

Linux Client im Windows AD

Ein Erfahrungsbericht

RBG-Seminar WS 06/07
21.11.2006

Holger Kälberer

Übersicht

- Einleitendes: Infrastruktur und Dienste, was ist AD?

1. Authentifizierung

2. Home-Verzeichnisse

3. Drucken

Vorüberlegungen

- Anforderungen an den Client:
 - Komfort für Windows-Nutzer (USB-mounts, verknüpfte Extensionen, Zusammenspiel der Applikationen) → Desktop-Umgebung/-Distribution (ubuntu)
 - → Auswahl einer (!) Oberfläche (gnome/xfce)

Vorüberlegungen (Forts.)

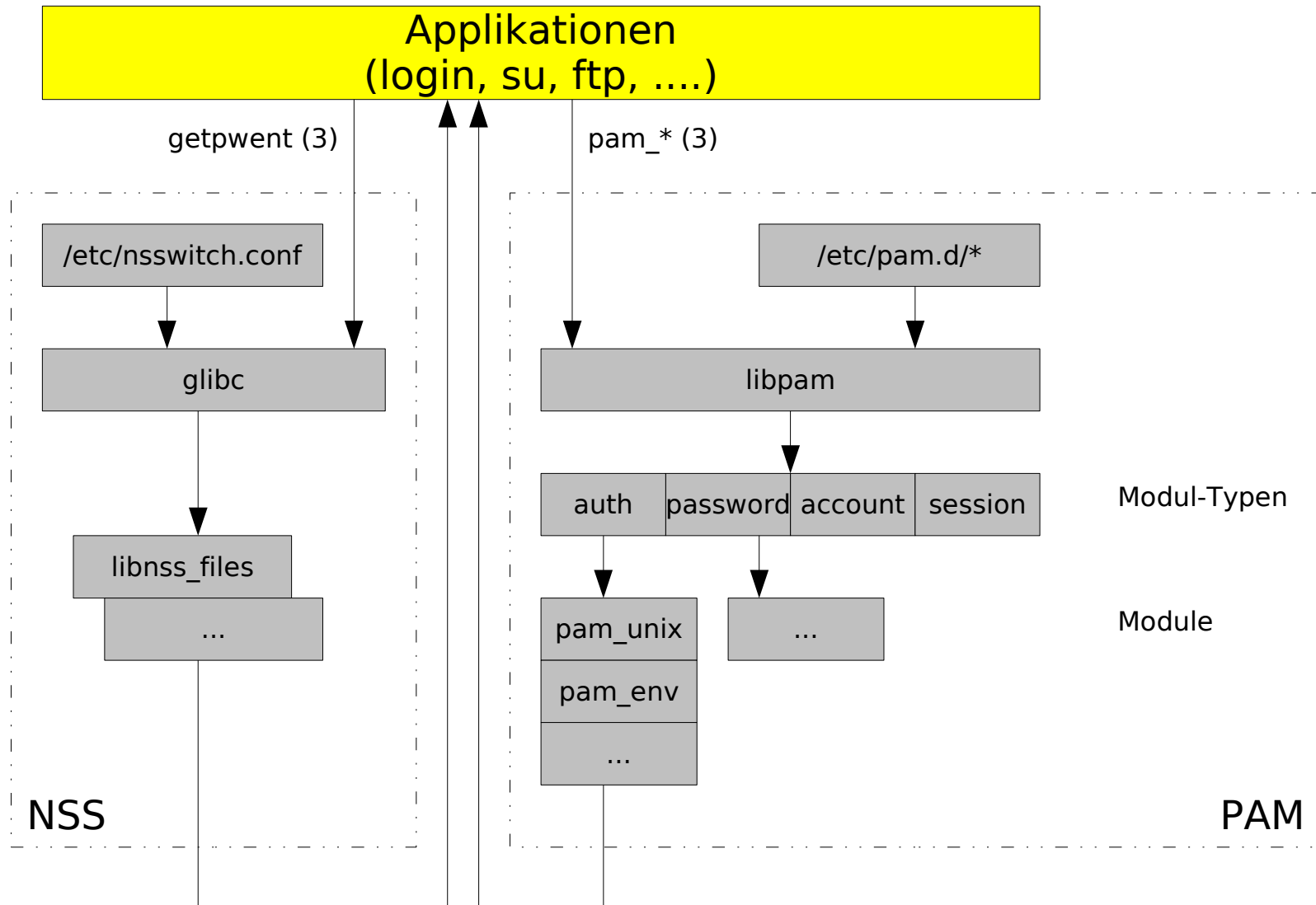
- Integration in die HRZ-Infrastruktur (AD):
 - Applikationen
 - viele Dienste, eine Authentifizierung --> "SSO"
 - Userbezogene Nutzung der Dienste:
 - smb-shares
 - individuelle Abrechnung der Druckerntuzung

AD

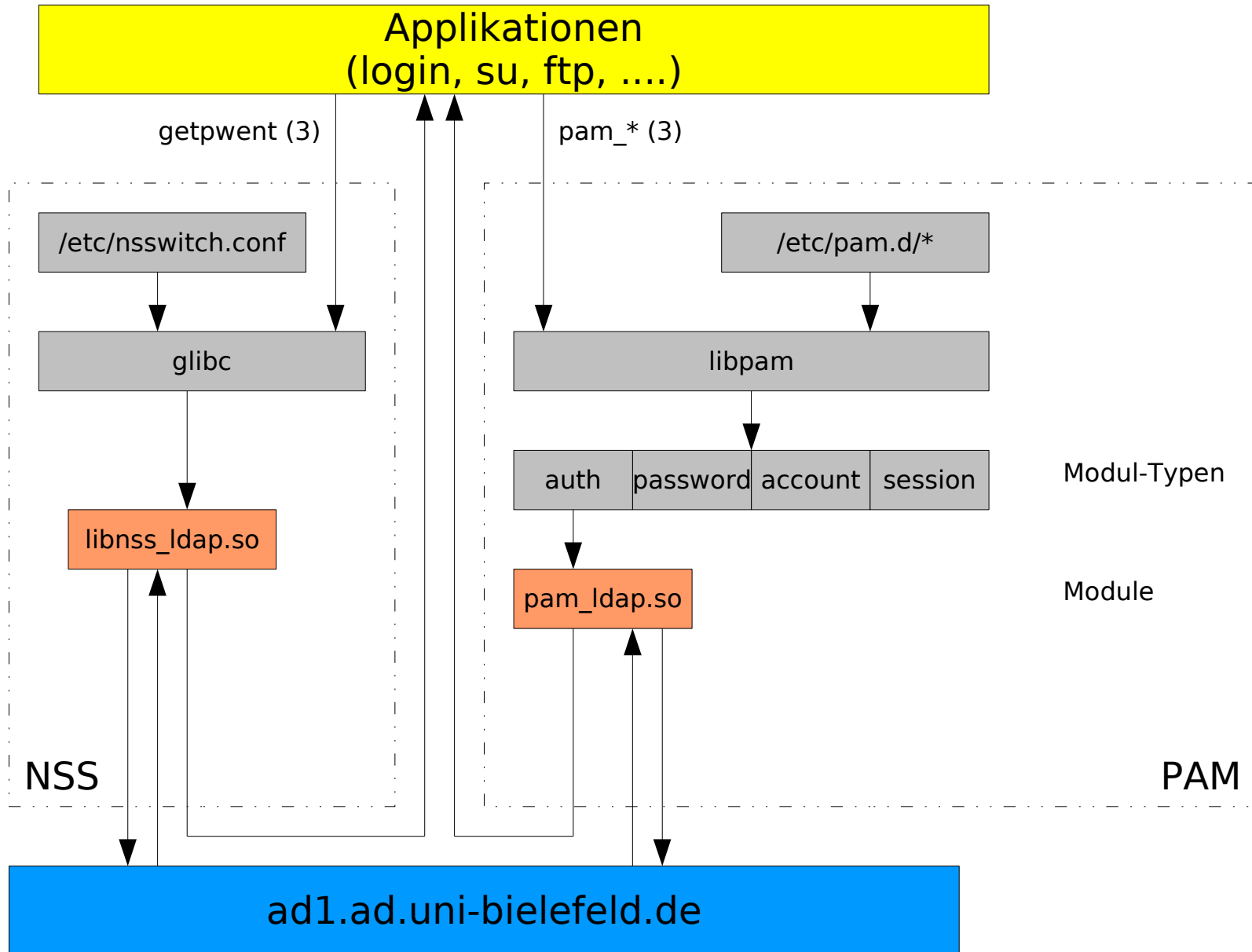
- hierarchischer Verzeichnisdienst fuer Nutzer, Ressourcen und Dienste
- AD = LDAP+Kerberos+CIFS +DNS
- AD-content= forest (trees (domains (OUs)))
- AD = Schemata+Konfiguration+Domains
- Multi-master Replikation zwischen rw-DCs durch trust-Beziehungen
- Identifikation von security principals: SID, RID, sAMAccountName

1. Authentifizierung

Linux PAM+NSS Architektur



(a) ldap



Idap (Forts.)

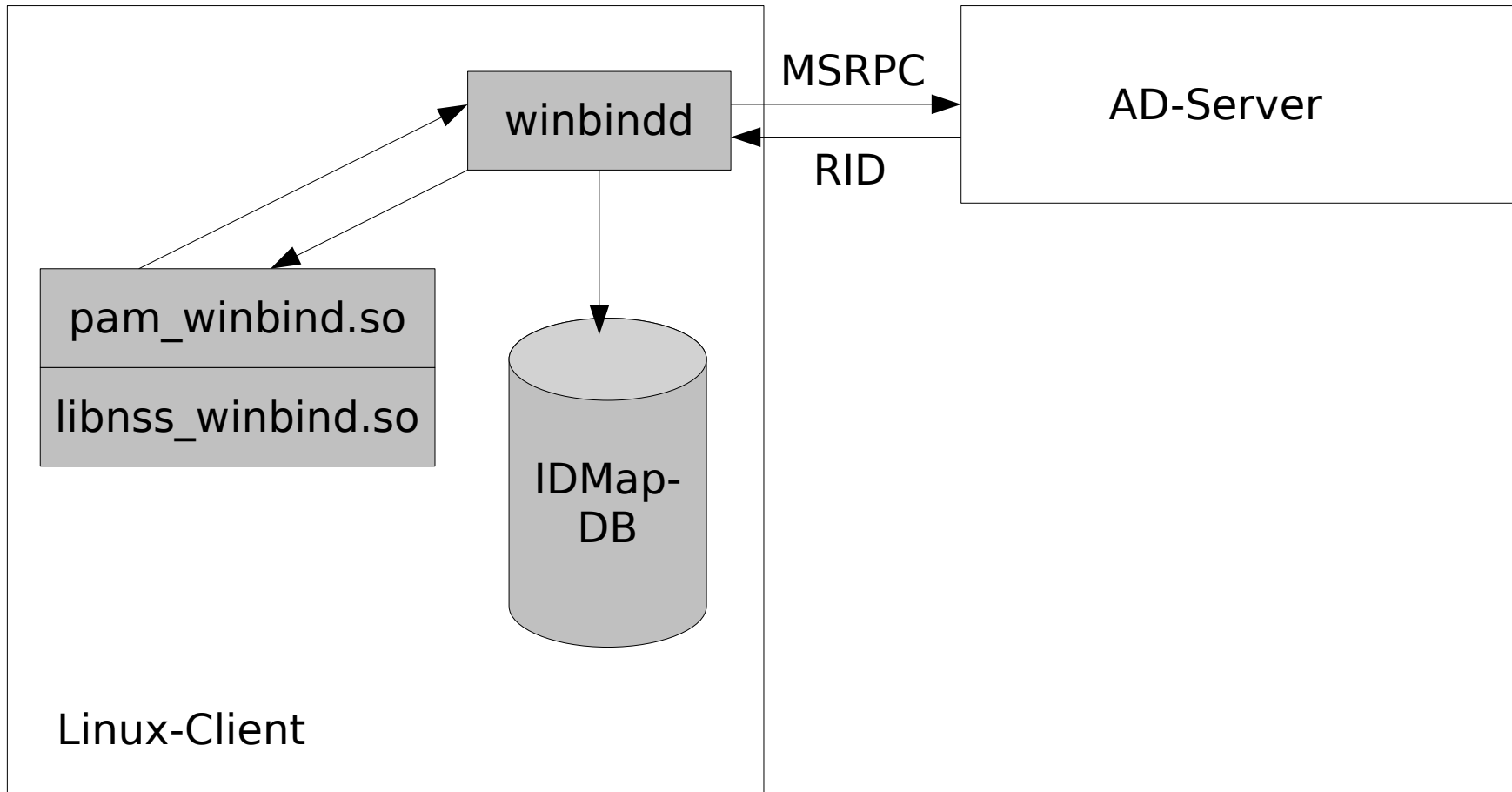
- direktes mapping von AD-Objekt-Attributen auf Idap-NIS-Attribute (`posixAccount`; RFC 2307)
- Voraussetzung: Schema-Erweiterung um die SFU
- potentielles Risiko --> Keine Option!

<code>nss_map_attribute</code>	<code>uid</code>	<code>sAMAccountName</code>
<code>nss_map_attribute</code>	<code>uidNumber</code>	<code>uidNumber</code>
<code>nss_map_attribute</code>	<code>gidNumber</code>	<code>gidNumber</code>
<code>nss_map_attribute</code>	<code>uniqueMember</code>	<code>member</code>
<code>nss_map_attribute</code>	<code>givenname</code>	<code>givenName</code>
<code>nss_map_attribute</code>	<code>ou</code>	<code>description</code>
<code>nss_map_attribute</code>	<code>gecos</code>	<code>displayName</code>
<code>nss_map_attribute</code>	<code>homeDirectory</code>	<code>unixHomeDirectory</code>
<code>nss_map_attribute</code>	<code>loginShell</code>	<code>loginShell</code>
<code>nss_map_attribute</code>	<code>shadowLastChange</code>	<code>pwdLastSet</code>

(b): winbind

- 3 Funktionen:
 1. Authentifizierung von credentials
(`pam_winbind.so`)
 2. ID-Mapping zwischen AD und Linux (NSS)
 3. DB für die Mappings (`winbind_idmap.tdb`;
erlaubt ldap-backend für einheitliche Mappings)
- Erlaubt Windows-auth für samba DCs

(b) winbind (Forts.)



smb.conf für winbind

```
[global]

workgroup = AD
server string = %h
security = ads
realm = ad.uni-bielefeld.de
password server = ad1.ad.uni-bielefeld.de
encrypt passwords = true

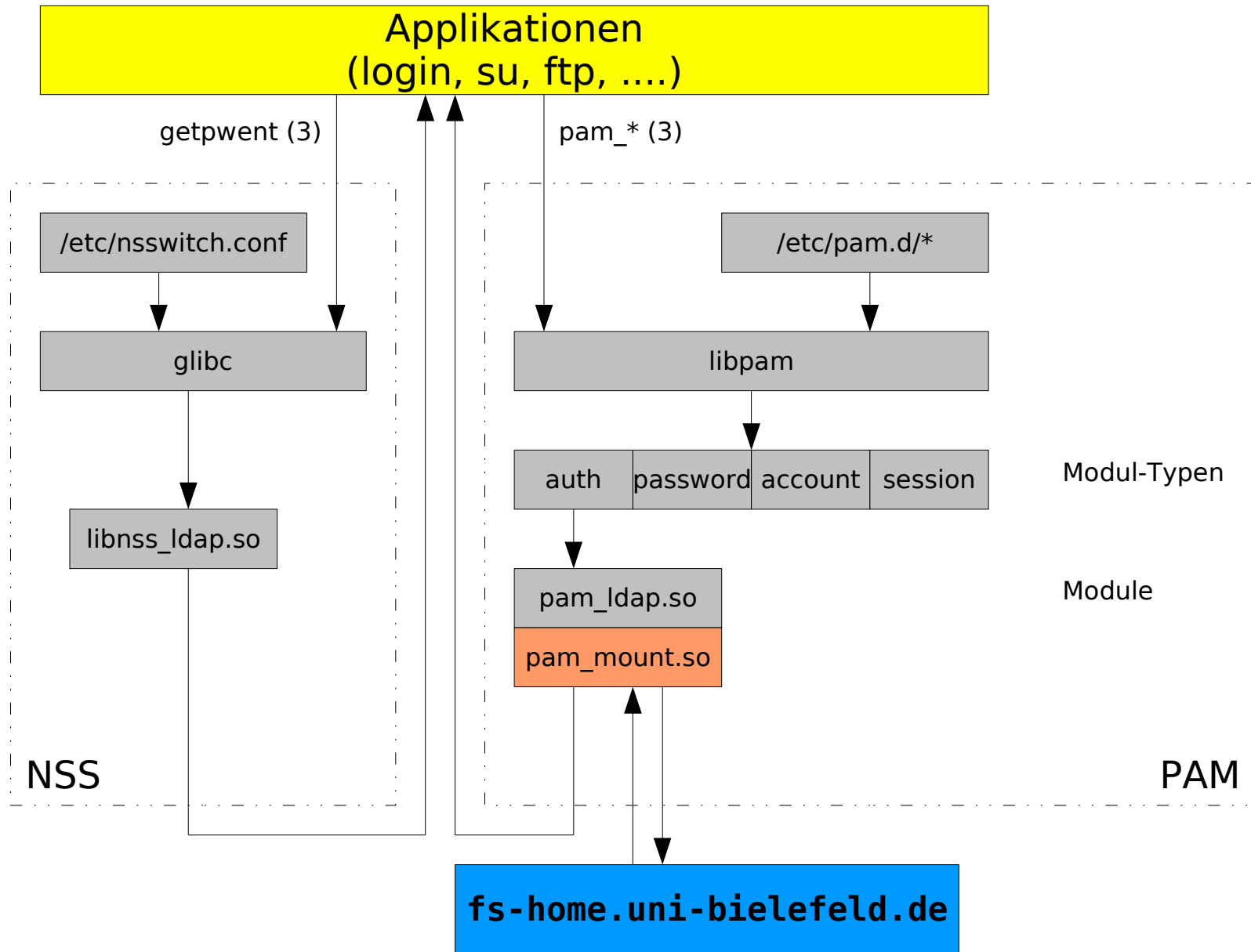
# ...

winbind separator = \
idmap uid = 10000-100000
idmap gid = 10000-100000
# allow enumeration of winbind users and groups
winbind enum users = yes
winbind enum groups = yes
# give winbind users a real shell (only needed if they have
telnet access)
template homedir = /home/%D/%U
template shell = /bin/bash
# ...
```

2. home-shares

- home-Verzeichnis bei login einbinden
- pam_mount mountet bei login von fs-home.unibielefeld.de
- Nachteile: braucht suid auf smbmount
- (pam_cifs)

pam_mount



homes: Probleme + Lösungen

(1) die home-Verzeichnisse (`\\fs-home.uni-bielefeld.de\\home\\hkaelberer`) sind “keine shares”

– bind-mount?

– “anonymer mount” des kompletten home-Baums

```
smbmount /usr/bin/smbmount //%(SERVER)/%(VOLUME) %(MNTPT) -o  
"username=%(USER),uid=%(USERUID),gid=%(USERGID)%(before=\", \"  
OPTIONS) "
```

```
volume * smbfs fs-home.uni-bielefeld.de home/ /home/AD/  
uid=&,dir_mode=0700,workgroup=AD - -
```

```
#volume * smbfs fs-home.uni-bielefeld.de home/& /home/AD/&  
uid=&,dir_mode=0700,workgroup=AD - -
```

homes-Probleme (Forts.)

(2) Einschränkungen der Dateitypen und
-Operationen im smb-mount → keine xfce-
/gnome-/kde-logins

- Lösung: iceauth-Informationen nach /tmp/
- kein KDE

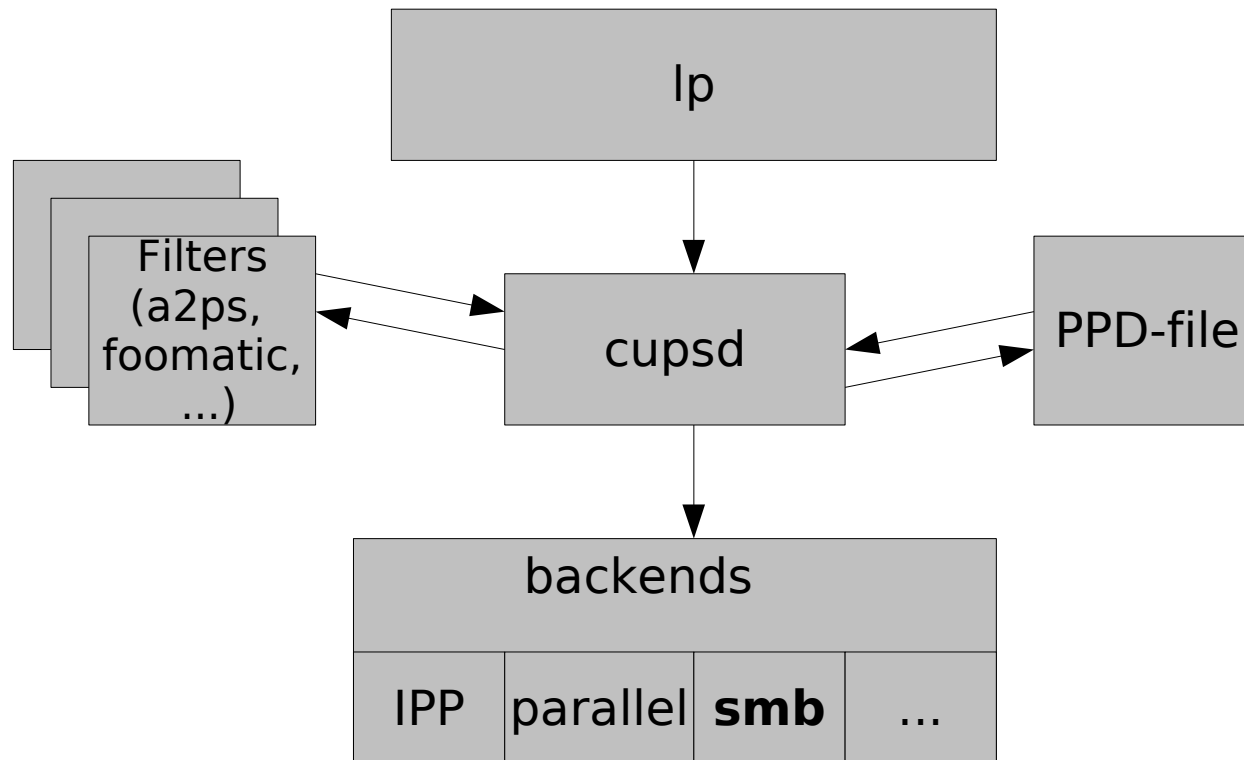
```
/--- /etc/gdm/Xsession  
ICEAUTHORITY="/tmp/.ICEauthority-${USER}"  
export ICEAUTHORITY  
---/
```

3. Drucken

- Anforderungen (HRZ):
 - SSO
 - individuelle Drucker-Nutzung (und Kosten-Abrechnung) !!!:
 1. aktuellen Nutzer kennen
 2. mit diesem Nutzer authentifizieren
- `hrz-pserv.ad.uni-bielefeld.de` **spricht nur smb**

a) CUPS

- Architektur:



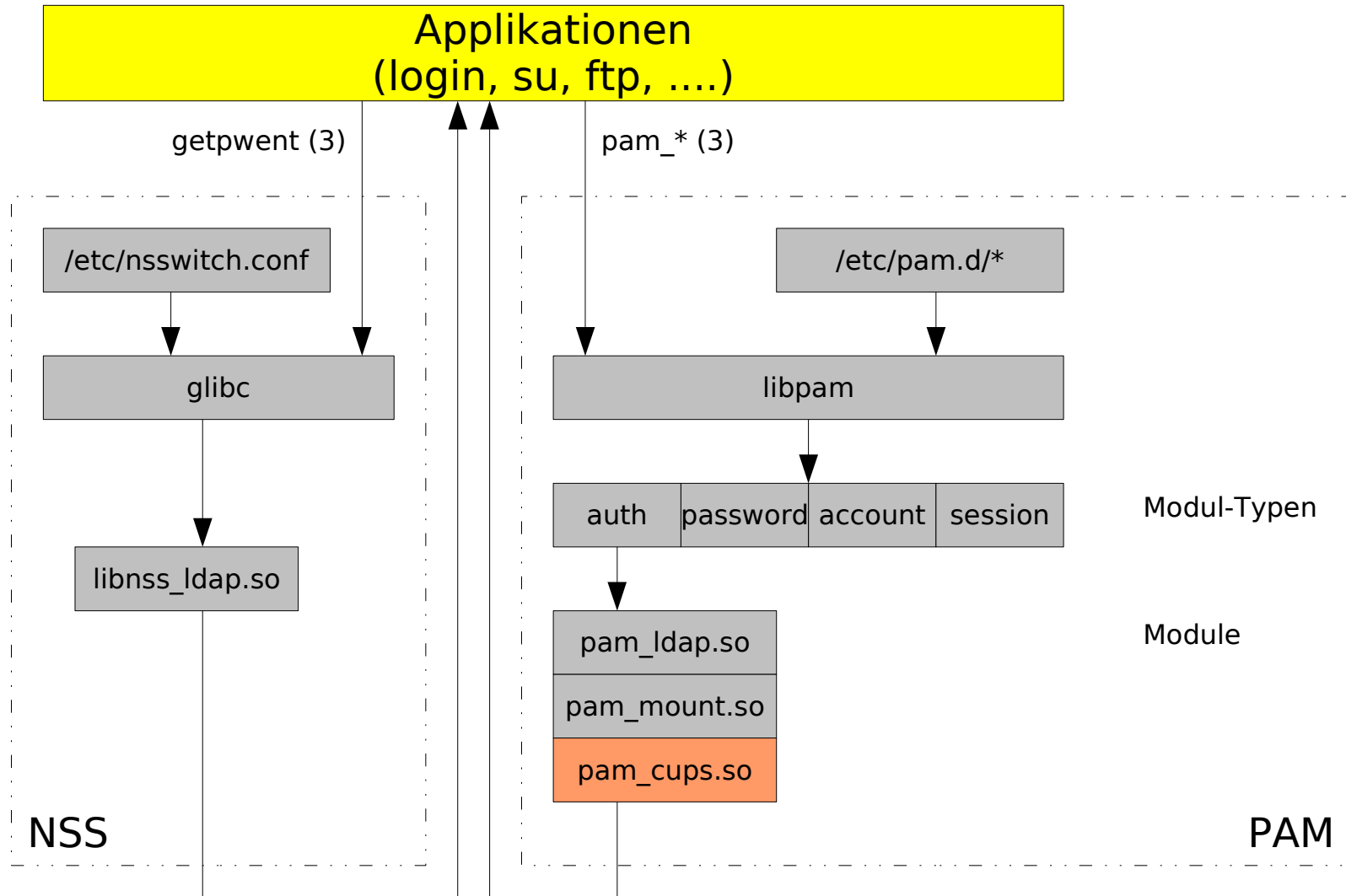
CUPS – smb-backend

- smbpool bekommt zwar Nutzer-Informationen von cups ... kann aber kein Kerberos und braucht ein Klartextpasswort

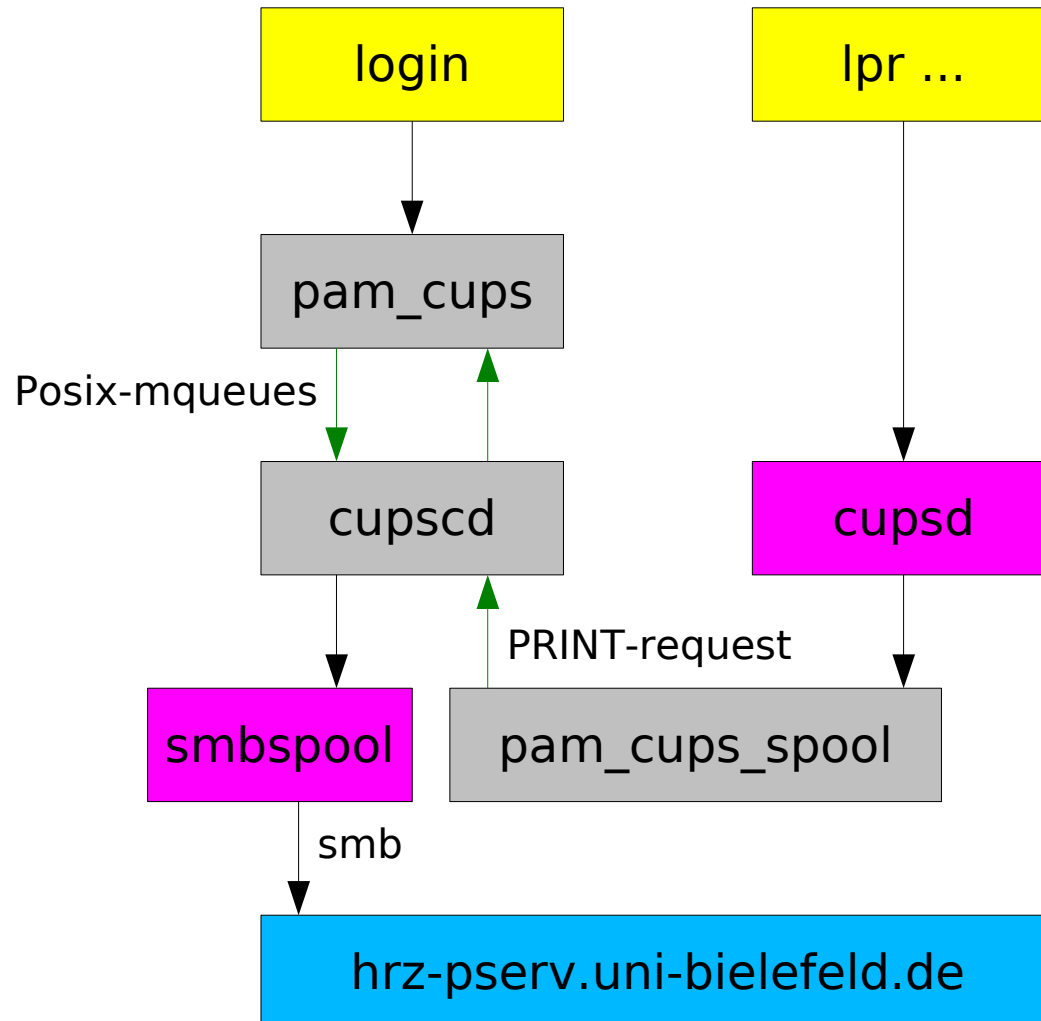
```
/--- /etc/cups/printers.conf
<Printer HRZ_HP_LASER>
[...]
DeviceURI smb://hkaelberer:SECRET@hrz-pserv.uni-bielefeld.de/HRZ_HP_LASER
[...]
---/
```

- Speichern der credentials im home ist keine Option

pam_cups (Forts.)



pam_cups



b) Alternativen

- smb-backend mit Kerberos-Unterstützung
 - evtl. als wrapper-script um smbclient; (nicht-triviale backend-Spezifikationen!)
- LPRng: smbprint als filter-script kann via smbclient credentials übergeben

```
cat FILE.ps | smbclient //hrz-pserv.uni-bielefeld.de/HRZ_HP_LASER  
-U hkaelberer -k -c "print -"
```

Zusammenfassung

- Integration eines Linux-Client in ein Win-AD ist auf Umwegen (samba-tools und pam-Module) möglich
- Nachteile (gegenüber Win-Clients):
 - nicht jeder WM
 - kein browsen der Drucker (müssen fest konfiguriert werden)
 - bisher ungelöst: kein smbmount bei logout
 - (weniger Applikationen)