

# Bürger erster Klasse: Integration der Windows- und Linux/UNIX-Welt in OpenSolaris

**Franz Haberhauer (Franz.Haberhauer@Sun.com)**

UVM Lizenz für die freie Nutzung unveränderter Inhalte

*Bei der Integration heterogener Linux/UNIX- und Windows-Infrastrukturen stellen die unterschiedliche Konzepte zur Authentifizierung von Benutzern eine besondere Herausforderung da. UNIX verwendet für Identitäten die POSIX UID, Windows die SID.*

*In OpenSolaris wurden im Kernel ein vereinheitlichtes Identitätsmodell mit einer Namensabbildung zwischen UIDs und SIDs eingeführt. Zur Implementierung eines kernelbasierter CIFS/SMB-Service bzw. allgemeiner Multi-Protokoll Filesyserver wurde dieses Konzept im VFS und ZFS für verallgemeinerte Zugriffskontroll-Mechanismen genutzt.*

*Bei der Nutzung von LDAP als Nameservice wird ein Schema gemäß RFC2307bis zugrunde gelegt, während Windows sich auf das Schema im Active Directory stützt. Neben dem seit langen vorhandenen "Native-LDAP" Nameservice-Backend auf Basis des RFC2307bis wird ein Nameservice-Backend entwickelt, der direkt ein Active Directory nutzen kann.*

*Ende 2007 wurden in OpenSolaris im Kernel Komponenten integriert, die eine weitergehende Integration zwischen der UNIX- und der Windows-Welt realisieren, als das mit den bisherigen Lösungen im Userspace möglich war. Dieser Vortrag stellt Architektur und Einsatzmöglichkeiten vor.*

## Protokolle für Filesharing in heterogenen Umgebungen

Zum Filesharing werden heute primär zwei Protokolle verwendet: NFS und CIFS.

NFS (Network File System) war Mitte der 80er-Jahre von Sun entwickelt und 1989 in der Version 2 als RFC 1094 standardisiert worden. Die Version 3 brachte 1995 (RFC 1813) nicht nur die Unterstützung großer Dateien (64 Bit), sondern auch deutliche Performance-Verbesserungen, insbesondere durch asynchrones Schreiben von Clienten zum Server. NFS wurde zum gebräuchlichsten Filesharing-Protokoll unter UNIX - nicht zuletzt, weil Implementierungen in allen gängigen UNIX-Varianten enthalten sind. Auch für zahlreiche andere Betriebssystemen, u.a. VMS oder MVS gibt es NFS-Implementierungen. Für Windows gibt es NFSv3-Implementierungen von verschiedenen Anbietern. Mittlerweile liefert auch Microsoft eine NFS-Implementierung als Teil der Windows Services for UNIX (SFU)<sup>1</sup>.

Unter UNIX erfolgt die Identifikation von Benutzern bzw. Benutzergruppen über Namen, die fuer die interne Nutzung in ganzzahlige Werte umgesetzt werden: die UID bzw. GID. Über die Zeit hat ist die Länge dieser Datentypen gewachsen: von vorzeichenbehafteten 8 Bit im 6th Edition UNIX über 16 Bit auf heute 32 Bit. An der Struktur eines flachen Namensraums innerhalb eines

lokalen Systems oder einer Domain hat sich allerdings nichts geändert.

Unter Windows werden dagegen universell eindeutige Bezeichner, sogenannte Security Identifier (SIDs) verwendet. Hier ein Beispiel:

S-1-5-12-7623811015-3361044348-030300820-1013

in dieser Struktur bedeutet:

S – Die Zeichenkette ist ein SID

1 - Revision level

5 - 48-bit Identifier Authority Value (5 = "NT Authority")

12-7623811015-3361044348-030300820 – Sequenz von Subauthorities zur Identifikation der Domain oder des lokalen Systems

1013 - 32-bit Relative Identifier (RID) innerhalb der oben definierten Domain.

Damit ist offensichtlich, dass für heterogenes Filesharing eine wechselseitige Abbildung der Benutzer-Credentials erforderlich ist. Bis zur Version 3 verwendete NFS ein einfaches Authentifizierungsverfahren basierend lediglich auf der Angabe der UNIX/POSIX uid/gid durch Clients. Im seit langen auf vielen Plattformen verfügbaren Secure NFS ist dagegen eine starke Authentifizierung möglich auf der Basis von Diffie-Hellman oder Kerberos – es ist sogar eine Verschlüsselung des Datenverkehrs möglich. Seit der NFS Version 4 (RFC 3010 aus dem Jahr 2000) ist starke Authentifizierung in jeder konformen Implementierung enthalten, allerdings muss zur Nutzung unter Windows dort Software installiert und konfiguriert werden.

Nicht zuletzt deswegen ist in heterogenen Umgebungen die Nutzung des in Windows enthaltenen CIFS (Common Internet File System) verbreitet, eines Protokolls, das 1996 von Microsoft als erweiterte Version von SMB (Server Message Block) eingeführt wurde. SMB war 1984 von der IBM als Filesharing-Protokoll auf der Basis von NetBIOS eingeführt worden. Während NFS spezifisch als Netzwerk-Dateisystem entwickelt wurde, umfaßt CIFS – neben dem Druckdienst - eine Reihe von Subprotokollen und weiteren Diensten, die wesentlich stärker mit dem Betriebssystem verzahnt sind als das bei NFS der Fall ist. Neben der Implementierung von CIFS in Windows ist Samba<sup>2</sup> eine populäre Implementierung unter UNIX/Linux. Darüber hinaus gibt es Implementierungen in Network Attached Storage (NAS) Appliances/Filern, die unter spezialisierten Betriebssystemen laufen. Sun Microsystems hatte 2005 die Rechte an der NAS-Technologie von Procom Technology erworben – und damit einen CIFS-Stack, der in der Sun StorageTek 5000 Produktlinie eingesetzt wurde.

Bei der Implementierung eines CIFS-Servers unter UNIX/Linux sind die unterschiedlichen Benutzer-Credentials nicht die einzige Herausforderung. Windows kennt andere andere Dateiattribute als die unter UNIX gebräuchlichen. Die Eindeutigkeit von Dateinamen bei Gross-/Kleinschreibung, spezielle – wenn auch selten verwendete Funktionalitäten – wie Alternate Data Streams sind weitere Herausforderungen. In NFS Version 4 wurde einige Interoperabilitätsthemen adressiert – so eine verallgemeinerte Authentifizierung über Strings und die Einführung von Access Control Lists (ACLs), auf der Basis des Windows NT Modells. ZFS als modernes lokales Dateisystem bietet jetzt eine ausgezeichnete Plattform für Fileserver unabhängig vom Protokoll, da ZFS spezifische Funktionalitäten zur

Interoperabilität enthält. ZFS unterstützt z.B. die NFSv4-ACLs und kennt verschiedene Modi zur Behandlung von Gross-/Kleinschreibung in Dateinamen.

Ein weiteres Thema sind Namensdienste. Solaris und viele andere UNIX-Implementierungen unterstützen als verteilten Verzeichnisdienst NIS (Network Information Service) und seit einigen Jahren auch LDAP, wobei als Schema hierbei das in NIS definierte Schema dient (RFC2307). Unter Windows ist Active Directory der integrierte Verzeichnisdienst. Microsoft bietet als Teil der Services for UNIX (SFU) Identity Management for UNIX, womit das POSIX schema unterstützt wird<sup>3</sup>. In OpenSolaris arbeiten mehrere Projekte<sup>4</sup> an einer weitergehenden Interoperabilität mit Active Directory, u.a. im Projekt Winchester an einem Nameservice-Backend für den Nameservice-Switch.

## Der Solaris CIFS Service

Im Oktober 2007 wurde in den Build 77 von OpenSolaris der Solaris CIFS Service in den Kern integriert – mit über 370.000 Lines of Code eines der größten Projekte überhaupt<sup>5</sup>. Mit Solaris Express 1/08 wurde der „Solaris CIFS Administration Guide“<sup>6</sup> ein umfassende Online-Dokumentation (AnswerBook) freigegeben.

Ein OpenSolaris-System kann damit Dateien sowohl über NFS wie über CIFS exportieren. Der Solaris CIFS Service kann dabei entweder im Workgroup Mode oder im Domain Mode betrieben werden. Während im Workgroup Mode die Authentifizierung von Benutzern lokal erfolgt, wird sie beim Domain Mode an den Domain Controller delegiert. Nachdem ein Benutzer erfolgreich authentifiziert ist, wird auf der Basis der SID ein Access Token erzeugt. Die Solaris Credentials wurden entsprechend erweitert. Im Dateisystem ZFS werden auch Windows ACLs und DOS-Dateiattribute unterstützt.

Die Implementierung in Solaris zeichnet sich dadurch aus, dass Benutzeridentitäten sowohl als UIDs/GIDs wie parallel dazu als SID verwaltet werden. Wird ein Benutzer durch den CIFS Service authentifiziert, wird die CIFS Identität über den idmap Service automatisch auf eine entsprechende UNIX Identität abgebildet. Existiert eine explizite Abbildung wird diese verwendet, ansonsten wird eine temporäre Abbildung, eine sogenannte Ephemeral ID, erzeugt, die bis zum nächsten Reboot gültig bleibt. Diese IDs werden nicht in Dateisystemen auf der Platte gespeichert. Falls eine ACL durch den CIFS Service abgelegt werden soll, wird die SID verwendet. SIDs werden in Form eines sogenannten FUIDs (Filesystem Unique Identifier)<sup>7</sup> gespeichert - einer 64 Bit Struktur aus einem Index in eine Tabelle mit SID Domain-Prefix und der RID bzw. UID/GID besteht. Solaris Utilities wie ls oder chmod unterstützen die Verwaltung von ACLs.

## Solaris als Multiprotokoll-Fileserver

In der Solaris-Distribution ist auch Samba enthalten. Zu einem Zeitpunkt kann allerdings nur ein CIFS Service aktiv sein, da für CIFS spezifische Ports belegt werden.

## Der Identity Mapping Service

Der Identity Mapping Service unterstützt drei Abbildungen zwischen SIDs und UID/GIDs

- Namensbasierte Abbildung: Ein Administrator bildet explizit Windows und Solaris Benutzernamen aufeinander ab.
- Ephemeral ID Abbildung: UID/GIDs werden dynamisch für SIDs erzeugt, für die es keine namensbasierte Abbildung gibt.
- Lokale SID Abbildung: Ein UID/GID wird auf eine algorithmisch lokal erzeugte SID abgebildet.

Über das Utility `idmap` können Namensbasierte Abbildungen erzeugt und verwaltet werden. Diese Abbildungen können unidirektional oder bidirektional angelegt werden, wobei Wildcards bei der Spezifikation möglich sind, z.B.

```
# idmap add 'winuser:*@example' 'unixuser:*'
# idmap add 'wingroup:*@example' 'unixgroup:*
```

zur namensgleichen bidirektionalen Abbildung aller Benutzer und Gruppen in

der Domain example. Explizite Abbildungen können auch über Dateien geladen werden in Samba- (smbusers rule-mapping) oder NetApps-Notation (usermap.cfg).

```
# idmap add -d winuser:sales@example unixuser:Sales
```

erzeugt eine unidirektionale Abbildung des Windowsnutzers „sales“ auf den Solaris-Nutzer „Sales“

Für die Ephemeral IDs wird genutzt, dass die POSIX UID/GIDs positive ganzzahlige Werte sind. Damit sind die Werte von 0x80000000 bis 0xffffffffe nicht belegt und können für Ephemeral IDs genutzt werden. Diese werden nach einem Reboot aufsteigend ab beginnend ab 0x80000000 vergeben.

## Konfiguration des Solaris CIFS Service

Soll ein Solaris Server als Fileserver in eine Windows Domain eingebunden werden, so muss Solaris als Active Directory (AD) Client konfiguriert werden. Hierzu muss das System als DNS Client eingerichtet sein und Kerberos Client konfiguriert werden. Dazu wird die AD Domain (z.B. DOMAIN.EXAMPLE) sowie der AD Domain Controller (z.B. dc.domain.example) in /etc/krb5/krb5.conf (krb5.conf(4)) eingetragen werden:

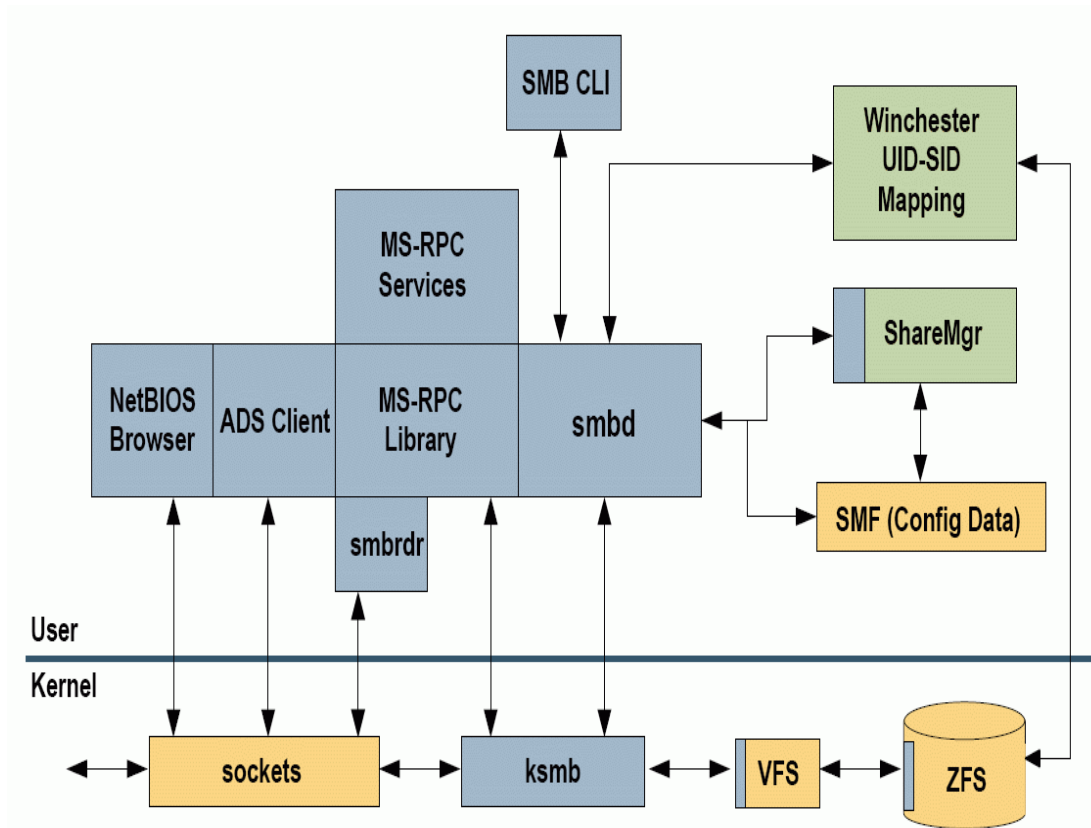
```
[libdefaults]
    default_realm = DOMAIN.EXAMPLE
[realms]
    DOMAIN.EXAMPLE = {
        kdc = dc.domain.example
        admin_server = dc.domain.example
        kpasswd_server = dc.domain.example
        kpasswd_protocol = SET_CHANGE
    }
[domain_realm]
    .domain.example = DOMAIN.EXAMPLE
```

Um das System in die AD Domain einzuklinken wird zunächst mit dem Kommando sharectl(1M), über das auch NFS verwaltet werden kann, durch das Setzen von Properties AD konfiguriert:

```
# sharectl set -p ads_domain=domain.example smb
# sharectl set -p ads_enable=true smb
```

Dann wird der CIFS Service gestartet in die Domain eingeklinkt:

```
# svcadm enable -r smb/server
# smbadm join -u admin-user domain.example
```



Architektur des Solaris CIFS Service<sup>8</sup>

Im nächsten Schritt werden CIFS Shares exportiert. Als lokales Filesystem eignet sich hierzu insbesondere ZFS, da ZFS wie oben bereits erwähnt erlaubt, Windows-Spezifika abzubilden wie DOS-Dateiattribute, ACLs oder auch die Eindeutigkeit von Dateinamen bei Gross-/Kleinschreibung. Dieser Modus wird beim Anlegen des ZFS Dateisystems spezifiziert - die Definition von Shares kann direkt über das Kommando `zfs(1M)` erfolgen:

```
# zfs create -o casesensitivity=mixed fs_for_cifs
# zfs set sharesmb=on fs_for_cifs
```

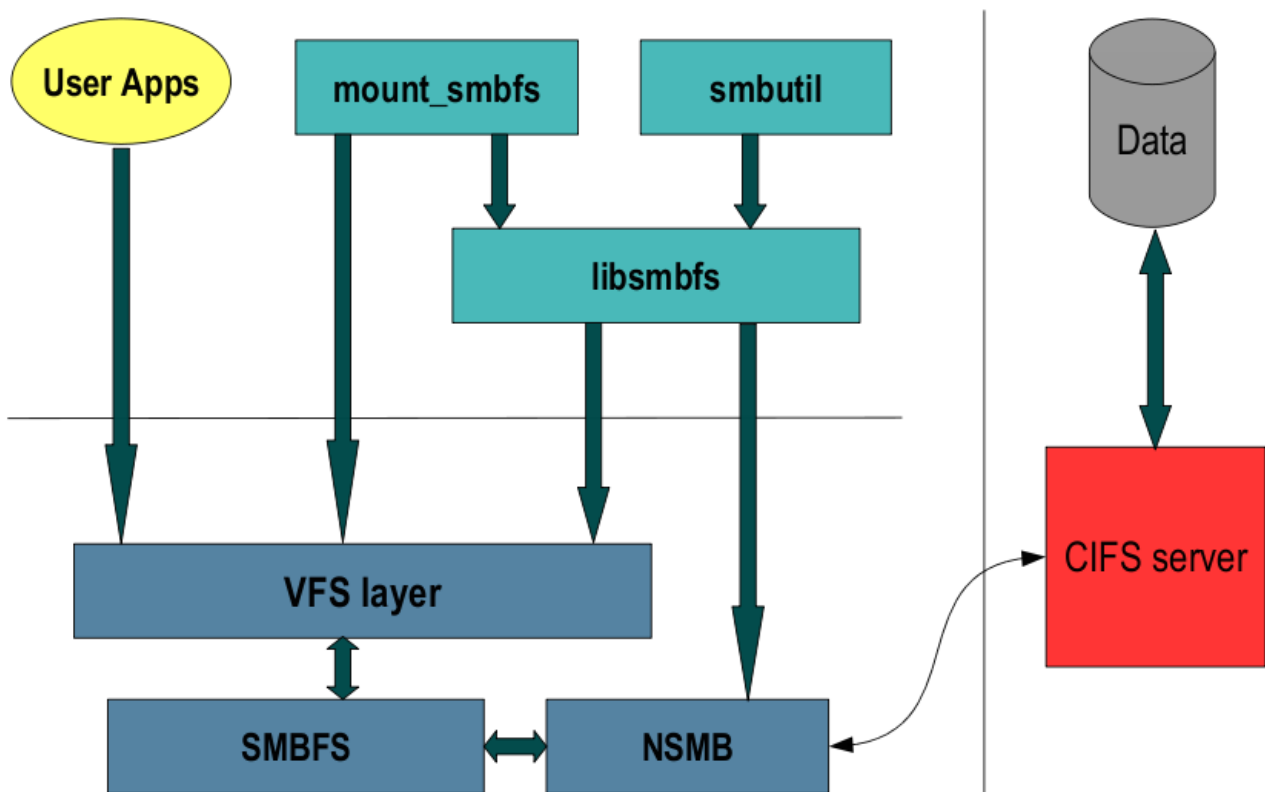
Dabei wird der Name des Dateisystems als Bezeichner für den Share verwendet. Ungültige Zeichen (z.B. „/“ in Pfaden wird ggf. durch „\_“ ersetzt). Es können auch explizit Namen für Shares spezifiziert werden:

```
# zfs set sharesmb=name=myshare fs_for_cifs
```

Für andere Dateisysteme erfolgt die verwaltung von CIFS Shares über das Kommando `sharemgr(1M)`.

## Der Solaris CIFS Client

Im Februar 2008 wurde in den OpenSolaris Build 84 ein CIFS Client integriert<sup>9</sup>. Davor waren mehrere Alpha- und Beta-Versionen verfügbar. Der CIFS Client ist als Solaris Virtual Filesystem implementiert. Ausgangsbasis waren die smbfs-Implementierung in Darwin/BSD und das VFS-Interface von OpenSolaris NFS. Die Phase 1 hat noch kleine Limitierungen, so können etwa ALCs nicht ausgewertet werden und es gibt keine Oplocks.



## Architektur des Solaris CIFS Client

Über das Kommando `smbutil(1)` kann abgefragt werden, welche Shares ein CIFS-Server exportiert:

```
$ smbutil view //franz@solarsystem
Password:
Share          Type          Comment
-----
netlogon       disk          Network Logon Service
ipc$           IPC           IPC Service (Samba Server)
tmp            disk          Temporary file space
public         disk          Public Stuff
home           disk          Home Directories
```

CIFS-Shares können dann wie andere Dateisysteme mit mount(1M) gemountet werden - siehe mount\_smbfs(1M):

```
$ mount -F smbfs //solarsystem/tmp /mnt
```

Interaktionen mit einem CIFS-Server erfordern eine Authentifizierung. Sofern Kerberos sowohl auf dem Client wie auf dem Server konfiguriert und ein Ticket-Granting Ticket vorhanden ist, wird dazu kein Passwort benötigt. Die Eingabe eines Passworts bei jeder Interaktion kann durch ein persistentes Passwort, das bis zu einem Reboot des CIFS-Client oder einem expliziten Logout gültig ist, vermieden werden.

```
$ smbutil login franz@solarsystem  
Password:
```

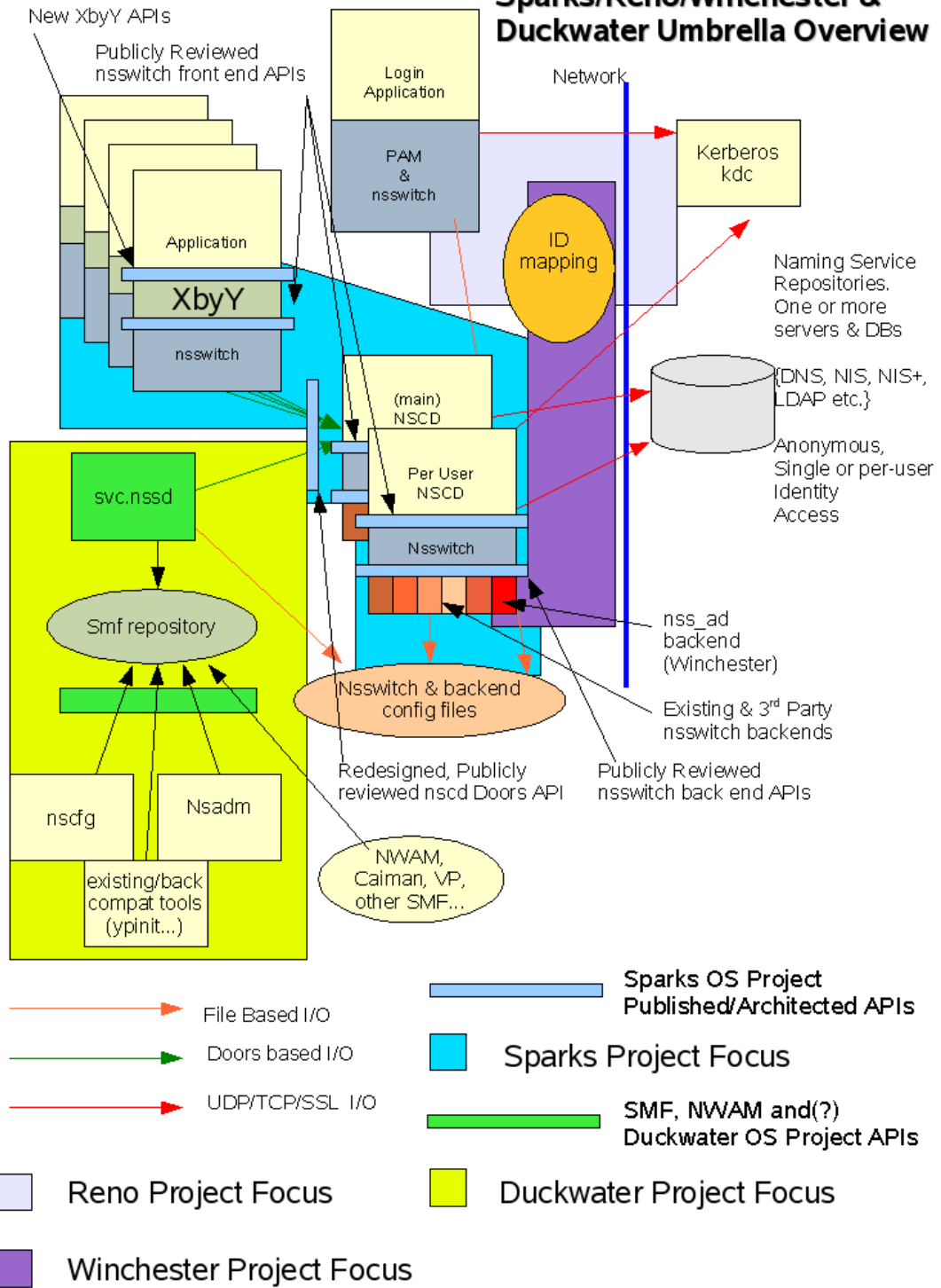
Persistente Passwörter können auch über das PAM-Modul pam\_smbfs\_login(5) gespeichert werden.

## Ausblick

In OpenSolaris wird im Rahmen verschiedener Projekte die Nameservice-Infrastruktur verallgemeinert und überholt. In diesen Projekten ist immer auch eine verbesserte Interoperabilität mit Windows/Active Directory ein wichtiger Faktor. Während das CIFS-Server Projekt für eine Rückportierung auf Solaris 10 zu groß und komplex ist, fließen Verbesserungen in der Infrastruktur auch in Solaris 10 Updates ein.

- Sparks: Nameservice-Switch-Infrastruktur. Neben einer Überarbeitung des Nameservice-Cache Demons ncsd Nameservice-Lookups. in diesem Projekt wird auch die benutzerspezifische Authorisierung von Nameservice-Lookups in SASL/GSS/Kerberos in einer mit dem Active Directory kompatiblen Art und Weise implementiert. Die erste Phase dieses Projektes wurde in OpenSolaris Build 50 und Solaris 10 8/07 integriert<sup>10</sup>.
- Winchester: Schema Abbildung und ID Mapping für die Interoperabilität mit Active Directory. hier wird auch an einem Nameservice-Backend für das Active Directory gearbeitet.
- Reno: Interoperabilität der Nameservices mit Login: Single Sign-On, Authentifizierung im Netzwerk, PAM etc.
- Duckwater: Vereinfachte Verwaltung der Nameservices

# Sparks/Reno/Winchester & Duckwater Umbrella Overview



## Projektübersicht<sup>11</sup>

## Zusammenfassung

Bei der Weiterentwicklung von Solaris ist die Interoperabilität mit Windows ein wichtiger Schwerpunkt. Durch die Integration der Windows Security Identifier und des CIFS-Service in den OpenSolaris Kern, die Implementierung eines CIFS-Clients als Virtuelles Filesystem und eine verbesserte Integration von Active Directory können OpenSolaris-Systeme sich nahtlos in heterogene Umgebungen einfügen.

- 1 Microsoft's Newest Version of Windows Services for UNIX Is Now Available Free of Charge  
<http://www.microsoft.com/presspass/press/2004/Jan04/01-15ServicesforUNIX2004PR.msp>  
<http://www.microsoft.com/windows/sfu/>
- 2 <http://www.samba.org>
- 3 Using Kerberos to Authenticate a Solaris 10 OS LDAP Client With Microsoft Active Directory, Wajih Ahmed and Baban Kenkre, März 2008  
[http://www.sun.com/bigadmin/features/articles/kerberos\\_s10.jsp](http://www.sun.com/bigadmin/features/articles/kerberos_s10.jsp)
- 4 <http://www.opensolaris.org/os/project/sparks/overview/>
- 5 [http://blogs.sun.com/amw/entry/cifs\\_in\\_solaris](http://blogs.sun.com/amw/entry/cifs_in_solaris)
- 6 Solaris CIFS Administration Guide <http://docs.sun.com/app/docs/doc/820-2429>  
docs.sun.com > Other Solaris Releases > Solaris Express > Administrator Collection >
- 7 Case Materials for PSARC 2007/064 Unified POSIX and Windows Credentials for Solaris, Mike Shapiro, November 2007  
<http://opensolaris.org/os/community/arc/caselog/2007/064/final-materials/spec-txt/>
- 8 Solaris gets serious about Windows interoperability with CIFS, Barry Greenberg, Januar 2008,  
[http://blogs.sun.com/barrygre/entry/solaris\\_gets\\_serious\\_about\\_windows](http://blogs.sun.com/barrygre/entry/solaris_gets_serious_about_windows)  
  
CIFS ... in Solaris, Alan Wright, November 2007,  
[http://blogs.sun.com/amw/entry/cifs\\_in\\_solaris](http://blogs.sun.com/amw/entry/cifs_in_solaris)
- 9 OpenSolaris Project: CIFS Client for Solaris <http://opensolaris.org/os/project/smbfs/>
- 10 Using Kerberos to Authenticate a Solaris 10 OS LDAP Client With Microsoft Active Directory, Wajih Ahmed and Baban Kenkre, März 2008,  
[http://www.sun.com/bigadmin/features/articles/kerberos\\_s10.jsp](http://www.sun.com/bigadmin/features/articles/kerberos_s10.jsp)
- 11 <http://www.opensolaris.org/os/project/sparks/overview/>