OpenAFS aus Anwendersicht

Ein Vortrag an den Chemnitzer Linuxtagen

März 2005

von Mathias Feiler

Vortragssprache

- Vortrag in Muttersprache
- Landessprache: nur bedingt Fähigkeiten
- Übersetzung in Deutsch bei Bedarf möglich
- Phatnasievolle Schreibweise

AFS-Themen

- Preabel
- Historie
- Das Problem
- Volume
- Mounten
- @sys
- Authentisierung
- Autorisierung
  - Gruppen
  - ACL
- Klienten
  - Überblick
  - Callback
  - PAG
- Account
  - Einloggen
  - Dateizugriff
- Was tun, wenn...
- Afs Kommandos
- Dunkle Kapitel
- Appendix

Was ist AFS?

- AFS (Andrew File System)
  - Eine Infrastruktur
  - Ein Globales Filesystem
  - Ein eindeutiger Namensraum
  - Ein spezieller Directoryservice um Daten zu finden
  - Ein Zugriffs-beschränkungs-System
  - Arbeitet mit Kerberos zu Authentisierung

AFS Historie 1

- 1984 CMU: AFS teil des Andrew-Projekts
  - Designziele des AFS
    - Performance
      - Eigene Threads : „Light weight processes“
      - Eigene RPC: RX
    - Sicherheit : Kerberos des MIT
    - Skalierbarkeit
    - Ortstransparenz
  - Sponsor und Projektpartner: IBM

AFS Historie 2

- 1989 Transarc vermarktet AFS kommerziell
  - Schwaches Marketingkonzept: „Qualität reicht“
- 1990 OSF wählt AFS als FS-Komponente „DFS“ zum DCE
  - 1996 : OSF + X/Open = „The open group“
- Transarc entwickelt AFS4 als DFS weiter

AFS Historie 3

- 1992 DCE/DFS soll AFS ablösen
  - Zu wenig Leidensdruck
  - DCE/DFS ist zu komplex und fragil
- ~1997 IBM greift nach Transarcs „Resourcen“
  - Vorerst erfolglos , doch Begehrlichkeiten bleiben.

AFS Historie 4

- ~1999 IBM greift nochmals nach Transarc
  - Die begehrten „Resourcen“ entweichen in andere Firmen (z.B. spinnacker.com) .
- AFS Gesamteigentum der IBM
  - IBM hat Verträge (Pflichten) aber keine „Resourcen“ .
  - Offenbar springen indische Programmierer ein.
  - IBM will die Produkte / Verträge los werden.

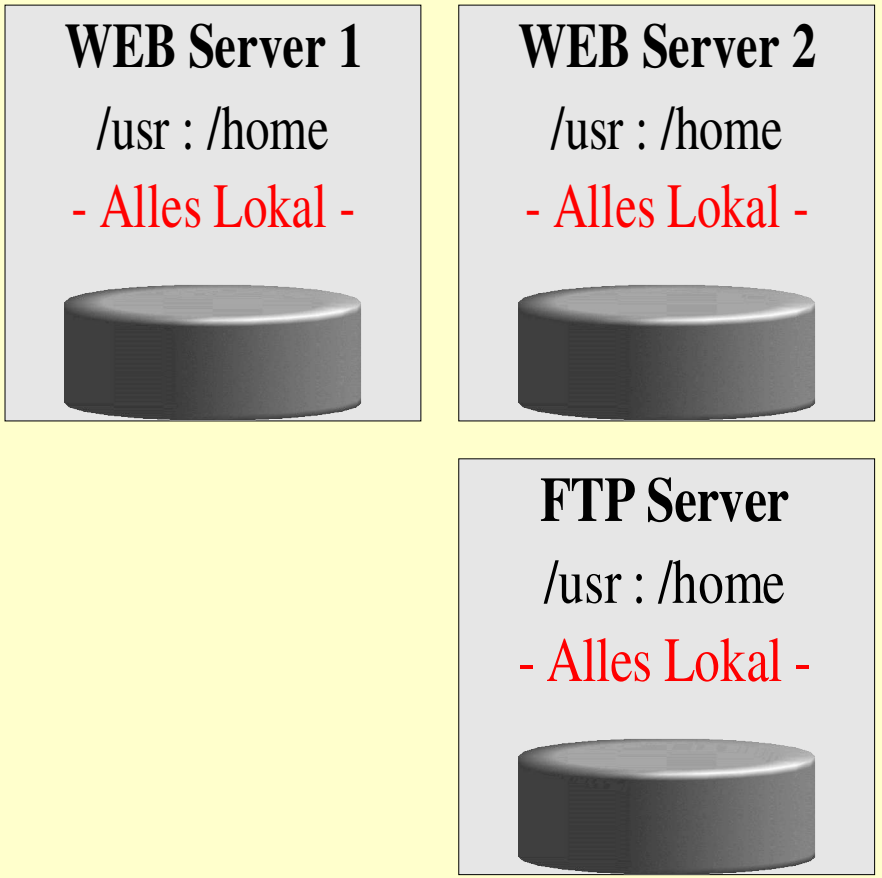
AFS Historie 5

- 31.10.2000 : IBM macht AFS3.6 zu Opensource
  - Reinfall wird in Trend-Erfolg umgemünzt.

- DCE/DFS :
  - Offenbar verzwickte Rechtslage für IBM
  - Offenlegung des IBM-Codes vorerst nicht erwartet
  - V1.2.2 (letzte Version der „The Open Group“) ist seit ca. 3 Wochen als Open Sourc verfügbar

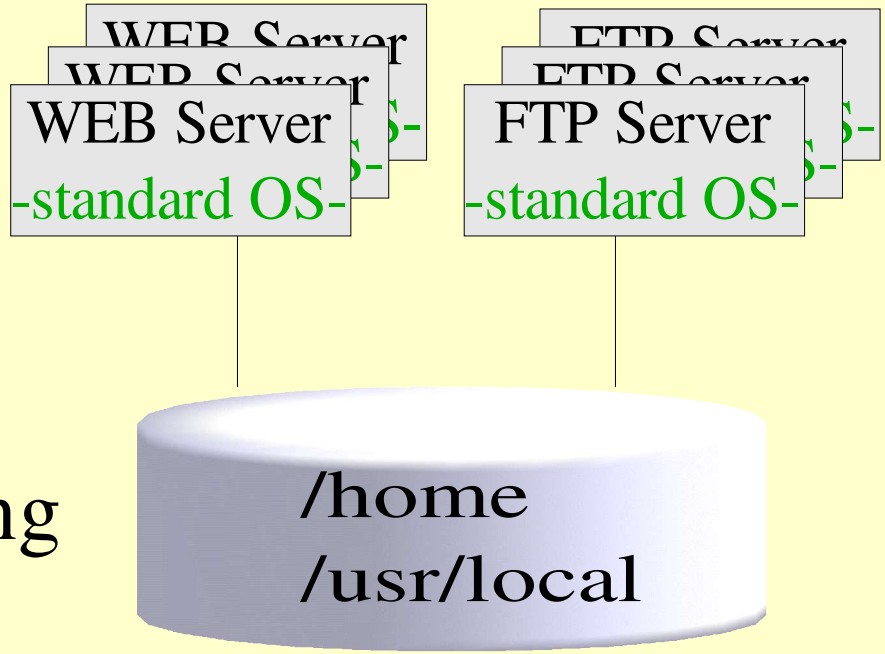
Das Problem

- Lokale Denke / Gewachsene Strukturen
  - Bei N Server:
    - N Datenlokationen
    - N Loginservices
    - N Passworte zu ändern
    - N Backups pro Tag
    - N Softwareinstallationen
  - Arbeit ~ Server



Die Lösung

- Zentrale Datenhaltung
  - Bei N Server
    - 1 Daten-Lokation
    - 1 Backup pro Tag
    - 1 Softwareinstallation
- Zentrale Authentisierung
    - 1 Passwort zu ändern
    - 1 Authentisierungsservice zu Pflegen
    - Ggf. Netzwerk-Gruppen



Wege zur Lösung

	Novel	NT4 XP?	Samba Cifs	NFS 3	DCE/DFS	AFS
Quota	+	-	O	O	+	+
Peak-tolleranz	O	-	-	-	+	+
ACL	++	+	O	O	++	+
Skalierbarkeit	-	-	-	-	+	+
Bekanntheit	+	+	O	+	-	-
Globalität	-	-	-	-	++	++
Global eindeutiger Namensraum	-	-	-	-	+	+

+ : gut O : neutral - : schlecht

Quelle: Projektstatusbericht bei der AFS-Tagung 2001 in Chemnitz sowie eigene Erweiterungen

AFS Gesamt-Struktur

- AFS
  - Globales Filesystem (global im Sinne des Globus)
  - Global eindeutiger Namensraum : /afs/<site>/...
  - Besteht aus einzelnen **AFS zellen**
  - Zellen finden : /usr/vice/etc/CellServDB oder DNS
  - Zellname meist wie Domäne, z.B.
    - uni-hohenheim.de
    - tu-chemnitz.de

AFS Zellen-Struktur

- AFS -Zelle
  - Fileserver-Maschinen
    - Volumeserver (VOLS)
    - Fileserver (FS)
  - Datenbankserver-Maschinen
    - Volume-location (VLS)
    - Kerberos (KAS)
    - Protection (PTS)
  - Klienten (Benutzermaschine)

Volume - Was ist das?

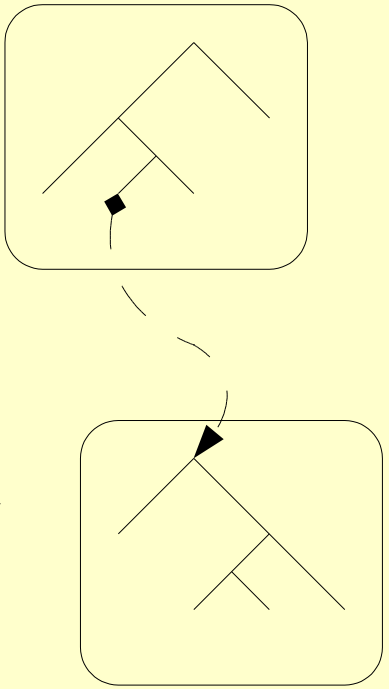
- Container für Dateien und Direktories des AFS
- Eine Art „Virtuelle Festplatte“
  - Mounten
  - Salvage (ak. fsck)
  - Endliche (bestimmbare) maximale Größe

Volume : Systemsicht

- Native Dateneinheit für
  - Clonen , Replikation , Backup , Verschieben , usw.
- Tausende Volumes auf einem phys. Laufwerk
- **Mehrere** Volume-**typen**
  - RW , Readonly(s) , Backup , RClone
- Pro User /Projekt etc. min. ein Volume sinnvoll
- Größenbegrenzung (quota)
- Max. Volumegröße < phys. Laufwerksgröße

Volume : Anwendersicht

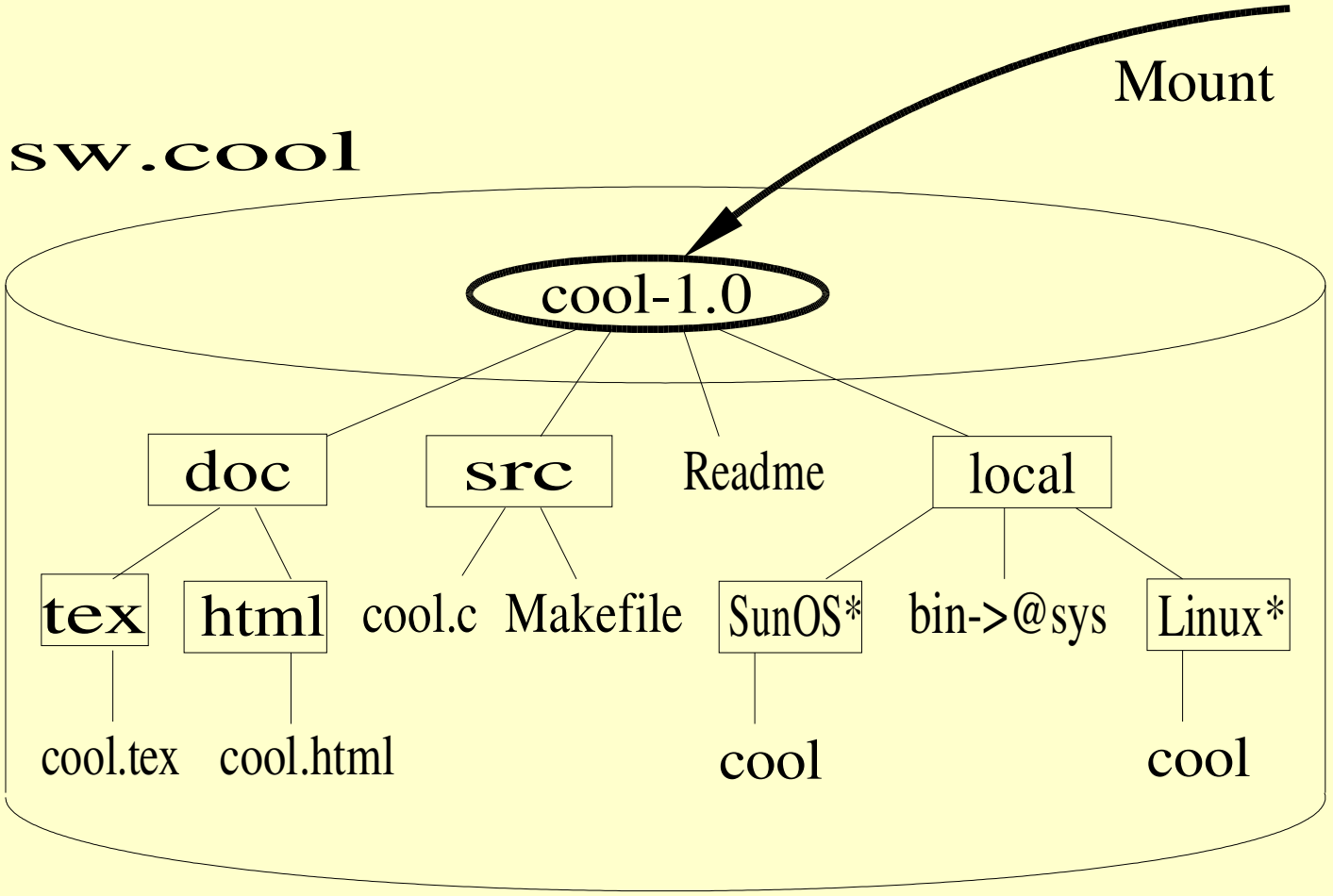
- Beinhaltet Teilbaum des AFS
- Logisch zusammenhängende Daten
- Zugriff über mountpoint
  - Client interpretiert Mountpoint (Symlink)
- Benutzer kann beliebig Volumes mounten
- Hält auch zugehörige Metadaten
  - Quota in KB , ACL , Zugriffsstatistik
- Eigner hat implizit ACL-Administrationsrecht



File aus Volume

- Volume
  - Volume-Lokation-Server (vlserver) verrät, wo es lebt
    - Daten aus der VLDB (Volume Location Data Basae)
- File aus Volume
  - Fileserver (FS) bedient mit Datei aus Volume
    - Falls die Rechte dazu ok sind
      (kommt später genauer)

Volume Darstellung



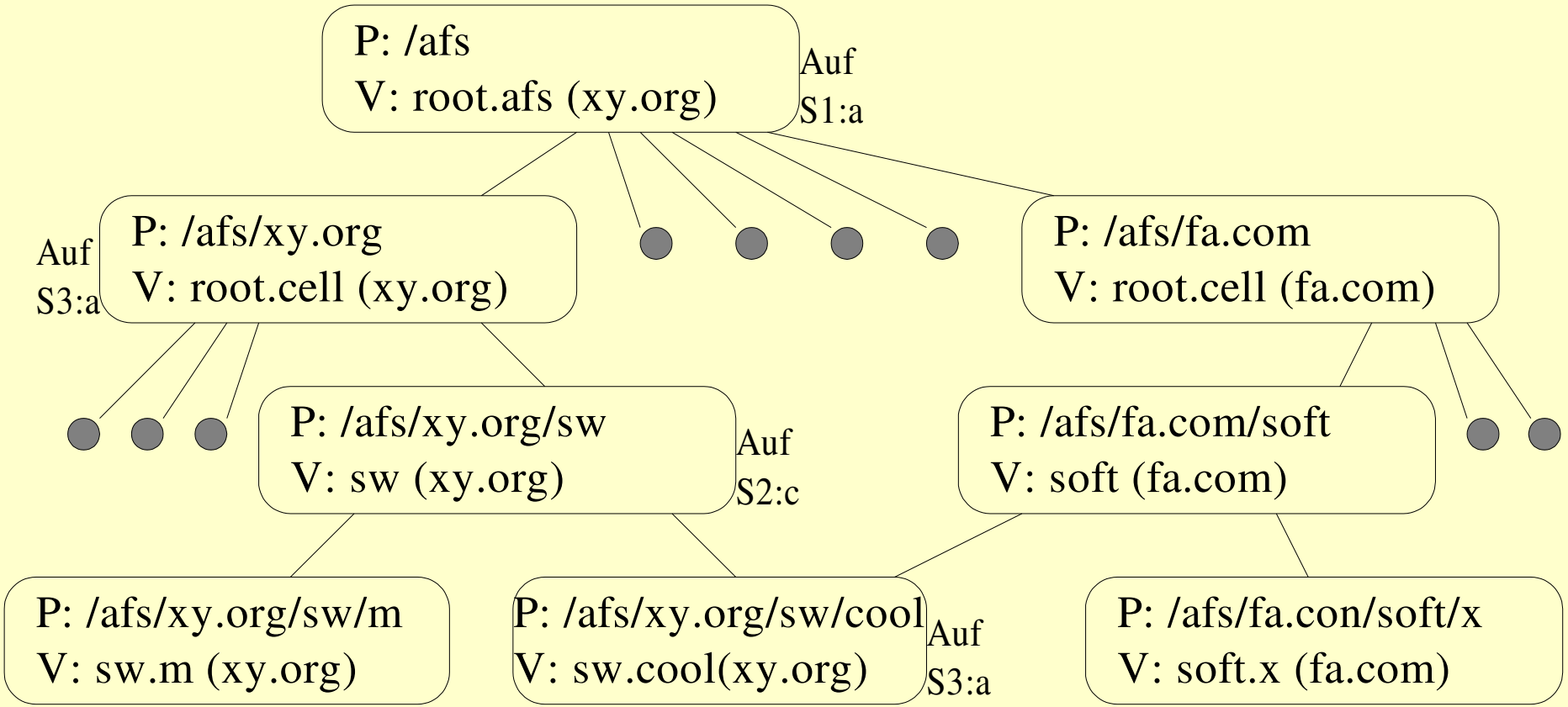
*) Reale 'Sysnames' sehen anders aus. Siehe 'fs sysname'

Exkurs : @sys

- @sys
  - Wird von der Klient-Maschine ausgewertet
  - Wird mit den Systemnamen des Klienten ersetzt
    - System-namen: siehe 'fs sysname' z.B.
      - i386_linux24
      - sun4x_59
  - Ermöglicht Klienten-abhängige Pfade
    - für Binary
    - für Configfiles (z.B. Keyboard oder .xinitrc)

Volume : Mount-Baum

- Volume-Mount-Baum Beispiel



Volume Mount (1)

- Volumemounts sind spezielle Symlinks
  - Standard Volume mount
    - #<Volumename><blank>
    - #<Zelle>:<Volumename><blank>
  - Read/Write Volume mount
    - %<Volumename><blank>
    - %<Zelle>:<Volumename><blank>
  - Beispiele (normal jedoch per 'fs makemount')
    - ln -s '#uni-hohenheim.de:root.cell ' uni-hohenheim.de

Volume Mount (2)

- Standard-mount
  - Keine Angabe zum Volume-typ
  - Veweist normalerweise auf den selben Volumetyp
  - Verweist implizit auf die eigene Zelle
- Expliziter mount von Volumetypen
  - RW : Option '-rw' beim mounten ('fs makemount')
  - RO : <Volume.Name>.readonly
  - Backup : <Volume.Name>.backup

Volume-Mount-Pfad

- Zwei Pfade (Bäume)
  - RO-Pfad (Readonly, default)
    - Einstieg : /afs/<site>/...
    - Führt mglst. durch RO-Volumes, dann durch RW-Volumes
  - RW-Pfad (Read/Write)
    - Einstieg : /afs/.<site>/...
    - Führt durch RW-Volumes
- Kein Backup-Pfad
    - Standardlink im Backupvolume zielt auf RO-Volume

Mountpfad-wechsel

- Mountpfade können gewechselt werden
  - In den RW-Pfad
    - Wenn kein RO-Volume da ist
    - Wenn ein expliziter RW-mount vorliegt
  - In den RO-Pfad
    - Wenn ein expliziter RO-mount vorliegt
  - In ein Backup-Volume
    - Nur durch expliziten mount dieses Backup-Volumes

Volumemount : Tücke (1)

- Mountpfad kann 'ausversehen' wechseln
  - z.B. Wenn kein RO-Volume da ist
- Timeout, wenn Volume nicht erreichbar ist
  - 'ls -l' oder 'ls --color' (linux-default) in /afs
    - Irgend eine Zelle ist sicher gerade nicht erreichbar...
    - Abhilfe :
      - „\ls“ an stelle von „ls“ benutzen
      - Alias umdefinieren

Volumemount : Tücke (2)

- Unerwartet viele Daten verfügbar für Klienten
  - z.B.: Explorer in Verzeichnis '/afs/<zelle>/home'
    - 'stat'-funktion auf 10.000 User-Volumes

      (Ist es ein Directory? Ja? Gelbes Hängeordnersymbol mappen...)
    - Stupiede Sortierung verketterer Liste
      - Ca. (10K/2)^2 = 25.000.000 Vergleiche von Objekten
    - 'read' auf 10.000 Directories zur Subdirectory-Suche (wegen des (+) links neben dem gelben Hängeordner)
  - .... Stunden später....
  - Ähnliches gilt für 'ls -l' oder 'ls –color'

Filesystem verstanden

- Ohne Benutzer wäre hier nun alles OK
- ABER .....

  .....Es gibt die Benutzer

  - Benutzer brauchen
    - Eine geschützte Privatsphäre
    - Die definierbare Öffnung für Teile der Privatsphäre

...Privatsphäre

- Privatsphäre braucht
  - Authentisierung
    - Ist er der , der er vorgibt zu sein?
  - Autorisierung
    - Darf er das tun, was er tun will?
  - Daten-Objekte auf die sich die Autorisierung bezieht
    - Directories
    - Files
    - Rechte

...Öffnung

- Definierbare Öffnung braucht
  - Daten-Objekte
    - z.B. Directory
  - Attribute
    - Art der Öffnung (z.B. Auflisten)
  - Ausgewiesene Interessenten (Wem gilt die Öffnung)
    - Einzelne (z.B gonzo, gringo, hugo,...)
    - Klassen bzw. **Gruppen** (z.B. Entwickler, Lohnbüro,...)

Symmetrie

Privatsphäre und definierbare Öffnung ....

... Sind die beiden Seiten der selben Medaille

Der Account ist das Substrat

Ist er der , der er sagt , daß er es sei?

- Authentisierung
  - Afs arbeitet mit Kerberos (ursprünglich vom MIT)
    - Der Klientmaschine wird mgl. misstraut, daher gibt es keine 'preshared key' zur Benutzerauthentisierung...
      ... aber ein 'preshared secret'
    - Braucht keine 'sichere' Leitung
    - Symmetrische Verschlüsselung
    - Vergibt Session-key und Ticket bzw. Token fur 'AFS'
      - Zeitlich begrenzte Gültigkeit
    - Kerberos kennt Prinzipale (Namen), keine Nummern

AFS-Authentisierung (1)

- Standard AFS-Installation : KA-Server
  - Basiert auf Kerberos4
  - Ist Uhrzeit-abhängig (wie alle Kerber**???**)
  - AFS spricht gerne via RX mit dem KA-Server
    - RX ist das AFS-RPC-Protokoll (über UDP)
  - KA-Server kann auch normales Kerberos4
    - Wird derzeit noch von Windows-Klienten benutzt
  - Hat (über RX) einige Erweiterungen

AFS-Authentisierung (2)

- PRO KA-Server
  - + Sehr gut integriert via RX
    (Läuft reibungslos so wie es mit OpenAFS kommt)
  - + Pre-authentication via RX
    (ermöglicht Limit auf Fehlversuche)
  - + Ubik: dynamisches Syncsite -> Hochverfügbar
  - + Ticket-Laufzeiten >> Kerb4(21¼ Stunden)
  - + Lange Zeit nichts mehr in den Security-listen ?!?

AFS-Authentisierung (3)

- CONTRA KA-Server
  - Schlechte/keine Integration mit anderen Produkten
  - Leider doch nur Kerberos4
    - verschiedene Design-Fehler und Schwächen
  - Kürzerer Schlüssel (DES =Data Encryption Standard)
  - Schwaches PCBC (Plain and Chipher Block Chain)

  = Wer NT (oder XP) naturbelassen benutzt kann auch nicht viel gegen Kerberos4 haben.

AFS-Authentisierung (4)

- AFS und Heimdal (Kerberos 5)
  - + Deutlich sicherer
  - + Standardisiert(er) und kann obendrein RX
  - + Gut gepflegt (schwedische Nächte sind lang ...)
  - + Kann weitere Attribute verwalten und liefern
  - + Systeme kommen i.A. mit Kerberos5
  - - Starre Syncsite (Master/Slave Modell)
  - - Etwas mehr Aufwand

Darf er tun, was er tun will?

- Autorisierung
  - AFS beinhaltet ein „ProtectionService“ (PTS)
    - PTS kennt Namen und zugehörige Nummern
      - Benutzername : ID > 0 , mgl. gleich Unix-ID
      - Gruppennamen : ID < 0
      - IP-Adressen : Name beginnt numerisch , ID > 0
    - PTS kennt 5 Rechte auf PTS-Objekte

s,S	Eintrag zeigen	-,a,A	Neue Mitglieder einfügen
-,O	Liste Gruppen in Besitz	-,r	Mitglieder entfernen
-,m,M	Liste Mitglieder/Mitgliedschaft		

Autorisierung / AFS-Gruppen (1)

- AFS- oder PTS-Gruppen sind Netzwerkgruppen !!!
  - Klientrechner kann Gruppen nicht 'frei interpretieren'
- Gruppe ist (leider nur) Menge von
  - 'Benutzer' (mit gültigen Token)
  - IP-Adressen
- PTS merkt sich (u.a)
  - Den Eigener einer Gruppe
  - Den Gruppennamen : (<Eigner>:<Gruppenname>)
  - Die Mitglieder einer Gruppe

Autorisierung / AFS-Gruppen (2)

- Gruppen können sich selbst gehören
  - Können sich dann selbst verwalten
  - Wird gerne für Projekte eingesetzt
  - Wird vom Admin so eingerichtet
- Benutzer kann selbst Gruppen anlegen
  - Name : <Benutzer>:Gruppenname
  - Name ist maximal 63 Zeichen lang
  - Typisch: maximal 20 Gruppen pro Benutzer
- Administratoren können Namen frei vergeben

Autorisierung /AFS System Gruppen

- system:anyuser
  - User ohne token , z.B: root, User anderer Zellen, etc.
- system:authuser
  - User mit gültigem Token für diese Zelle
- system:administrators
  - Mitglieder dieser Gruppe dürfen zusätzlich:
    - Alle ACL der Zelle ändern.
    - Quota für alle Volumes der Zelle bestimmen.

Verknüpfung

- Wir haben (A) Volumes und Files
- Wir haben (B) Kerberos und PTS mit Gruppen

Nun brauchen wir eine Verbindung ....

....Wir brauchen **ACL**'s

ACL – Was ist das?

- ACL – Access Control List (Zugriffsrechteliste)
  - Attributierte Verknüpfung von
    - Verbraucher (Benutzer)
    - (begehrtem) Objekt
- ACL im AFS
  - Vergabe von (leider nur Filesystem) Zugriffsrechten
    - Für : User , Gruppen , IP-adressen (vom PTS)
    - Auf : Verzeichnisse (und Dateien) (des FS)

AFS - ACL (1)

- Positive und Negative ACL
  - Positive ACL (Freigabe)
    - z.B.: Mitglieder der Gruppe 'Entwickler' dürfen schreiben
  - Negative ACL (Beschränkung)
    - z.B.: Aber 'dokudog' darf bestehende Dateien nicht ändern
- ACL vererben sich auf neue Unterverzeichnisse
  - Leider keine initial-ACL
- Maximal 20 ACL-Einträge pro Verzeichnis

AFS - ACL (2)

- 7 Rechte : r,l,i,d,w,k,a
  - 3 Dateirechte
    - 'r' „read“ - Recht, Inhalte ansehen zu dürfen
    - 'w' „Write“ - Recht, eine Datei zu überschreiben
    - 'k' „Lock“ - Das Recht, (per sys-call) zu locken
  - 4 Verzeichnisrechte
    - 'l' „lookup“ - Recht, Verzeichniseinträge listen zu können
    - 'i' „Insert“ - Recht, Dateien anlegen zu dürfen
    - 'd' „delete“ - Recht, Verzeichniseinträge zu löschen
    - 'a' „Admin“ - Recht, ACL zu verändern
  - 8 Undefinierte Rechte : A,B,C,D,E,F,G,H

AFS - ACL (3)

- Namen für Rechte-Set (zur Vergabe)
  - 'all' == 'rlidwka'
  - 'write' == 'rlidwk'
  - 'read' == 'rl'
  - 'none' == ''
- Rechte für einen 'Briefkasten'
  - 'li' dirList , Insert

AFS-ACL und Modebits

- AFS-ACL werden pro Verzeichnis vergeben
- Verzeichmisse
  - ACL gilt exklusiv
  - Unix Modebits gelten nicht
- Dateien
  - ACL des Verzeichnisses gilt
  - Die Owner-Modebits (Linke drei) gelten allgemein
    - z.B.: Skript hat 'x'-bit

AFS-ACL Beispiel

- Bis zu 20 ACL-Einträge pro Verzeichnis
- Eine ACL hat 7 mögliche belegte Rechte
- Eine ACL bezieht sich auf je ein PTS-Objekt.

Beispiel: `fs listacl ./directory`

```
Access list for ./directory is
Normal rights:
   system:administrators rlidwka
   softwaregrp rlidwk
Negative rights:
   gonzo dwka
```

SetUID aus AFS

- SetUID aus AFS ist Sicherheitskritisch
  - Akzeptanz (ob , aus welchen Zellen)
    - 'root' entscheidet pro Klientrechner
    - Default: SetUID aus der eigenen Zelle zugelassen
      - Abschalten mit 'fs setcell <cell> -nosuid'
    - Zugelassene Fremdzellen: /usr/vice/etc/SuidCells
      - Sollte mgl. leer sein !
  - S-Bit Setzen
    - Nur Mitglieder von 'system:administrators'

Server ...

- Filesserver .....
- Volume-Lokation-Server .....
- Kerberos .....

brauchen

Klienten

Klientmaschine allgemein

- Kontaktet (ggf. anderen) Server
- Publiziert AFS gegen Programme und User
- Interpretiert Volume-Mountpoints
- Verwaltet AFS-Tokens für die Benutzer
- Betreibt einen Cache
- Invalidiert ggf. den Cache (callback)
- Schreibt Daten nach 'fclose' auf den FS zurück

Klientmaschine technisch

- Kernelmodul
  - Hält credentials (z.B. Tokens) pro **PAG**
  - Lenkt vfs-calls ggf. auf das AFS um
- Daemons ('afsd' , mindestens 5 instanzen)
  - Callback: und 'Server-Pings' verarbeitung
  - Maintenance: Kernel memory , RO-Vol's stündlich prüfen
  - Cache-truncation: Cache zurückschreiben, freigeben
  - Server-connection: RW-Files-Check und Uhrzeit
  - Background: prefetch / delayd-Write

Was ist ein 'Callback'?

- Callback :
  Das Versprechen des Fileservers,
  gewisse Ereignisse dem Klienten zu melden
  - RO-Volume verändert – cache ungültig
  - RW-Datei hat sich geändert – cache ungültig
- 'Callback brechen'
  - Das ereigniss ist eingetreten

Was ist 'PAG' ?

- PAG – Process Authetication Group
  - Mehr als ein Prozess in einer PAG
  - Ggf. mehr als eine PAG pro Benutzer
- PAG definiert die logische Grenze eines Tokens
  - Neue PAG mit 'pagsh'
  - UID ist Token-Grenze, wenn keine PAG existiert
- U*.*X - Hack
  - PAG-ID wird in zwei Gruppennumern aufbewahrt

PAG (2)

- PAG oder UID als Token-Grenze (Token-ID)
  - 'groups' oder 'id' zeigt ob eine PAG vorliegt
  - UID ist default, wenn keine PAG existiert
  - UID ist schnell mal gewechselt, PAG wandert mit
    - Setuid-Programme
    - 'su' -Kommando
  - Logout-skript mit 'unlog'
    - ohne PAG können andere Sitzungen betroffen werden

Account-Overtüre

- Im Orchster
  - Bass KAS
  - Cello PTS
  - 1. Geige Volumelokation-Server
  - 2.Geige Fileserver
  - Dirigent Cachemanager
- Spielen ein Stück aus 'User Account'

AFS-Account : Einloggen

- Authentisierung
  - Aus Account 'A' und Passwort wird Key 'K' generiert
  - Gegen den geprüften Key K erhält vom Kerberos
    - ein Token 'T' des Prinzipals 'A' für den Service 'AFS'
    - Ein Session-Key 'S' in K verschlüsselt

      Anmerkung: T= (A, AFS ,S ,Zeit ,Lebensdauer,..)
- Authorisierung / Service
  - Gegen T und in S verschlüsselte (Zeit,A) erhält man AFS-services (Dateien, Gruppen usw.)

AFS-Account : Dateizugriff (1)

- Account A will Datei X sehen
  - Programm (z.B. less) öffnet Datei X (fopen)
  - Kernelmodul erkennt 'AFS-Welt'
    - Leitet call in den Cachemanager (CM) um.
  - Volumename ist vom Pfad (Mountpoint) bekannt
  - CM : Kein passendes Cache-Objekt.... schade.
  - Passenden Zell-Datenbank-Server auswählen
    - CellServDB, DNS, Server-Rank

AFS-Account : Dateizugriff (2)

- Datenbankserver (VLDB) anfragen. wo V existiert
  - VLDB antwortert mit Liste der mgl. Fileserver (FS)
- CM wählt Fileserver (FS) und passendes Token T aus
  - VLDB-antwort , Server-Rank ,
  - PAG, Account , Kerneltabellen
- Anfrage an FS bestehend aus :
  - (Zeit,A)S (auch „Authenticator“ genannt)
  - Token T (T= {A, AFS ,S ,Zeit ,Lebensdauer,..})
  - „Bitte um X aus V für A“

AFS-Account : Dateizugriff (3)

- Fileserver FS prüft Anfrage:
  - Kann das Token T entpackt werden?
  - Kann (Zeit,A) mit S aus T entschlüsselt werden?
  - Ist das Paket frisch genug (max. 5 Miuten)?
  - Gilt das Token noch?
  - Ist Anfrager A mit Prinzipal aus Token identisch?
  - Ist das Prinzipal A laut ACL berechtigt X zu bekommen?
    - Ist A direkt in der ACL aufgeführt?
    - Ggf. Ist A in einer Gruppe, die in der ACL aufgeführt ist?
    - Ist A in einem negativen ACL-Eintrag geführt?
    - Ist A in einer Gruppe mit negativen ACL?

AFS-Account : Dateizugriff (4)

- FS liefert (falls OK) Datei X aus
  - Typisch in 64K-Häppchen
  - In S verschlüsselt, falls vom CM gefordert
- Cachemanager nimmt X-Häppchen entgegen
  - Ggf. entschlüsseln der X-Häppchen mit S (des Accounts)
  - Ablage des/der X-Häppchen im Cache
  - Auslieferung via Kernelmodul an User-prozess
- Proramm (z.B. less) zeigt X an

AFS-Account : Dateizugriff (5)

- User A will Datei X nocheinmal sehen
  - Programm (z.B. less) öffnet Datei X (fopen)
  - Kernelmodul erkennt 'AFS-Welt'
    - Leitet call in den Cachemanager (CM) um.
  - CM: Heureka , passendes Cache-objekt ist da !!!
    - Prüfen, ob Aufrufer das Cacheobjekt so nutzen darf
    - Ggf. Auslieferung von X via Kernelmodul an Aufrufer
  - Proramm (z.B. less) zeigt X an

AFS-Account : Datei schreiben (1)

- Programm (z.B. syslog) öffnet datei 'X' (fopen)
  - .... cm ... synonym dem Lesen ... ...
  - Programm schreibt Δ1 in X
    - Cachemanager cachet Δ1
  - Stunden später ....
  - Programm schreibt Δ2 in X
    - Cachemanager cacht Δ2
  - ......

AFS-Account : Datei schreiben (2)

- Programm schließt X (fclose)
  - Kernelmodul signalisiert fclose(X) an CM
  - CM prüft ggf. auf welchem FS Volume V existiert .....
  - CM wendet sich an FS
    - (Zeit,A)S auch „Authenticator“ genannt
    - Token T
    - „Bitte um Speichern dieser Blöcke in X aus V für A“
  - FS prüft ..... (und speichert ggf.)
    - Synonym dem Lesen

Was tun wenn...

....dann....

Was tun wenn... (save1)

- ... Eine Datei nicht gespeichert werden kann
  - Sofortmaßnahme : vorläufig auf /tmp sichern
  - Token abgelaufen?
    - 'tokens' , ggf.: 'klog <prinzipal>'
  - hinreichende Zugriffsrechte?
    - „fs listacl“ , ggf ACL einrichten (lassen)
  - Quota ausgereizt?
    - „fs listquota“ , ggf. Mülltrennung! Admins **nicht** anrufen!

Was tun wenn... (save2)

- Server erreichbar ?
  - „fs checkservers“ ,
  - Ggf. 'traceroute <FS>'
  - Ggf. (Netzwerk-)Admins anrufen!
- Volume gelockt?
  - 'fs exa <dir> ' ---> VolName
  - 'vos exa <VolName>'
  - Ggf. Warten ... Warten ...
  - Ggf. Admins anrufen!

Was tun wenn... (Token)

- ... das Token/Ticket ausläuft
  - Ein neues Token von KAServer besorgen:
    - 'klog'
  - Ein neues Ticket vom Kerberos besorgen
    - 'kinit' um ggf. ein TGT zu erhalten
    - 'aklog' um ein neues Ticket in den Kernel zu stopfen

Was tun wenn... (klog)

- .... Einloggen oder 'klog' nicht funktioniert
  - Als Afs-Nutzer : 'root' anrufen
  - Als 'root' :
    - Uhrzeit richtig stellen
      - 'ntpdate <timeserver> '
    - bei welcher Zelle versucht man zu authentifizieren?
      - 'fs wscell' ,
      - ggf.: Lokaler temporaerer account und
        'klog <user> -cell <heimatzelle>'
    - PAM korrekt konfigurieren

Was tun wenn... (DB)

- ... Ein Datenbankserver ausgefallen ist
  - Sofortmaßnahme :
    - Keine – man braucht keine .
    - Bei vernünftigem Setup übernimmt ein anderer Datenbankserver den Dienst
  - Falls man doch was tun will
    - Serverrank für den ausgefallenen Server hochsetzen, um diesen möglichst nicht anzusprechen.

Was tun wenn... (CM1)

- .... der Klient eine 'seltsame' Sicht der Dinge hat
  - Als AFS-Anwender
    - CM soll Volumeinformationen überprüfen
      - 'fs checkvolumes'
    - CM soll Daten im cache invalidieren
      - 'fs flush [-path <dir>]' = alles zu diesem Directory
      - 'fs flushmount [-p <dir>]' = alles zu diesem Mountpoint
      - 'fs flushvolume [-p <dir>]' = alles zu dem entsp. Volume

Was tun wenn... (CM2)

- .... der Klient eine 'seltsame' Sicht der Dinge hat
  - Als 'root' auf dem Klienten
    - Den gesamten Cache verwerfen
      - 'fs setcachesize 1' benutze im weiteren nur einen Cache-Block
      - 'fs setcachesize 0' benutze konfiguriert viele Cache-Blöcke
    - Manchmal muss man brutal werden ... (Vorsicht!)
      - Falls cache eine eigene Partition ist :
        - umount /afs ; mke2fs -m 0 -L<label> /dev/<dev> ; reboot'
      - Sonst
        - umount /afs ; cd /usr/vice/cache/ ; rm *[el]Items ; reboot'

Was tun wenn... (ACL)

- .... Die ACL geändert wurden,
  der Zugriff aber trotzdem nicht funktioniert?
  - Der Benefiziar besorge sich ein neues Token
    - Danach werden die Rechte vom CM neu geprüft.
  - Hat der Benefiziant auch den Weg bis zum Zielort freigegeben?
    - ggf. Benefiziar zulassen ..... neues Token siehe oben

Was tun wenn... (gelöscht)

- ....Datei 'WICHTIG' gelöscht wurde?
  - Die Datei aus einem anderen Volume zurückholen
    - In welchem Volume lag die Datei 'Wichtig'
      - 'fs exa <dir>' ---> VolName
    - Gibt es eine Volume-Instanz von vor dem Löschen
      - 'vos exa <VolName>.readonly' , 'vos exa <VolName>.backup'
    - Ggf. das '**.**backup' (oder '**.**readonly') Volume mounten
      - 'fs mkm .backup <VolName>.backup
    - Datei zurückkopieren
      - 'cp -p .backup/<path>/WICHTIG <dir>

Was tun wenn... (überschrieben)

- ....Datei 'WICHTIG' überschrieben wurde?
  - Siehe 'Was tun wenn...(gelöscht)
- ....Datei 'WICHTIG' kaputt ist
  - Siehe 'Was tun wenn...(gelöscht)

Wie kann man machen daß...

Theorie ohne Praxis ist wie...

Salz ohne Suppe

AFS-Kommandos (1)

- Haupt-Kommando-Gruppen „command suits“
  - Leider nicht immer ganz sauber getrennt / zugeordnet
  - FS (Filesystem relevantes)
  - KAS (Rund um den KAServer , Authetisierung)
  - PTS (Gruppen- und Benutzer- Management)
  - VOS (Volume relevantes)
  - BOS (AFS-Server Betreuung und Beobachtung)

AFS-Kommandos (2)

- Weitere Kommandos
  - 'pagsh' um eine neue PAG anzulegen
  - 'klog' oder 'klog.krb' , um Token zu besorgen
  - 'kapasswd' , um das Passwort zu ändern
  - 'tokens' , um die Tokens aufzulisten
  - 'up' , rekursives kopieren von ACL und Dateien
  - 'unlog' , um Tokens zu verwerfen
  - Diverse andere, siehe Software-Paket.
    z.B. mit : `rpm -qil 'openafs-1.*'|less`

AFS-Kommandos (3)

- Kommandos geben i.A. gerne Hilfe:
  - Unterkommandos finden: '<suit> help'
  - Unter-Hilfe: '<suit> help <SubCmd>'
  - Benutzungshilfe: '<suit> <SubCmd> ... -h'
- Beispiel : pts help liste

pts listentries: list users/groups in the protection database
Usage: pts listentries [-users] [-groups] [-cell <cell name>] [-noauth] [-force] [-help]
Where: -users list user entries
-groups list group entries
-noauth run unauthenticated
-force Continue oper despite reasonable errors

AFS-Kommandos (4)

- Kürzel sind möglich
  - bis an die Grenze der Mehrdeutigkeit
  - Lang:
    - 'fs listacl'
    - 'fs listquota'
  - Kurz:
    - 'fs lista' als Spezialkürzel : 'fs la'
    - 'fs listq' als Spezialkürzel : 'fs lq'

AFS-Kommandos (5)

- Oft kann man Optionen weglassen
  - Wenn die Parameter-Reihenfolge eingehalten wird
  - Beispiel :
    - 'fs setacl -dir Subdir -acl rlidwk'
    - 'fs sa Subdir write'

Filesystem-Kommandos

- Wieviel Speicher ist für dieses Volume verfügbar
  - 'fs listquota [<Pfad>]'
- In welchem Volume befindet sich ein Verzeichnis?
  - 'fs examine [<Verzeichnis>]'
- Welcher Server liefert dieses File/Directory?
  - 'fs whereis [<path>]'
- Welchen Typs ist der Klient ? (--> @sys)
  - 'fs sysname'

Copy-Kommandos

- ACL Kopieren
  - 'fs copyacl <qelldir> <zieldir>'
- Dateien und ACL rekursiv kopieren
  - 'up <quellbaum> <zielbaum>'

Volume-Kommandos

- Wie heißt das Volume zum Directory
  - 'fs exa <dir> '
- Was weiß man über das Volume
  - 'vos exa <dir> [-extended]'
    - ....
    - Angelegt am
    - Zuletzt geändert am
    - Lock?
    - Bussy?
    - Nicht verfügbar? (z.B. wegen Server- oder Netz-Ausfall)

Volume-Mount-Kommandos

- Ein Volume mounten
  - 'fs mkm <dir> <Volume.Name> [-cell <Zelle>] [-rw]
- Ist dies ein Mountpoint, wenn ja, wohin?
  - 'fs lsm <dir> '
- Einen Mountpoint löschen
  - 'fs rmm <dir>

ACL-Kommandos

- Wie lautet die ACL zum Verzeichnis
  - 'fs la [<dir>]'
- Wer ist in dieser Gruppe
  - 'pts mem <gruppe>'
- ACL verändern
  - 'fs sa <dir> <PtsId> <Rechte>'
    - z.B.: 'fs sa ./src gonzo:freunde read'
- ACL Kopieren
  - ' fs copyacl <qelldir> <zieldir>'

Gruppen-Kommandos

- (M)eine Gruppe Anlegen
  - 'pts creategroup <(m)einePtsId>:<Gruppenname>'
- Jemanden in (m)eine Gruppe aufnehmen
  - 'pts adduser <PtsId> <(m)einePtsId>:<Gruppenname>'
- Gruppen-mitgliedschaft feststellen
  - 'pts member <(m)einePtsId>:<Gruppenname>'
  - 'pts member <PtsId>'
- Jemanden aus (m)einer Gruppe entfernen
  - 'pts removeuser <user> <(m)einePtsId>:<Gruppenname>'

Alles immer propper ?

..... auch im AFS gibt es die

Schmuddel-Ecken

Dunkle Kapitel (Datei)

- Keine Posix-semantik sonder Open/Close
  - Kein Byterange-locking
  - Open-Cloes : Der letzte Schreiber gewinnt
- Derzeit noch 2GB-Grenze für Files
  - Entwickler sind dabei, dies zu ändern.
  - Integration für Admins vermutlich etwas aufwendig
  - In OpenAFS 1.4 (voraussichtlich) enthalten

Dunkle Kapitel (ACL)

- ACL sehr unvollständig implementiert
  - Kein ACL auf : PT-Objekte,Volumes, Files, ACL, ...
  - ACL nur auf Directorys
  - Keine Initial-ACL
- Neue ACL wirken nicht immer sofort
  - z.T. Muss erst ein neues Token besorgt werden

Dunkle Kapitel (Volume)

- Volumes
  - Es gibt keinen .backup-Mountpfad
  - Nur eine Readonly-Generation (Snapshot)
  - RO's können nicht kopiert werden

Dunkle Kapitel (KAS/PTS)

- AFS kommt aus der Kerberos4 - Welt
  - Alles AFS-Services benutzen den selben ServerKEY
  - Keine deligierten Server möglich (ohne Schweinkram)
  - Nur eine Zelle pro KAS (ohne Programmieraufwand)
  - Keine Server/Service-Prinzipale (ohne tuning)
- Gruppen
  - Gruppen können keine Gruppen beinhalten

Dunkle Kapitel (VOS)

- Eine „denail of service attack“ ist möglich
  - 'Vos listvol <Server>' beschäftigt den Server heftig
    - Pro Volume:
      - FS: Volume Sync und lock aufgeben
      - VOLS: Volume lock, Daten auslesen, unlock
      - FS: Volume erneut locken
  - Leider keine ACL auf vos/vols bzw. Volumes

Dunkle Kapitel (CM)

- Cachemanager ist Trickprogrammierung
  - PAG in U*.*X -Gruppen : eher Hack als Konzept
  - Mit Linux 2.6 wird es noch etwas schlimmer
    - Fehlende Syscall-Tabelle / nicht modifizierbar
    - Kein 'credential'-Konzept im Linux-Kernel
      - Hier und da kleine bunte komische Workarounds
  - Neuerdings Bestrebungen zum generischen CM
    - Für diverse Filesysteme , z.B.: NFS4 , AFS, ...

Fragen und Diskusion

- ...

Appendix(1)

ACL	- Access Control List
AFS	- Andrew File System
CM	- Cachemanager (Klientmaschine)
CMU	- Carnegie Mellon University
DCE	- Distributed Computing Environment
DFS	- Distributed File System
FS	- Fileserver
IBM	- International Bussines Machines
KAS	- Kerberos4 Server des AFS , spricht RX
LWP	- Light weight processes = AFS-Threds
MIT	- Massachusetts Institut of Technologie
OSF	- Open Software Foundation
PAG	- Process authentication group – gültigkeitsgrenze für Token

Appendix (2)

Prinzipal	- Accountname im Kerberos
PTS	- Protection-service / Protectionserver
RO	- Readonly
RW	- Read/Write
RX	- RPC des AFS
Ticket	- Verschlüsselte Datenstruktur , Server kennen Schlüssel
Token	- Ticket für 'AFS' das im Kernel gehalten wird
UID	- Unix-ID – Nummer eines Unix-users
VLDB	- Volume Location Database
Volume	- AFS-Volum = Virtuelle Festplatte
VOS	- Volumeserver z.B. um Volumes zu verschieben
X/Open	- X/Open Company Ltd.

Quellen

- Gut: http://www.openafs.org/doc/index.htm
- Besser: Kommentare zum Softwarepaket
- Noch besser: Hilfen der Kommandos
- Am aller besten: SOURCE CODE

- Ausserdem: Andere Anwender und Admins