryのスキーマを拡張して実装
    - Windowsのパスワード情報をUNIX側から参照

SFUの機能紹介(5)

- NISサーバ機能(続)
  - パスワードに関する注意
    - UNIX側でパスワードを変更するとWindows側と同期しなくなる
      - パスワード変更は、必ずWindows側で
    - パスワードはcrypt形式
    - nismapで新規追加したアカウントのパスワードはランダムな文字列
      - 運用の前に、一度パスワード変更が必要

SFUの機能紹介(6)

- NISサーバ機能(続)
  - NISマップの作成
    - nismapコマンドを使用(GUIからは不可)
      - 一部のマップ(passwdなど)はGUIから設定可能

```
C:¥>nismap add -e "local4:iJa/W9kDv6lDY:10004:10004::/home/local4:/bin/tcsh" -a W2K2 passwd

動作 = 追加中 マップ = 'passwd'...

成功
オブジェクトを Active Directory に追加します。
オブジェクト = 'local4'
オブジェクト クラス = 'User'
コンテナ = 'LDAP://localhost/CN=Users,DC=W2K2,DC=HOME,DC=MONYO,DC=com'.
```

NISマップ名を指定

    - nis2adコマンドで既存のNISマップからの移行も

SFUの機能紹介(7)

- パスワード同期機能
  - WindowsとUNIXのパスワード変更を同期
    - Windows →UNIX
      - UNIX側でssodというデーモンを実行
    - UNIX→Windows
      - PAMにpam_sso.so.1を追加
  - しかし「使えない」機能
    - サポートプラットフォームが古い
      - ssodなどのソースは添付されているが、そのままコンパイルしても動作せず

SFU 3.0のパスワード同期サポートプラットフォーム

Solaris 7 (SPARC版)
Red Hat Linux 7.1(Intel版のみと思われる)
AIX 4.3.3(WindowsからUNIXへの同期のみ)
hp-ux 11.00 / HP-UX 10.20

まとめ

- UNIXとWindowsとの共存におけるオプションとそのメリット、デメリットについて理解する
- Sambaの機能について理解する
  - 基本的な機能
  - 高度な機能
- SFUの機能について理解する