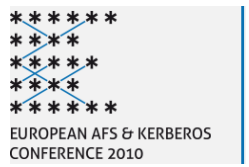
Setting-up your AFS cell and first steps

christof.hanke@rzg.mpg.de
wgehrke@dia.uniroma3.it

Christof Hanke,Wolfgang Gehrke

RZG,DIA

Rechenzentrum Garching MPG & IPP
DIA, Università degli Studi Roma Tre



Outline

1. The global picture
2. Helper services
3. Authentication
4. User information
5. Hands on!
   - Basics
   - Setting up a cell
   - First Steps
   - ACLs
   - Troubleshooting
   - Advanced topics
     - Client Cache
     - Network
     - Backup
     - cross-cell
6. Selling the infrastructure
7. Questions

EUROPEAN AFS & KERBEROS CONFERENCE 2010

What is AFS?

general

Andrew file system (Andrew project at CMU)
distributed file system with client-server architecture
product by Transarc Corporation (now IBM Pittsburgh Labs)
IBM branched the source of the AFS product as OpenAFS
to mention: DCE/DFS available under LGPL since 2005

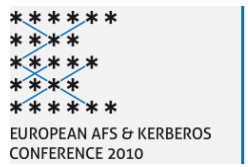
What is AFS?

general

Andrew file system (Andrew project at CMU)
distributed file system with client-server architecture
product by Transarc Corporation (now IBM Pittsburgh Labs)
IBM branched the source of the AFS product as OpenAFS
to mention: DCE/DFS available under LGPL since 2005

pros

union of storage of servers
homogeneous name space
location-transparent file space
client side caching
security and scalability
RO replication
clients for UNIX, Windows

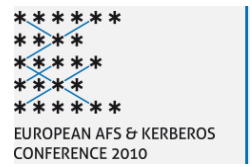


What is AFS?

general

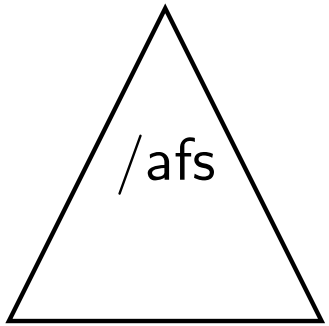
Andrew file system (Andrew project at CMU)
distributed file system with client-server architecture
product by Transarc Corporation (now IBM Pittsburgh Labs)
IBM branched the source of the AFS product as OpenAFS
to mention: DCE/DFS available under LGPL since 2005

pros	**cons** (some work in progress)
union of storage of servers	requires Kerberos
homogeneous name space	own UID, own GID, own ACL
location-transparent file space	not full POSIX, no special devices
client side caching	no byte-range locks, not all hard links
security and scalability	still DES and fcrypt
RO replication	no RW replication
clients for UNIX, Windows	kernel module, SMB/CIFS

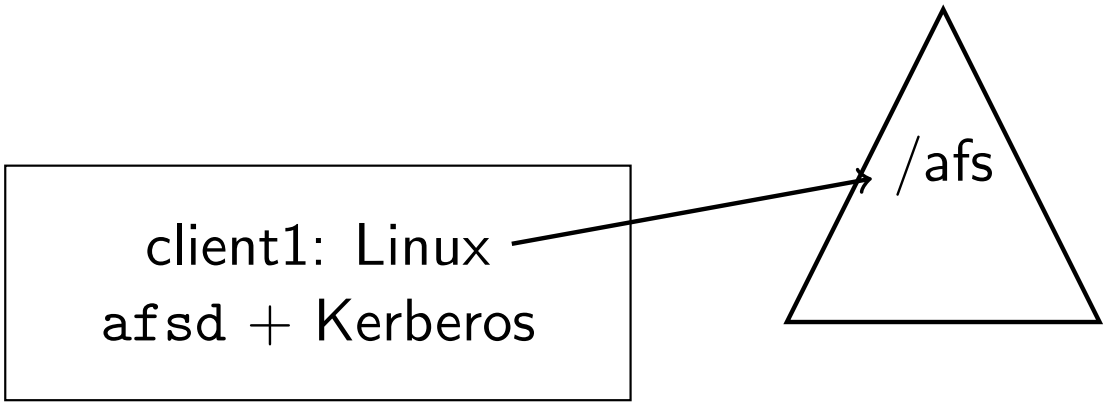


Sketch

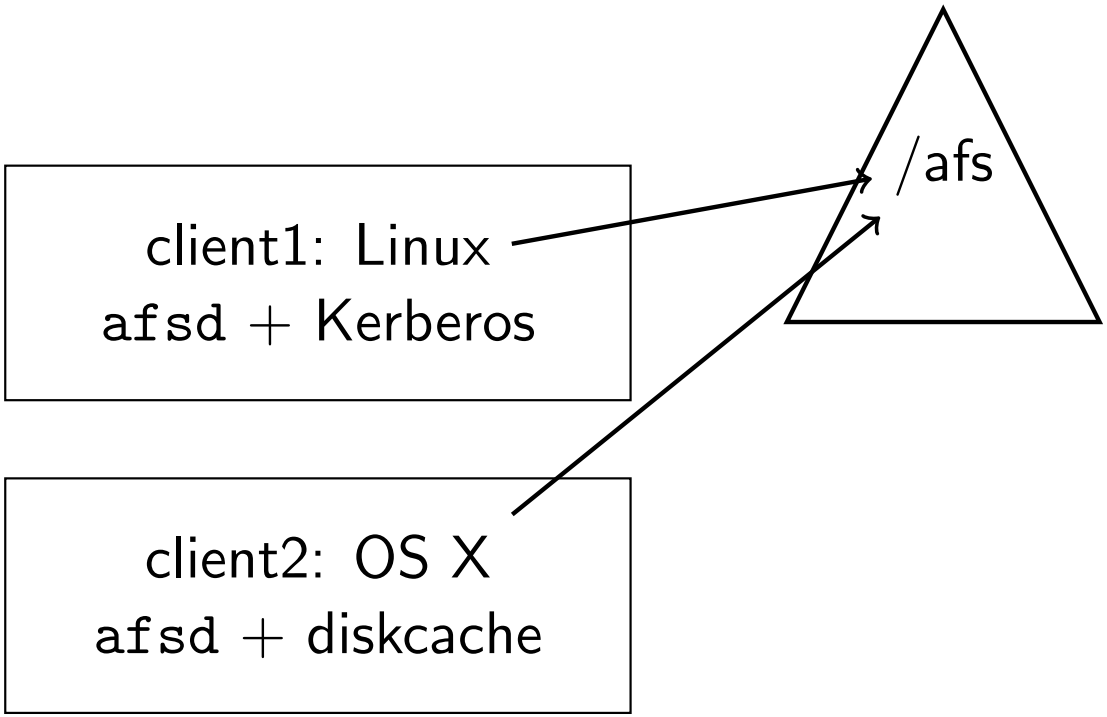
Sketch



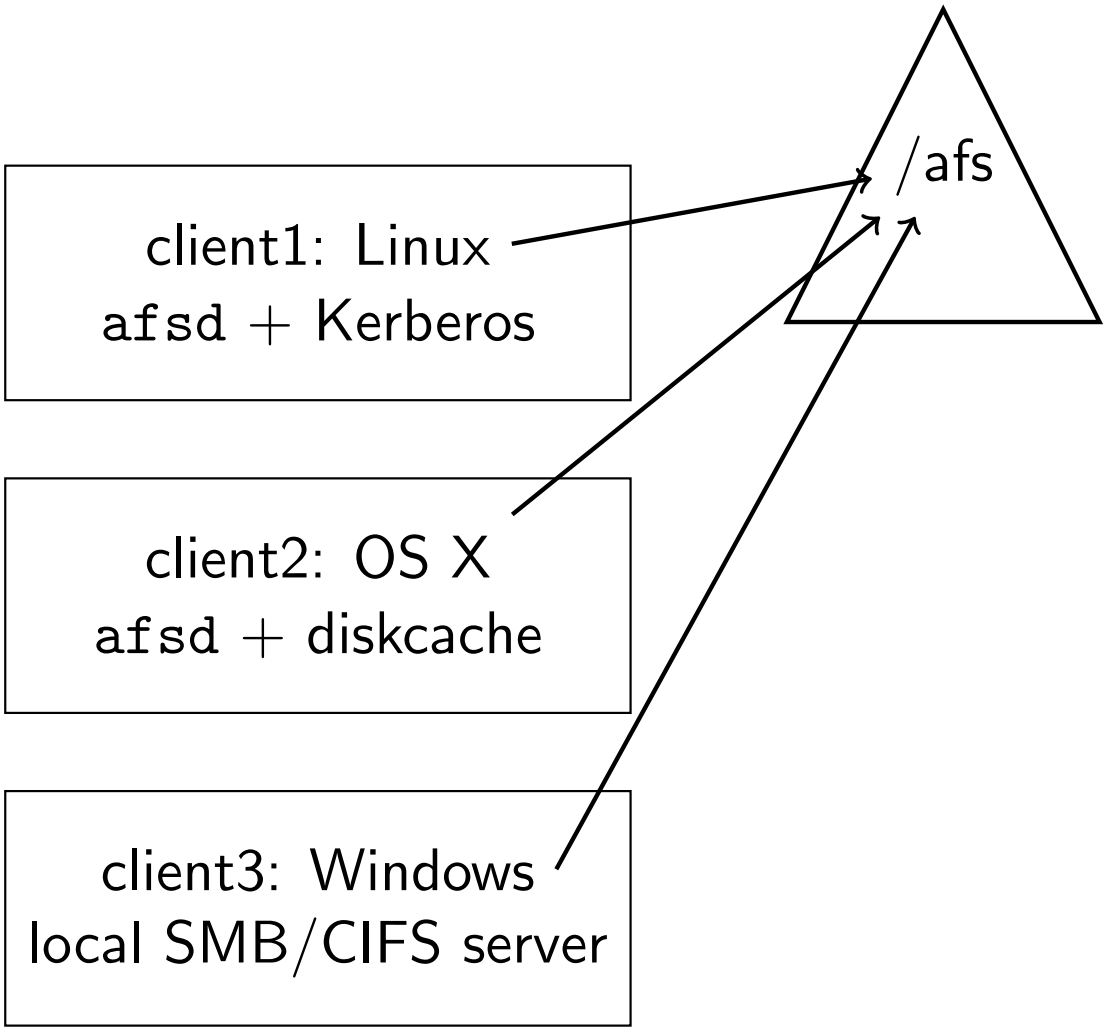
Sketch



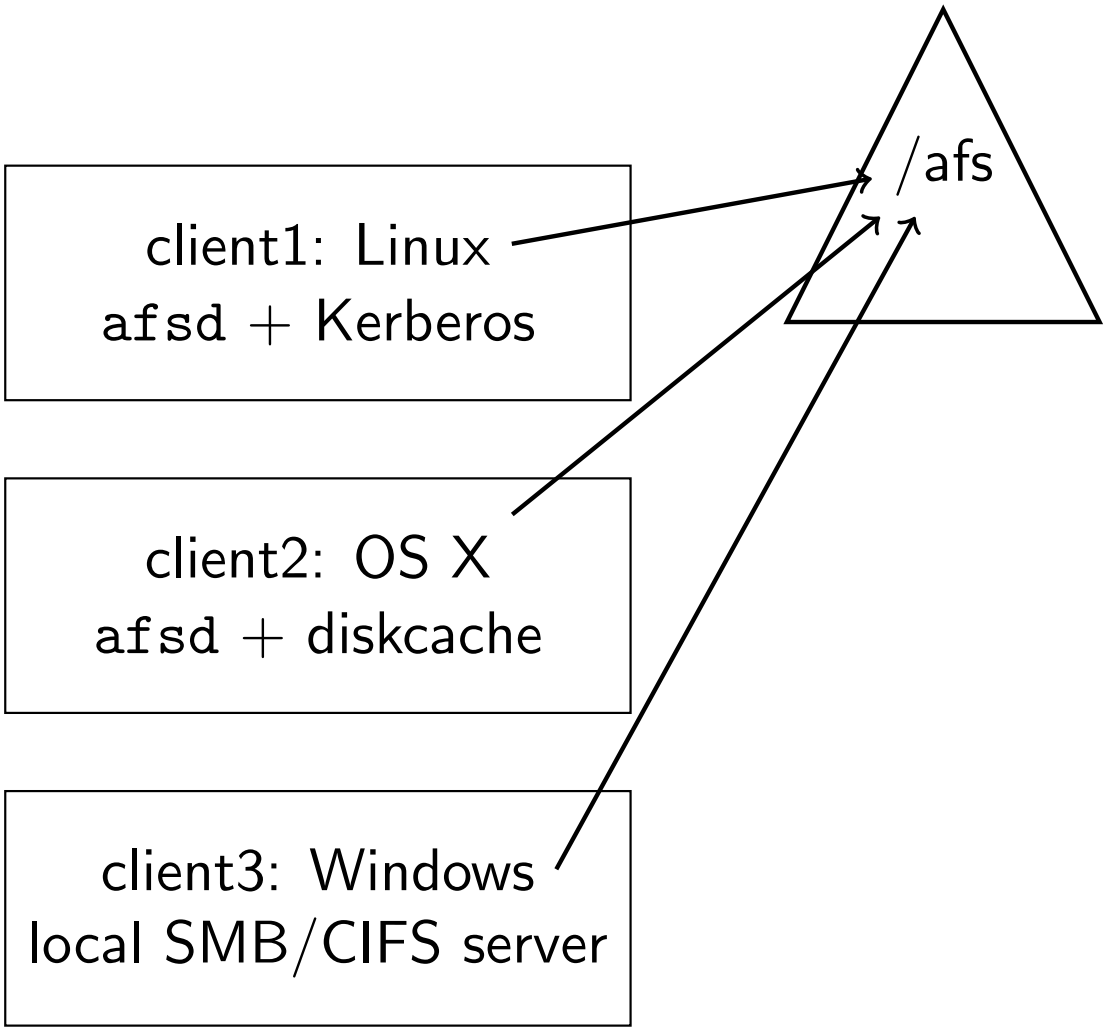
Sketch


client1: Linux
afsd + Kerberos
client2: OS X
afsd + diskcache
/afs

Sketch


/afs
client1: Linux
afsd + Kerberos
client2: OS X
afsd + diskcache
client3: Windows
local SMB/CIFS server

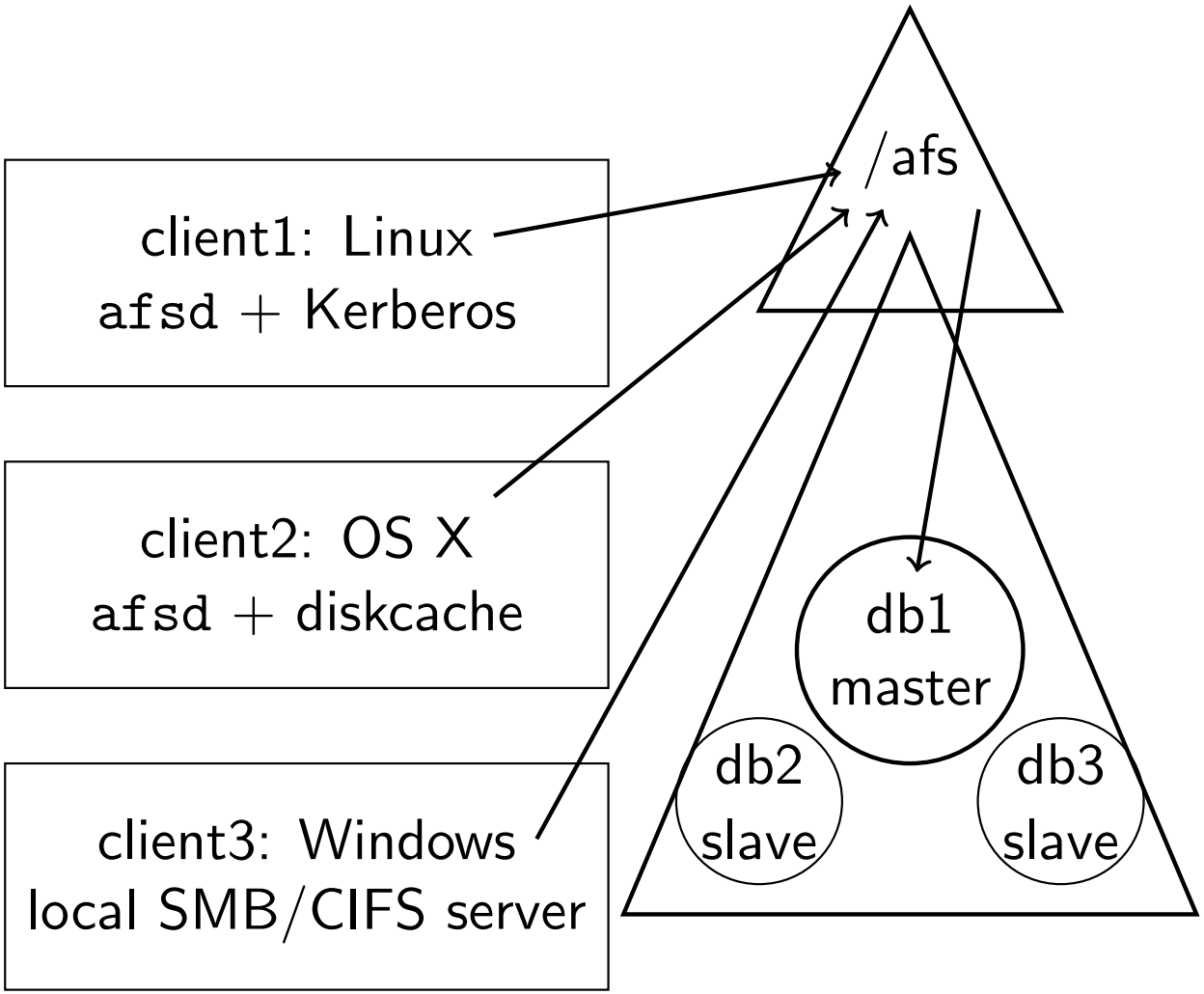
Sketch



. . . , BSD versions, . . .

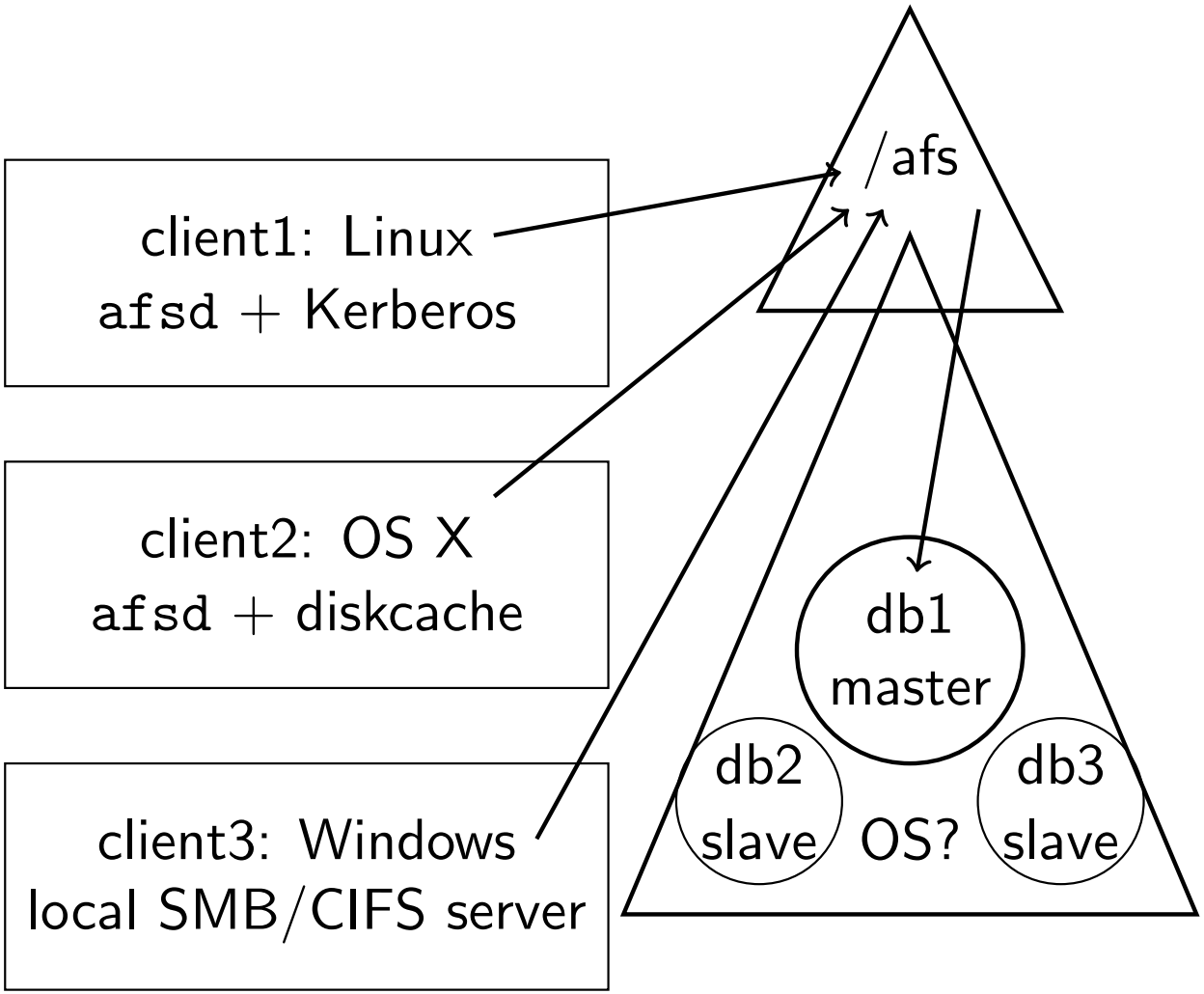
EUROPEAN AFS & KERBEROS CONFERENCE 2010

Sketch



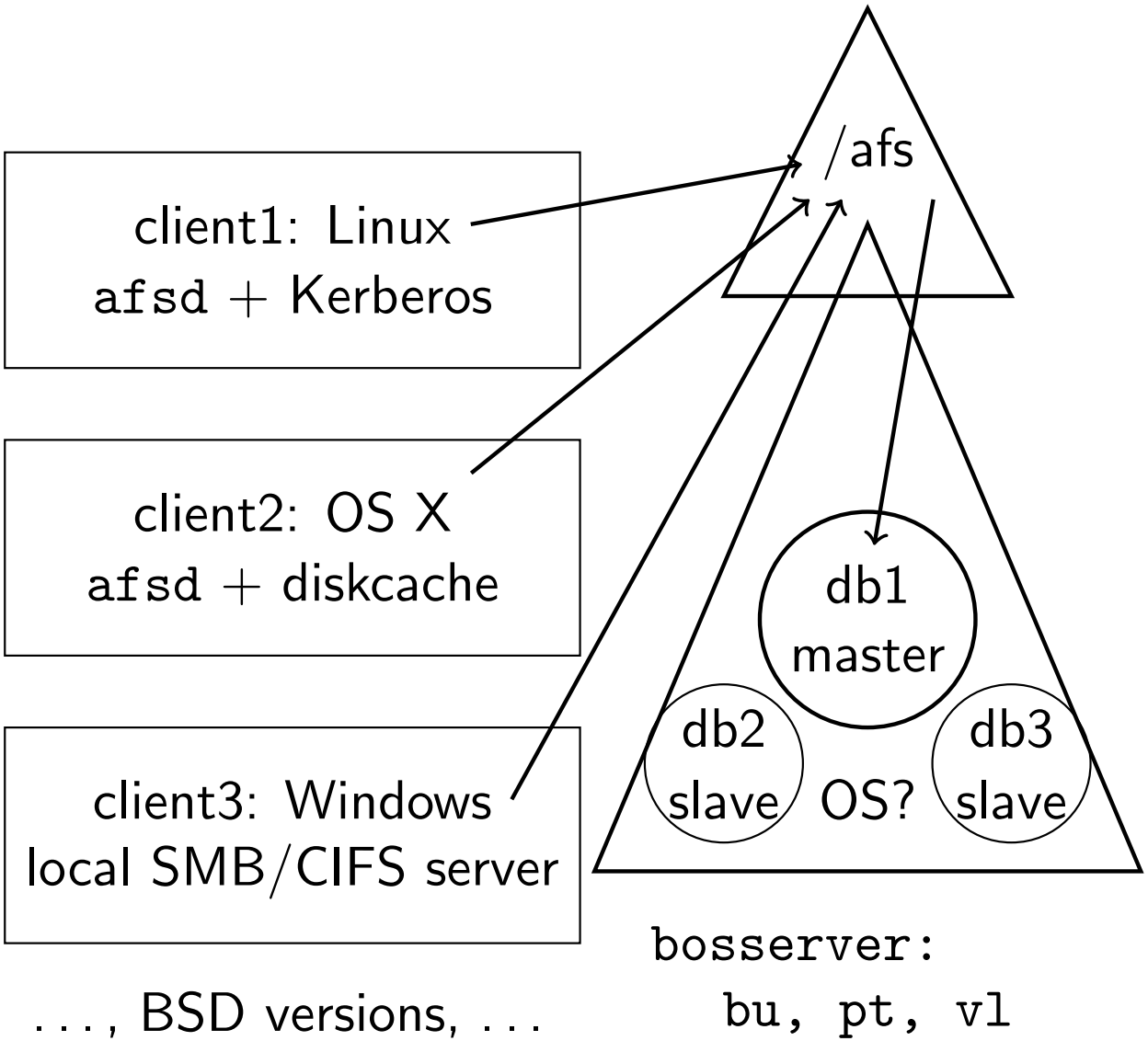
. . . , BSD versions, . . .

Sketch

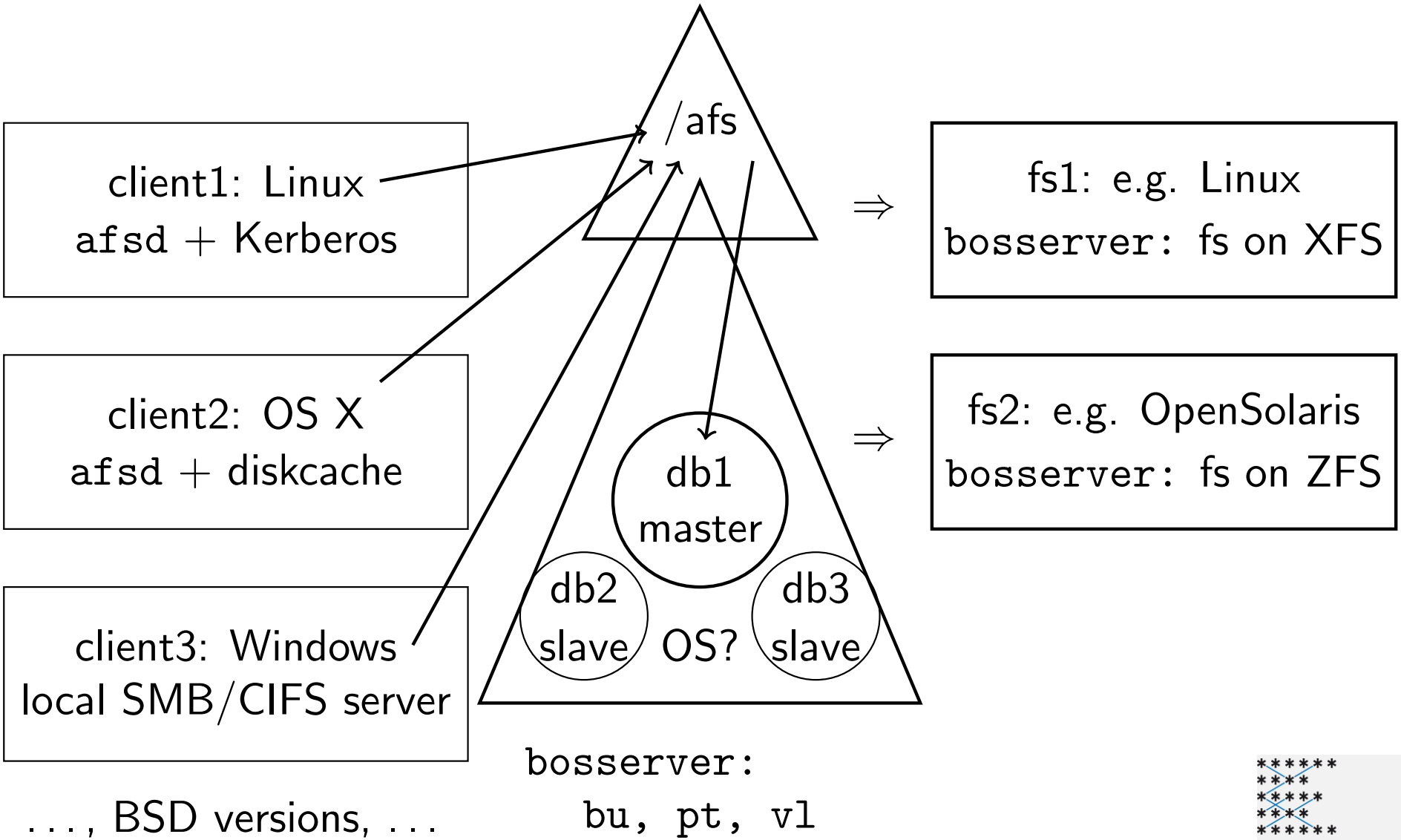


. . . , BSD versions, . . .

Sketch

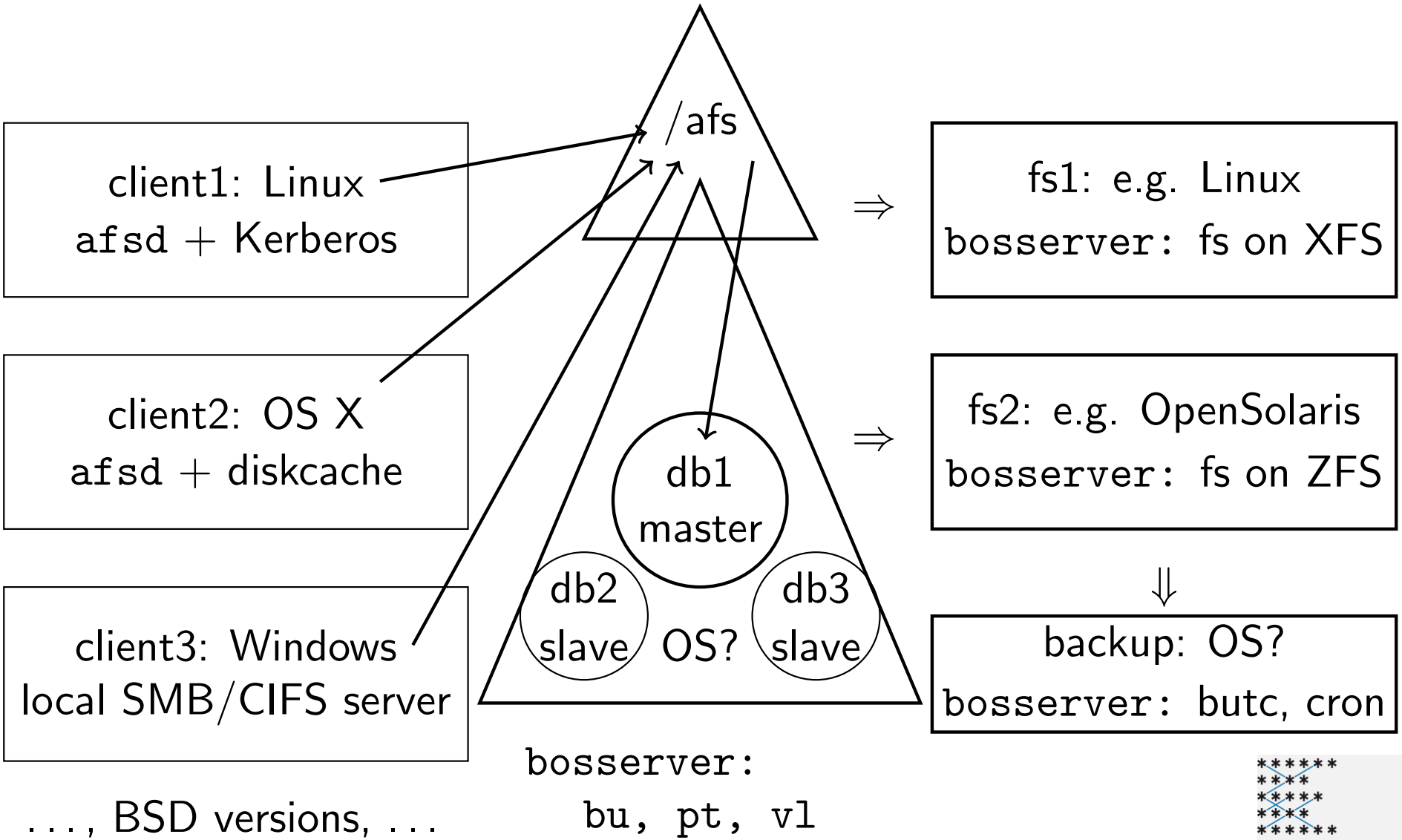

client1: Linux
afsd + Kerberos
client2: OS X
afsd + diskcache
client3: Windows
local SMB/CIFS server
/afs
db1
master
db2
slave
OS?
db3
slave
bosserver:
bu, pt, vl
..., BSD versions, ...

Sketch



EUROPEAN AFS & KERBEROS CONFERENCE 2010

Sketch


client1: Linux
afsd + Kerberos
client2: OS X
afsd + diskcache
client3: Windows
local SMB/CIFS server
. . . , BSD versions, . . .
/afs
db1
master
db2
slave
OS?
db3
slave
bosserver:
bu, pt, vl
⇒
fs1: e.g. Linux
bosserver: fs on XFS
⇒
fs2: e.g. OpenSolaris
bosserver: fs on ZFS
⇓
backup: OS?
bosserver: butc, cron

How about related technology?

device level

- LVM - Logical Volume Manager
- DRBD - Distributed Replicated Block Device
- iSCSI - SCSI over Ethernet

How about related technology?

device level

- LVM - Logical Volume Manager
- DRBD - Distributed Replicated Block Device
- iSCSI - SCSI over Ethernet

local file system

- ext2, ext3, ext4 (only those for disk cache)
- XFS, ReiserFS
- ZFS (btrfs)

How about related technology?

device level

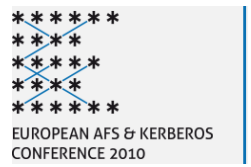
- LVM - Logical Volume Manager
- DRBD - Distributed Replicated Block Device
- iSCSI - SCSI over Ethernet

local file system

- ext2, ext3, ext4 (only those for disk cache)
- XFS, ReiserFS
- ZFS (btrfs)

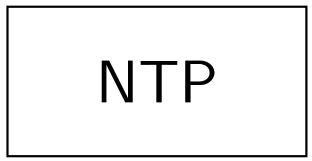
network file system

- NFSv3, NFSv4, DCE/DFS
- Lustre, pNFS
- GlusterFS (Ceph)

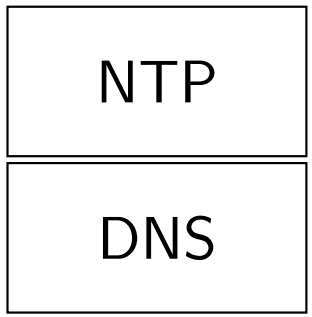


Which is the more general context?

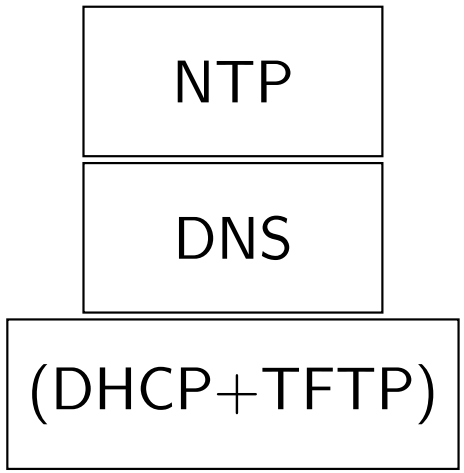
Which is the more general context?



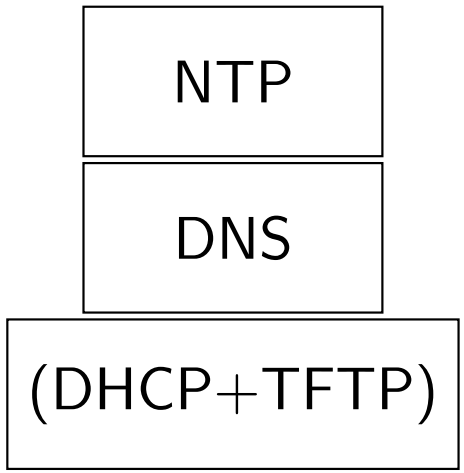
Which is the more general context?



Which is the more general context?

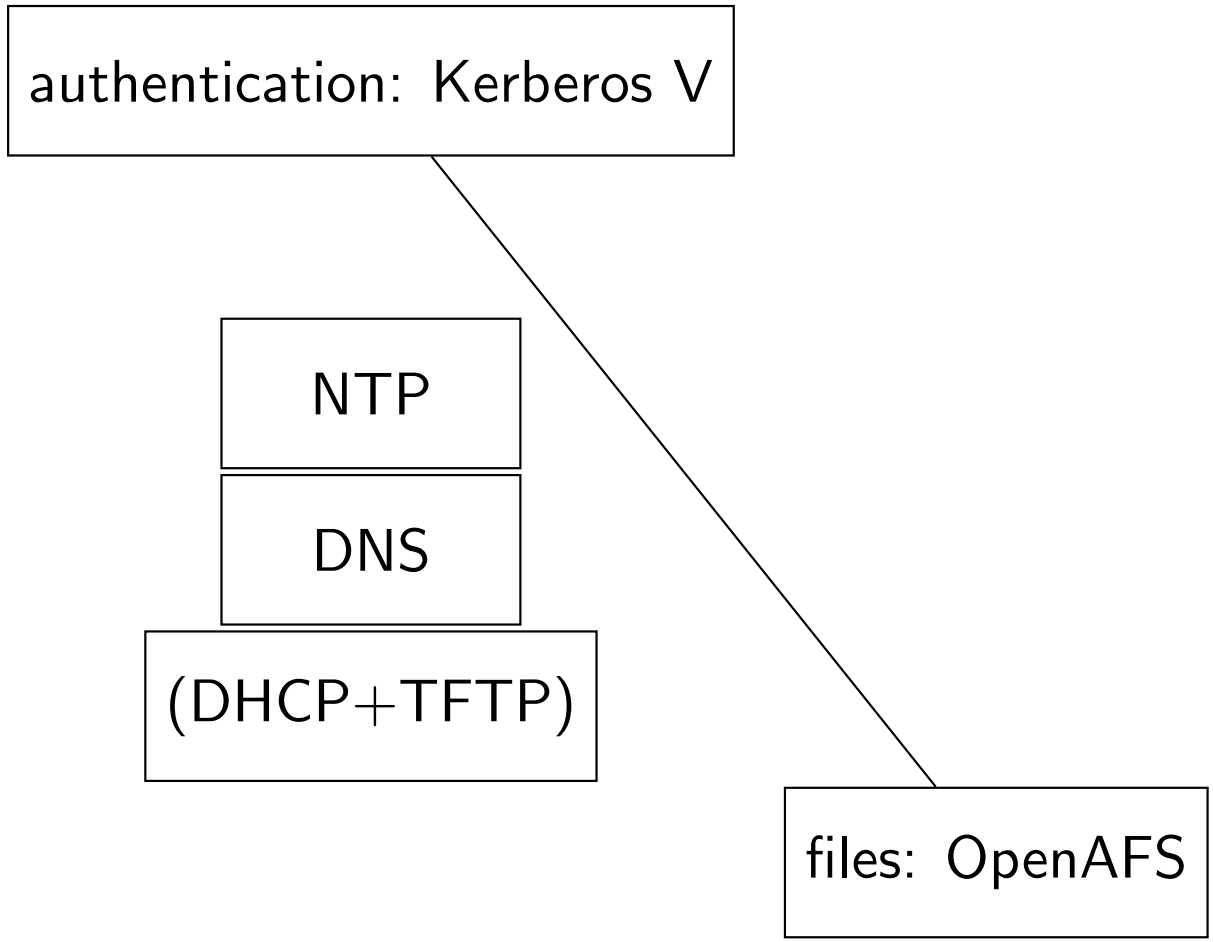


Which is the more general context?



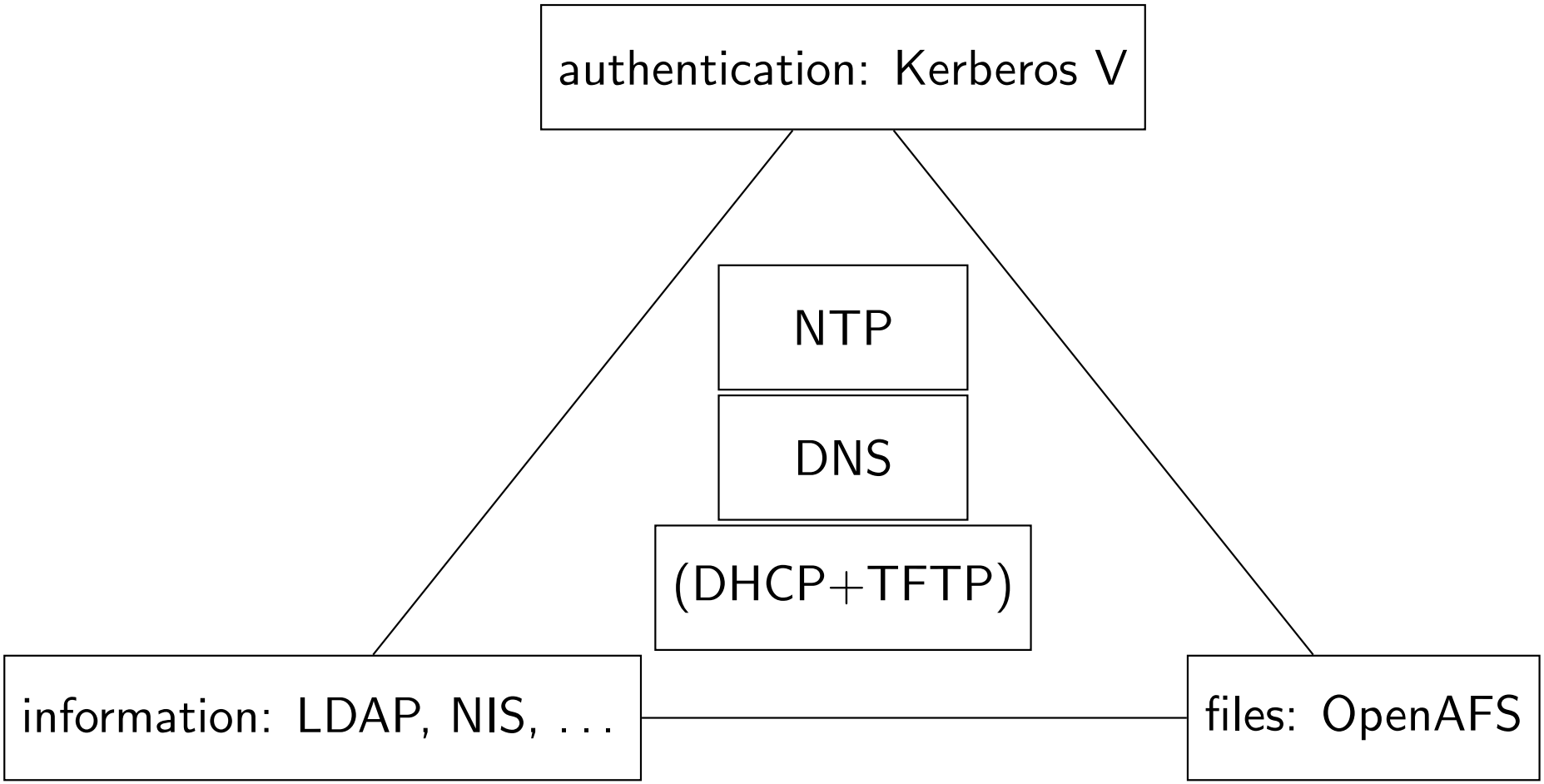
files: OpenAFS

Which is the more general context?



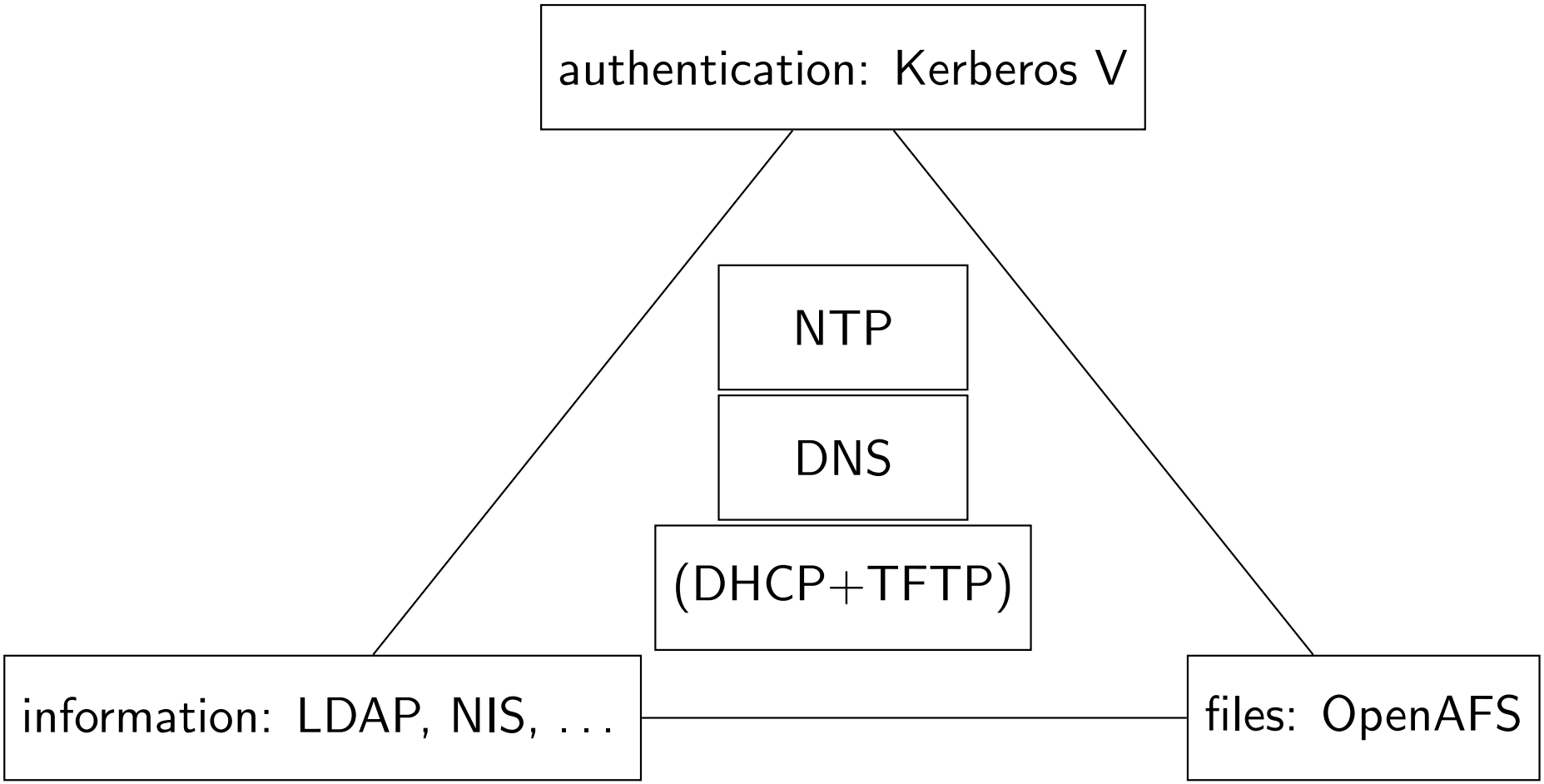
EUROPEAN AFS & KERBEROS CONFERENCE 2010

Which is the more general context?



EUROPEAN AFS & KERBEROS CONFERENCE 2010

Which is the more general context?



redundancy: several servers, leader election,
master-slave, multi-master

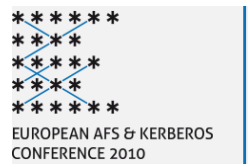
EUROPEAN AFS & KERBEROS CONFERENCE 2010

Network Time Protocol

- synchronizing clocks of computers
- needed for Kerberos authentication
- loose synchronization sufficient ($<$ 5 minutes difference)
- *stratum 0:* atomic clocks
- good: reference time server connected to GPS or radio clock
- better with authentication
- `ntpdate` once an hour could suffice
- otherwise `ntpd` or `OpenNTPD`

Domain Name System

DNS	name to IPv4 or IPv6 address translation
A record	IPv4 address
AAAA record	IPv6 address
AFSDB record	e.g. `<domain>.  IN AFSDB 1 <db name>.`
CNAME record	name alias
NS record	name server
PTR record	name (for reverse lookup)
SRV record	e.g. `_afs3-vlserver._udp SRV 0 0 7003 <db name>.`
SOA record	start of authority
TXT record	e.g. `_kerberos.<domain>.  IN TXT "<REALM>"`
DNSSEC	for more security (roots *just* got signed)



DHCP and TFTP

purpose simplification of client configuration

DHCP and TFTP

purpose simplification of client configuration

Dynamic Host Configuration Protocol

- dynamic IP address assignment
- network and server information
- no security features
- could be linked to a dynamic DNS server

DHCP and TFTP

purpose simplification of client configuration

Dynamic Host Configuration Protocol

- dynamic IP address assignment
- network and server information
- no security features
- could be linked to a dynamic DNS server

Trivial File Transfer Protocol

- very basic form of file transfer
- used for PXE boot with optional checksum
- no authentication and no encryption
- also with *multicast*

Kerberos

- client-server model
- mutual authentication
- symmetric key cryptography
- optionally asymmetric key cryptography (e.g. smart cards)
- trusted third party (KDC = key distribution center)
- time stamped tickets (TGT = ticket granting ticket)
- database of secret keys on the KDC
- requires synchronized clocks on servers and clients
- replication to increase availability

Different possibilities

MIT

- reference implementation
- freely available
- uses policies for e.g. password strength

Different possibilities

MIT

- reference implementation
- freely available
- uses policies for e.g. password strength

Heimdal

- European Kerberos implementation
- early adopter of `pkinit`
- good AFS integration

Different possibilities

MIT

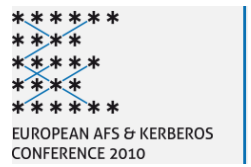
- reference implementation
- freely available
- uses policies for e.g. password strength

Heimdal

- European Kerberos implementation
- early adopter of `pkinit`
- good AFS integration

Active Directory

- Kerberos, LDAP, DNS “mix”
- can be used, too, in AFS context
- necessary in a Microsoft centric environment



Details

REALM decide your realm name (best domain name uppercase)

principle entry in Kerberos data base

ticket authentication data

Details

REALM decide your realm name (best domain name uppercase)

principle entry in Kerberos data base

ticket authentication data

K/M@<REALM> Kerberos data base encryption

kadmin/admin@<REALM> internal

kadmin/changepw@<REALM> internal

kadmin/history@<REALM> internal

Details

REALM decide your realm name (best domain name uppercase)

principle entry in Kerberos data base

ticket authentication data

K/M@<REALM> Kerberos data base encryption

kadmin/admin@<REALM> internal

kadmin/changepw@<REALM> internal

kadmin/history@<REALM> internal

krbtgt/<REALM>@<REALM> attention to lifetimes

afs/<cell>@<REALM> attention to encryption type

krbadmin/admin@<REALM> example Kerberos administrator

afsadmin@<REALM> example AFS administrator

Details

REALM decide your realm name (best domain name uppercase)

principle entry in Kerberos data base

ticket authentication data

K/M@<REALM> Kerberos data base encryption

kadmin/admin@<REALM> internal

kadmin/changepw@<REALM> internal

kadmin/history@<REALM> internal

krbtgt/<REALM>@<REALM> attention to lifetimes

afs/<cell>@<REALM> attention to encryption type

krbadmin/admin@<REALM> example Kerberos administrator

afsadmin@<REALM> example AFS administrator

redundancy master-slave replication or redundant back-end (e.g. LDAP)

EUROPEAN AFS & KERBEROS CONFERENCE 2010

Again choices

Network Information Service

- oldest standard
- only for certain types of information
- widely available for UNIX

Again choices

Network Information Service

- oldest standard
- only for certain types of information
- widely available for UNIX

Lightweight Directory Access Protocol

- more modern and secure naming service
- more flexible for new types of information
- can be integrated with Kerberos

Again choices

Network Information Service

- oldest standard
- only for certain types of information
- widely available for UNIX

Lightweight Directory Access Protocol

- more modern and secure naming service
- more flexible for new types of information
- can be integrated with Kerberos

PTS

- as name service switch library
- just for Linux with `/etc/nsswitch.conf`
- different implementations available

OpenLDAP

- most flexible and recyclable for other tasks
- best search base similar to domain name
- SASL GSSAPI `ldap/<hostname>@<REALM>`
  `authz-regexp` combined with suitable `rootdn`
- define administrative user for LDAP
- LDAP sync replication
- `posixGroup` schema for groups
- `posixAccount` schema sufficient for users
- adding `inetOrgPerson` schema might be better

Possible Extensions

- OpenLDAP allows for N-way multi-master
- Kerberos server with LDAP back-end
- DNS server with LDAP back-end
- DHCP server with LDAP back-end
- certificate storage in directory
- OCSP or CRLs over LDAP

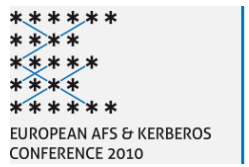
Hands on!

Since we now know a lot about AFS
we should get our hands dirty.

Has everyone got a working client?

Overview

- looking at AFS from the clients point of view
- path is constructed from volumes
- what are volumes?
- how does client know how to get to the right fileserver?
- users & access rights
- how are ACLs enforced?



First words

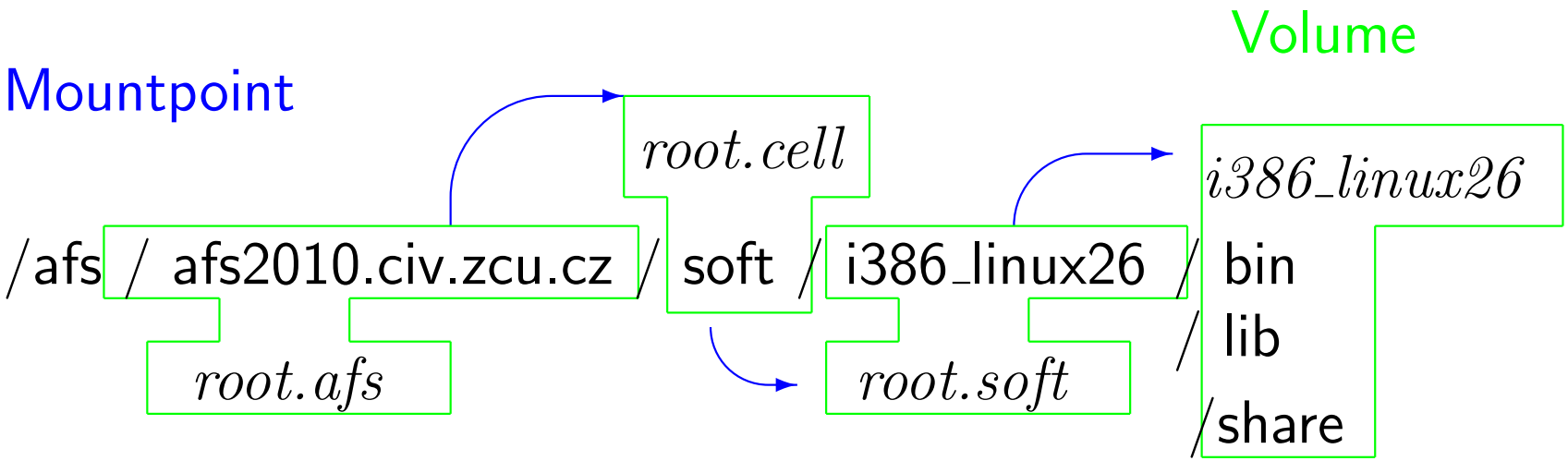
AFS	a world-wide accessible distributed filesytem
AFS cells	administratively independent subunits
approximately	AFS is a web/grid/cloud of distributed filesystems
AFS clients	belong to one cell, but can access others

Volumes and paths I

Path:

/afs / afs2010.civ.zcu.cz / soft / i386_linux26 / bin

Behind the scenes :



Volumes are a basic building block of the AFS-space

They are connected by mountpoints

What are volumes?

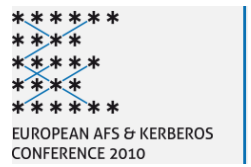
Volumes are finite directory trees, consisting of the usual:

- files
- symbolic links
- hardlinks within the same directory
- directories
- mountpoints to other volumes

Following objects cannot be in AFS:

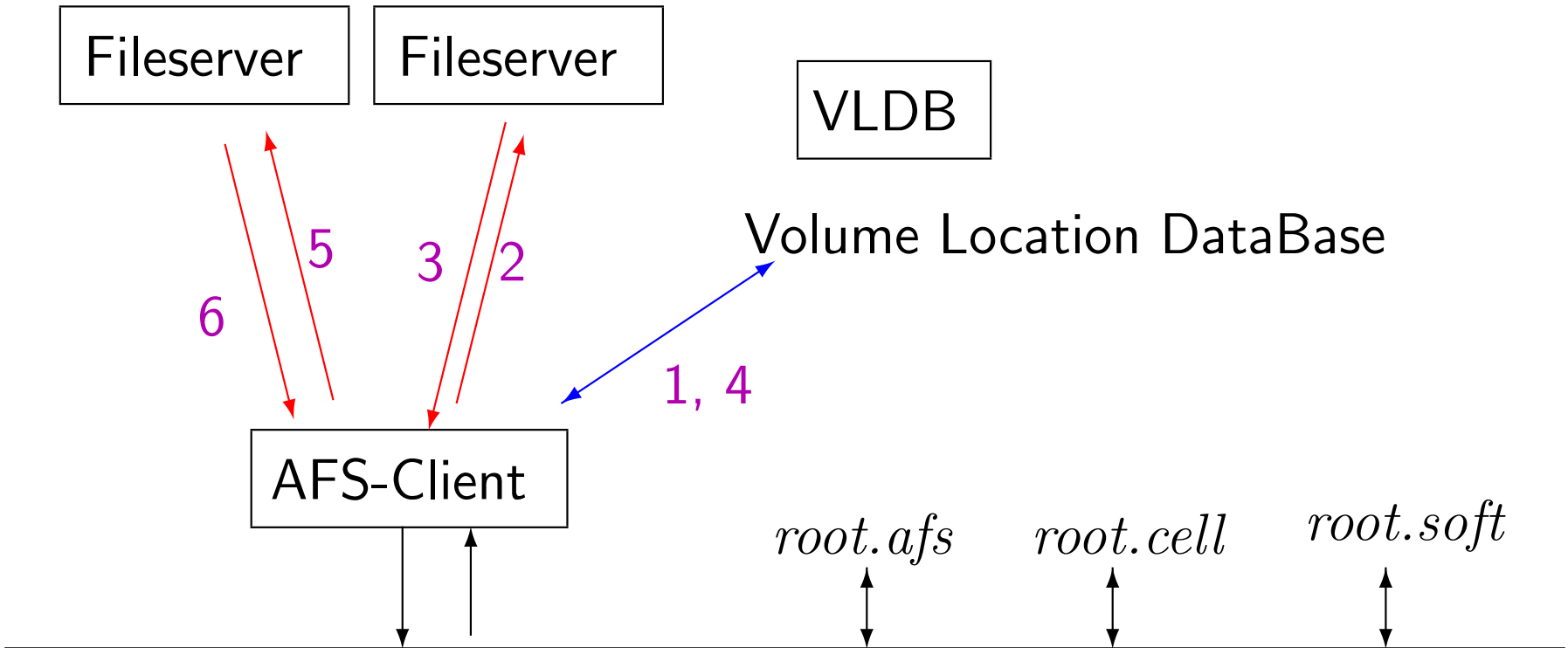
- unix-sockets
- named pipes

Volumes can be moved from one server to another, without disturbing the work of the client.



Accessing AFS by a client I

How does the client know where to fetch the data ?



EUROPEAN AFS & KERBEROS CONFERENCE 2010

Types of volumes

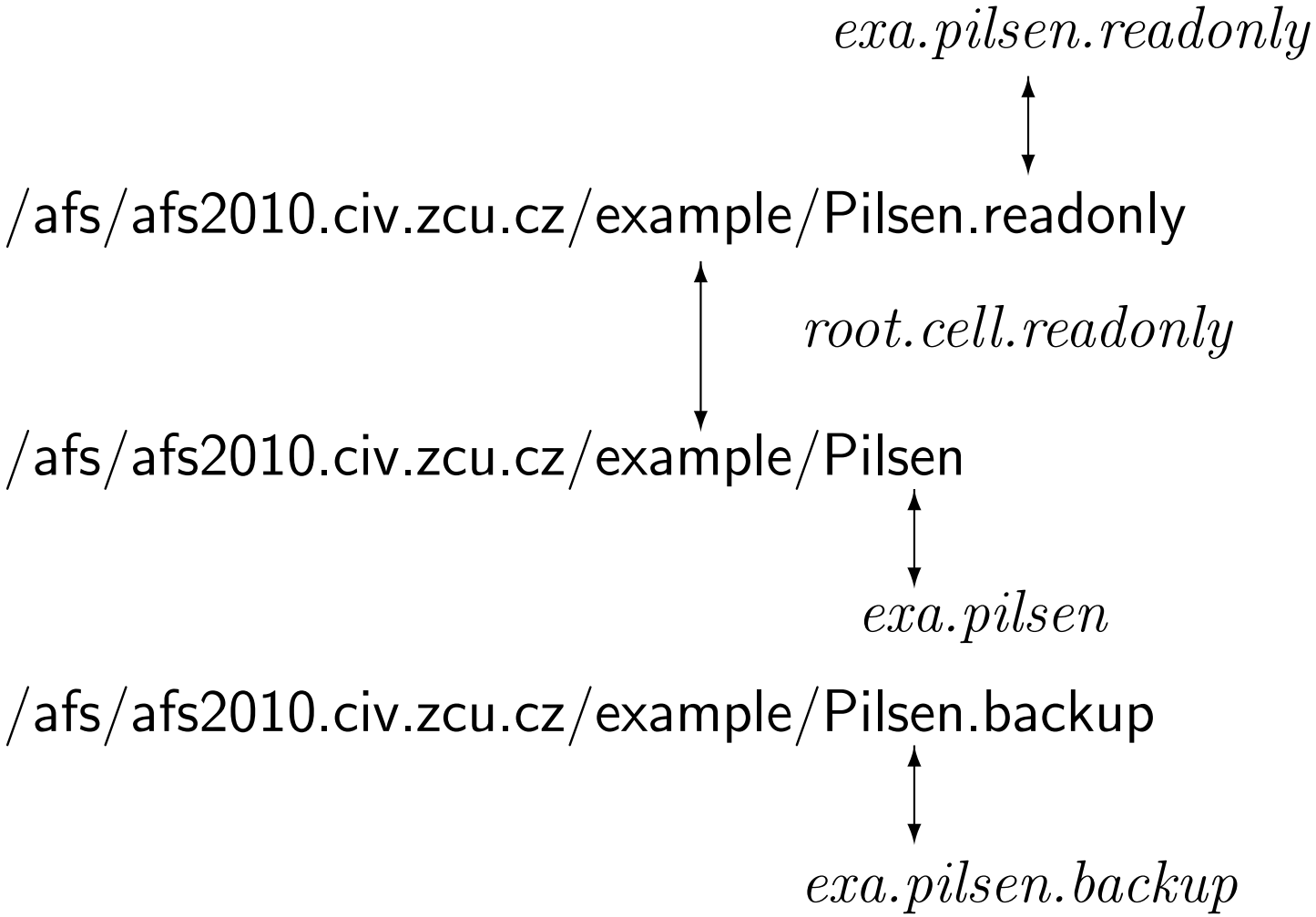
A volume is a finite directory tree.
It can be moved from one server to another
without reconfiguring the client.

Read-Write	where you can change the data
Read-Only	RO replica, which must be synchronised by a "release"
Backup	snapshot of a RW on same server and partition
Clone	intermittent server only copy (used when releasing a volume)
Shadow	bit of a hack, just do not use it

Only up to 4 RO volumes per RW volume are possible!

Accessing AFS by a client II

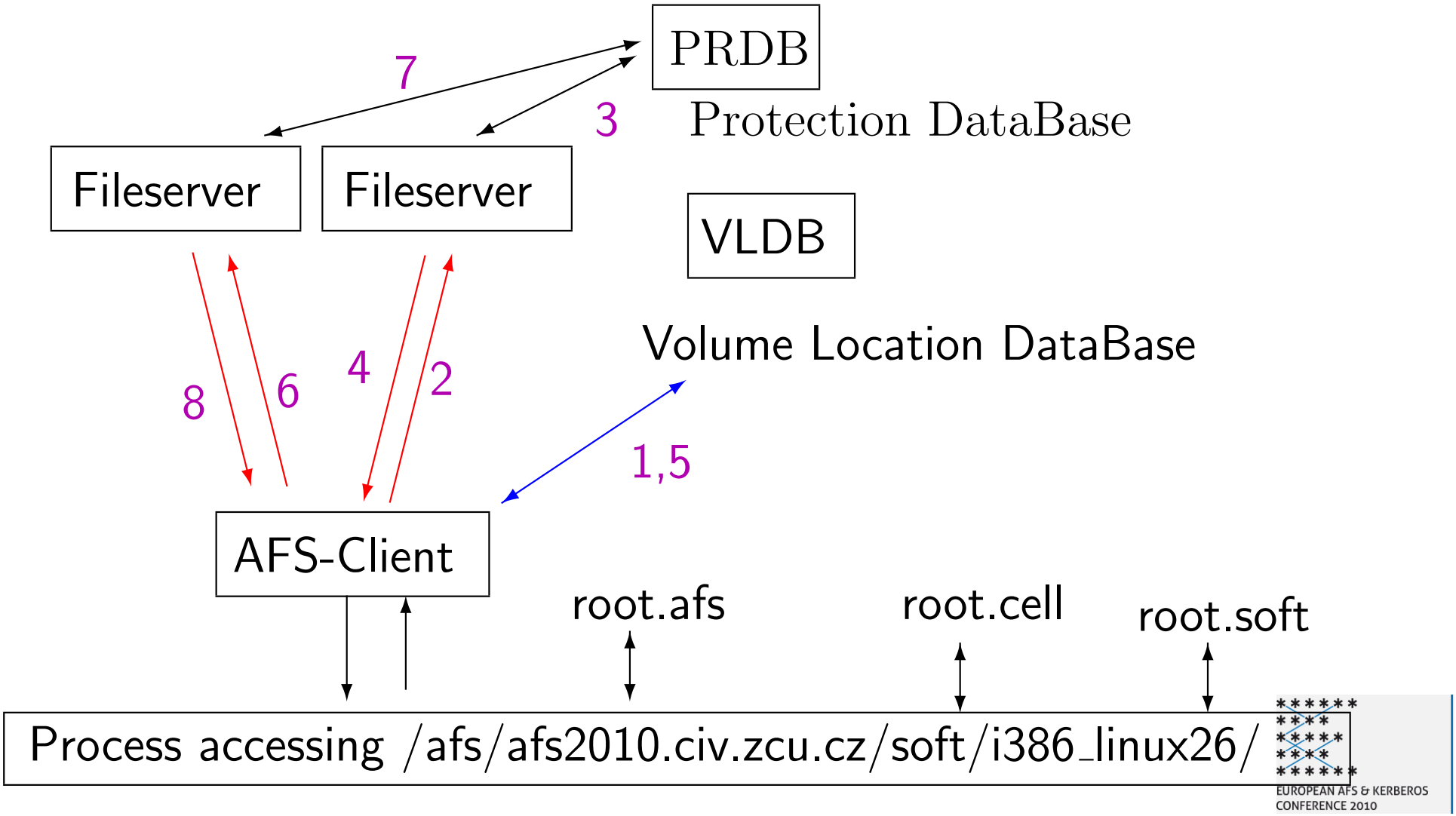
Difference between RO and RW Volumes



EUROPEAN AFS & KERBEROS CONFERENCE 2010

Accessing AFS by a client II

How is access control working ?



EUROPEAN AFS & KERBEROS CONFERENCE 2010

AFS-users

- AFS has its own userdatabase, managed by the PTServer,
- independent of the OS-userdatabase.
- correlated to the Kerberos-database.
- groups and groups of groups are possible.

Types of servers

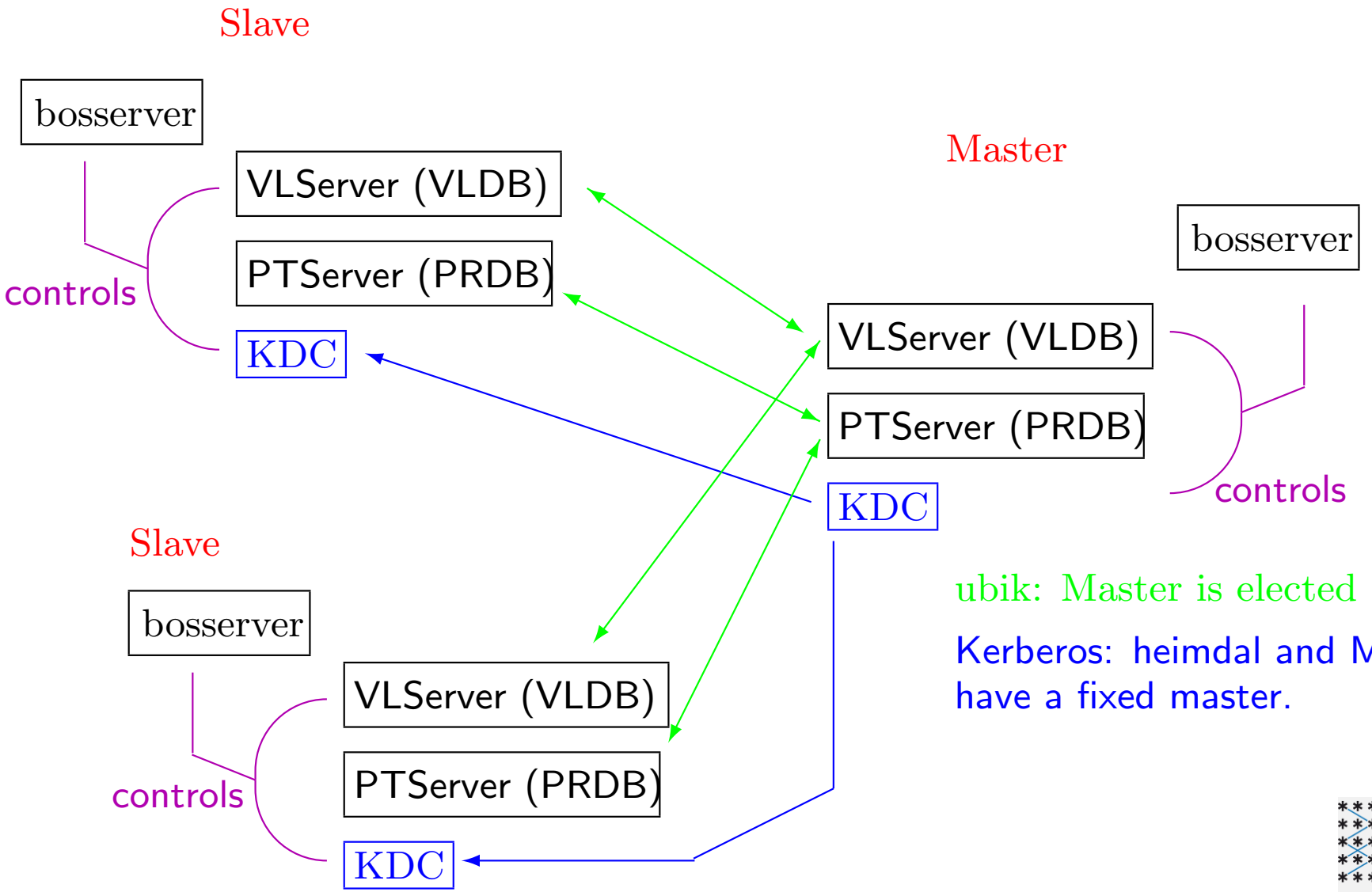
database server metadata and authorisation

- replication with leader election (3 is a good number)
- programs:
  - bosserver
  - vlserver instance
  - ptserver instance

file server

- actual file content
- programs:
  - bosserver
  - fileserver instance
    - ★ fs
    - ★ volserver
    - ★ salvager

Database server setup

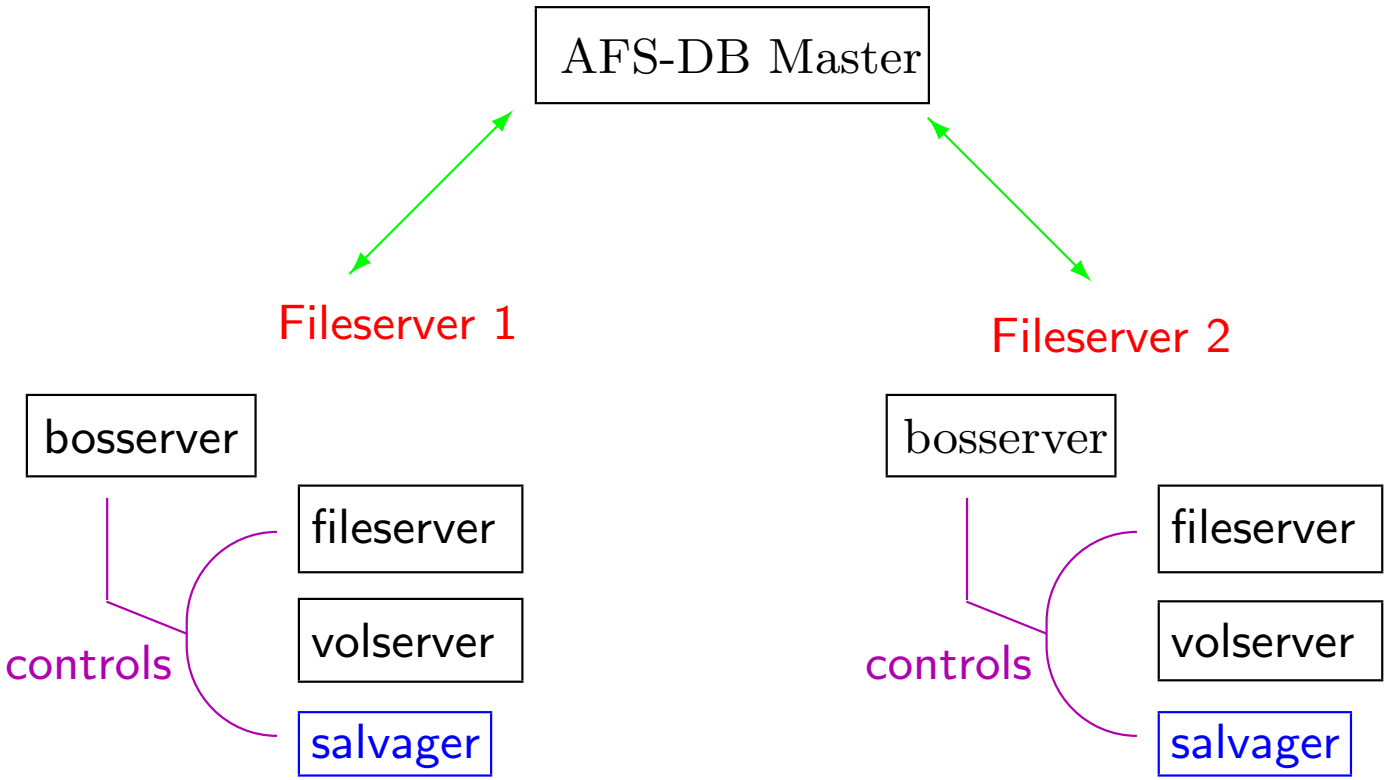


ubik: Master is elected

Kerberos: heimdal and MIT have a fixed master.

EUROPEAN AFS & KERBEROS CONFERENCE 2010

Fileserver setup



Fileservers are largely independent of each other.
Only RW/RO - volumes couple them.

Junction point

What now ?

- Setting up a cell
- First steps in a cell
- Advanced topics

Requirements

- kerberos-kdc
- afs db-server
- afs fileserver
- optional: clients

Installation requires some special "bootstrapping".

names: more than Schall und Rauch

Kerberos realm domain name in capital letters.

cell name Kerberos realm

AFS database-servers 3 IP-addresses for 3 servers, master is lowest.

Caution: Sometimes, direct pathes are referred to.
The correct pathes for your installation should be in the man-pages.

Kerberos

If you have access to a Windows-AD, use it.
instructions not implemented yet.

service principal

- afs/<cell>@<REALM>
- verify properties

export as keytab

- just with DES encryption
- verify exported file

import into KeyFile

- `asetkey add` command
- verify import `asetkey list` or later `bos listkeys`

Database-server

start bosserver with `KeyFile` (or start with "-noauth")

set cell name `bos setcellname <hostname>` (or edit "<afsconfdir>/ThisCell" file)

add DB servers `bos addhost` (or edit "<afsconfdir>/CellServDB)

check servers `bos listhosts`

create instances `bos create`

- `ptserver`
- `vlserver`

check status `bos status`

File-server

copy configuration including KeyFile – securely !

create Fs-partitions up to 255: /vicepa-/vicepiu

start bosserver KeyFile and config in place ?

create fs-instance `bos create -server <hostname> -instance fs -type fs -cmd` *commands*

check status `bos status <hostname>`

First user and volume

Kerberos create a Kerberos principal e.g. "admin"

create AFS-user `pts createuser admin`

add to admin-group `pts adduser admin system:administrators -localauth`

check group `pts membership system:administrators`

check user `pts examine`

verify partitions `ls /vicepa .`

verify volume `vos examine`

Preparation of a client

compile kernel module
- on Debian Linux with `module-assistant`
- rpm-based distributions : install kmp-packages

configure client
- `ThisCell`, `CellServDB`, `cacheinfo`
- Cache -Type and -Size
- `dynroot`, `afsdb`

start client `/etc/init.d/openafs-client start`
- df /afs
- linux: lsmod — grep afs
- fs wscell

authenticate `kinit admin` and `aklog`.

Preparing path-layout of the cell

basic pathes point to different volumes

/afs AFS root *root.afs*

/afs/<cell> *root.cell.readonly*

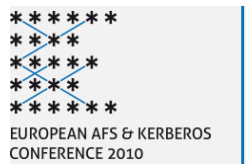
/afs/.<cell> *root.cell*

/afs/<cell>/user *root.users*

- /afs/<cell>/u/<user> *user.¡username¿*
- /afs/<cell>/home/<u>/<user>

/afs/<cell>/software root for software

- /afs/<cell>/software/common
- /afs/<cell>/software/i386_linux24 *i386_linux24*
- /afs/<cell>/software/i386_linux26 *i386_linux26*



Replication of infrastructure-volumes

add replication site `vos addsite`

release the volume `vos release`

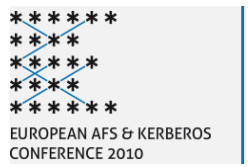
good practice replicate all infrastructure volumes

check volumes `vos examine root.cell`

More on these commands later.

Overview

- Pathes: RW or not RW ?
- Volume operations
- Quota
- ACLs
- Troubleshooting
- Limits



Volumes and paths II: RW or not RW?

Mountpoints can point to either RW, RO, or BK volumes
"fs listmount" provides the information

RW-mountpoint %*volumename*

- "fs makemount *volumename Path* -rw"
- always leads to RW-Volume

advisory RO-mountpoint #*volumename*

- "fs makemount *volumename Path*"
- leads to RW if we are in a RW-Volume
- leads to RW if there is no RO-Volume

mandatory RO-mountpoint #*volumename*.**readonly**

- "fs makemount *volumename*.**readonly** *Path*"
- always leads to RO-Volume

BK-mountpoint #*volumename*.**backup**

- "fs makemount *volumename*.**backup** *Path*"

Volume operations I: single volume operations

vos create	create a volume, add entry in VLDB
vos move	move between file servers and/or partitions
vos remove	destroy volume, remove VLDB entry
vos release	synchronise ROs with RW, including ACLs!
vos zap	delete the volume, do not bother with VLDB
vos addsite	add replication site to VLDB *only*
vos delsite	delete replication site from VLDB *only*
vos rename	rename a volume $\rightsquigarrow$ *old mountpoints invalid!*
vos examine	show information about a volume

VLDB does not necessarily reflect the reality on fileserver

Volume operations II

vos listvol show volumes on a server

vos listvldb show volumes in VLDB

vos syncserv sync central VLDB $\rightsquigarrow$ local server

vos syncvldb sync local server $\rightsquigarrow$ central VLDB

vos listaddrs show ipaddr of severs registered in the VLDB

- beware of 127.0.0.1 entries
- remove obsoleted entries

vos listpart list partitions on a server

vos partinfo show partition info on a server

Fileserver operations

`fs` command queries the client and the fileserver

- fs listacl shows ACLs of a directory
- fs setacl sets ACLs of a directory
- fs setquota set quota
- fs listquota show quota
- fs examine shows details about a file/dir
- fs where shows location of a server

fs getserverprefs shows preference of fileserver

fs setserverprefs sets preference of fileserver

Special path element @sys

The path-element *@sys* is a placeholder only and replaced by the afs-client:

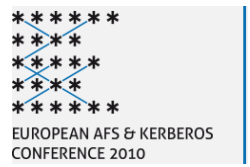
- usually after an OS like i386_linux26
- used to distinguish client OS

Example:

/afs/afs2010.civ.zcu.cz/@sys/bin could expand to :

- /afs/afs2010.civ.zcu.cz/i386_linux26/bin/
- /afs/afs2010.civ.zcu.cz/sun4x_59/bin/

Putting /afs/<cellname>/@sys/bin in your search-path helps.



Quotas

Space (blockquota) is based on volumes.
It can be exceeded by a few bytes, because of the transfer protocol.
(Just in case you wondered: it is not a hard-hard limit.)

set a quota fs setquota *path*

show a quota fs listquota *path*

The previous commands also show partition usage.
Problematic, since only 32bit and ambiguous for RO-volumes.

Users and groups

server PT-server (PR-server)

users normal or IP-addresses

groups system owned or user owned

supergroups group of groups

ACLs are given on directory level

IP ACLs should not be given to an IP directly, but to a group containing them

predefined groups

- system:administrators
- system:backup
- system:anyuser
- system:authuser
- system:ptsviewer

About ACLs

- i insert files
- d delete files
- r read files
- w write files
- k lock files
- l list directory (lookup)
- a administer (give rights to others, create mountpoints)

About ACLs

- i insert files
- d delete files
- r read files
- w write files
- k lock files
- l list directory (lookup)
- a administer (give rights to others, create mountpoints)

Abbreviations

read	rl
write	rlidwk
all	rlidwka
none	-

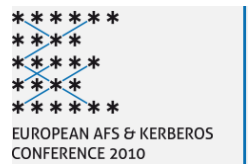
About ACLs

- i insert files
- d delete files
- r read files
- w write files
- k lock files
- l list directory (lookup)
- a administer (give rights to others, create mountpoints)

Abbreviations

read	rl
write	rlidwk
all	rlidwka
none	-

negative ACLs are possible but complex



Notes about ACLs

ACLs are stored at directory level

- at least "l" right required to traverse
- difference read and lookup (????? for stat())
- *comment:* RW must be released such that ACLs get valid in ROs
- *dirty trick:* use symlinks to have different ACLs within on directory.

Credentials

ticket	Kerberos credential stored in the "CC", usually a file
token	derived from Kerberos ticket in kernel-space (key-ring)
pagsh	group of processes which share the same token
krb-equiv	group of processes using the same Credential Cache

Beyond your cell

- just try `/bin/ls /afs`
- do not try to index `/afs`
- you might use the `"-dynroot"` option for `afsd`
- perfectly possible to acquire and hold tokens for multiple cells

User encountered problems

use *translate_et* when only errorcode is available

Permission denied no token, user is not allowed to traverse path

No such device mountpoint leads to non-existent volume

File too large too many entries in this directory

Invalid cross-device link hard links across directories are not allowed (may be other volume)

Cannot get token client-time correct?

Connection timed out Client could not reach fileserver for a while, thus refusing to enter volumes on this fileserver. fix with *fs checkserver*

When really strange things happen, try "bos salvage".
Could happen when partition gets full.

Admin view

vos release	salvage the volume remove obsoleted servers from VLDB
on lost fileserver	vos convertROtoRW requires RO on a different fileserver
network	use NetRestrict/NetInfo files to restrict network traffic

Hints on debugging

date	verify correct time
kinit/klist	get and test a Kerberos ticket
aklog/tokens	get and test an AFS token
rxdebug	for Rx protocol
udebug	for Ubik protocol
cmdebug	for cache manager
/etc/services	to find related ports or `man`
bos status	check instances
vos listvldb	list the VLDB
vos listvol	list volumes on a server
vos examine	for state of a specific volume
fs whereis	get location of file
fs examine	get file status
translate_et	translates error code numbers

Limits

You stay save and sane, if you keep within:

max size of vicep partition 2 TB

max size of volume 2 TB

max number of ROs 4

max number of files in directory theoretically 64,000
in practice 16,000 – 24,000

Overview

- side-effects of caching
- side-effects of distributed nature
- callbacks
- simple backup mechanism

ACLs are checked by the fileserver

issue ACLs are checked by the fileserver

effect data lying in the client cache can be accessed by any user on that machine.

example try directory listing with and without tokens
use fs flushvolume to demonstrate

Following output I

If an application just appends to a file,
its data will remain on the client for a while
and cannot be followed from other clients.
(Problem for output files from simulations).

```
1  #!/usr/bin/env python
2
3  import time
4  AFSBaseDir="/afs/ipp-garching.mpg.de/u/hanke/Vorlagen/"
5  FileName="growing.txt"
6  f=file("%s/%s" % (AFSBaseDir,FileName),"w+")
7  while 1:
8      f.write("Next line at %s\n" %
9          time.asctime(time.localtime()))
10     time.sleep(1)
11     f.flush()
12 f.close()
```

Following output II

Using an explicit fsync() forces the afs-client
to sync all data with the fileserver.
Then, the file can be followed from other clients.
A flush alone is not enough!

```
1  #!/usr/bin/env python
2
3  import time,os
4  AFSBaseDir="/afs/ipp-garching.mpg.de/u/hanke/Vorlagen/"
5  FileName="growing.txt"
6  f=file("%s/%s" % (AFSBaseDir,FileName),"w+")
7  while 1:
8      f.write("Next line at %s\n" %
9          time.asctime(time.localtime()))
10     time.sleep(1)
11     f.flush()
12     os.fsync(f.fileno())
13 f.close()
```

Deleting opened files

situation	client has opened file "qwe" (e.g. with "less")
then	in another shell, we delete this file
result	first client renames this file "qwe" to .__afs_XXX, where "XXX" is a random sequence
finally	this file is deleted when it is closed (application terminates)
caveat	file can be leftover in case of network-problems

Callbacks

If a file is changed on one client, the fileserver needs to inform the other clients which have cached this file to throw it all away.

This is required by the client-side caching.
We show here an example, where one (all) client(s) is (are) denied the callback (network interrupt).

If they have the filedata still cached, they serve the outdated data.
If not, they cannot access the files, since the fileserver cannot reach the client anymore.

Multiple IPs on a client

- sometimes called "multihomed".
- Beware of private managment networks,
- Can cause trouble on fileservers :
  private addresses in VLDB is a bad idea (TM).
- use NetInfo/NetRestrict for this.
- Different for client and servers!

Backup possibilities

vos release gives users access to a previously released RO volume

vos backup create a backup volume

vos backupsys mass backup volume creation

bos create allows for cron instances to backup

Backup possibilities

vos release gives users access to a previously released RO volume

vos backup create a backup volume

vos backupsys mass backup volume creation

bos create allows for cron instances to backup

vos dump to a file

vos restore from a file

vos copy to another server/partition with another name

vos convertROtoRW restore a RW from a remaining RO

Backup possibilities

vos release gives users access to a previously released RO volume

vos backup create a backup volume

vos backupsys mass backup volume creation

bos create allows for cron instances to backup

vos dump to a file

vos restore from a file

vos copy to another server/partition with another name

vos convertROtoRW restore a RW from a remaining RO

/dev/backup name of a possible backup device

tapeconfig dimension of backup device

CFG_backup configuration parameters of backup device

butc can run as instance of `bosserver`

backup user interface for backing up

Simple backup mechanism

ACLs must be stored outside the actual backup and reapplied after restore. Just a simple scheme how you could setup some AFS aware backup:

create tar	tar cjf /tmp/backup.tbz .
collect ACLs	find . -noleaf -type d -exec fs listacl {} \; > backup.acls
extract tar	tar xjf backup.tbz
re-apply ACLs	fs setacl . . . in a shell script

looking around in other cells

- some vos commands don't need authentication
- afs-client not the only way to access data!
  vos dump / dumptool

Web

Kerberos

- authentication
- SPNEGO method

Web

Kerberos

- authentication
- SPNEGO method

LDAP

- UserDir
- vhost configuration

Web

Kerberos

- authentication
- SPNEGO method

LDAP

- UserDir
- vhost configuration

OpenAFS

- authenticated web server
- WebDAV with autoversioning

Mail

Kerberos

- SMTP authentication
- IMAP authentication

Mail

Kerberos

- SMTP authentication
- IMAP authentication

LDAP

- mail aliases
- user certificates

Mail

Kerberos

- SMTP authentication
- IMAP authentication

LDAP

- mail aliases
- user certificates

OpenAFS

- authenticated mail server
- Maildir(++)

Collaboration

Kerberos

- authentication
- also over SSH

Collaboration

Kerberos

- authentication
- also over SSH

LDAP

- `mod_authz_svn` just reads a file
- can be generated from a directory

Collaboration

Kerberos

- authentication
- also over SSH

LDAP

- `mod_authz_svn` just reads a file
- can be generated from a directory

OpenAFS

- project setup (admin group, member group)
- FSFS backend for *subversion*

And more

Kerberos

- single sign-on solution including web
- more services e.g. PostgreSQL

And more

Kerberos

- single sign-on solution including web
- more services e.g. PostgreSQL

LDAP

- own configuration stored inside
- user modifiable attributes

And more

Kerberos

- single sign-on solution including web
- more services e.g. PostgreSQL

LDAP

- own configuration stored inside
- user modifiable attributes

OpenAFS

- configuration data (e.g. for cluster nodes)
- read only content (e.g. for TFTP server)

Further reading

- manual pages
- http://docs.openafs.org/index.html
- http://openafs-wiki.stanford.edu/AFSLore/WebHome/
  (you can help using openID!)
- mailing-lists see http://www.openafs.org/
- Alf Wachsmann "Tutorial: Introduction to AFS and its Best Practices"
- Ken Hornstein "Tutorial: Introduction to Kerberos"
- Richard Campbell
  "Managing AFS: The Andrew File System"
  Prentice-Hall 1998
- Franco Milicchio, Wolfgang A. Gehrke
  "Distributed Services with OpenAFS"
  Springer 2007

Are there any questions?