



# Private-Clouds 1/2

Warum?

rsync, unison,

Powerfolder, Seafile,

ownCloud, ownCloud@KNF

Klaus Franken



# Yes, we scan!



# Warum keine fremden Server/ Services?

- USA zwingt alle Anbieter zur Preisgabe der Daten, auch dann wenn der Server nicht in USA steht (Heise).
- Dropbox: Looking Inside the (Drop) Box: Forscher reverse-engineerern Dropbox Client, können Accounts übernehmen.
- Adobe: > 9000 Mitarbeiter, 3,8 Milliarden USD Umsatz
  - Diebstahl von 153.000.000 Accountdaten



# Es darf nicht nur Cloud geben!

- Es muß Wahlfreiheit geben.
- Suche nach Lösungen zum Selberbetreiben immer schwieriger.



# Die Aufgabe

- automatisch **Dateien** syncen zwischen verschiedenen Geräten
- via Internet, mobil, online, offline
- Datei/Ordner-Freigaben an andere Benutzer
- ohne Fremdanbieter (Server)
  - zur Not proprietäre Software
- **Sicherheit:** Schutz vor Fremdzugriff



# Produkte „Nein“:

- Dropbox
- Strato
- Apple
- OnDisk
- Google Drive
- Leitz Cloud
- SpiderOak
- BtSync (Bittorrent)
- AeroFS
- Wuala



# Produkte „Vielleicht“:

- Goodsync
- Synology
- Teamdrive
- Tahoe-LAFS
- iFolder (Novell)
- Kolab 3.1
- [http://en.wikipedia.org/wiki/List\\_of\\_free\\_software\\_web\\_applications](http://en.wikipedia.org/wiki/List_of_free_software_web_applications)



# Produkte „Ja“:

- rsync
- r2sync
- Unison
- PowerFolder
- Seafile
- ownCloud





# rsync

- Kommandozeile
  - prüft anhand Zeitstempel, welche Dateien
  - effizient bei grossen Dateien (blockweise Deltas)
  - schnell, sicher (ssh), überall verfügbar
  - unidirektional
- => perfekte Lösung für Sync A -> B



# rsync

```
rsync
  --del
  --archive
  --verbose
  --one-file-system
  --hard-links
  --bwlimit=100000
  --dry-run
  /Users/ich/
  server:/backups/ich/
```

```
rsync --del -avxH /Users/ich/ server:/backups/ich/
```



# bidirektional:

## Was ist daran so schwierig?

- Was wenn Datei gelöscht?
- Konflikt, wenn Datei auf beiden Seiten geändert

=> Datenbank über Ist-Zustand



# Was kann schief gehen?

- Scheiß encoding



# UTF-8: Alles wird gut!

- Nein: es gibt NFC und NFD

```
kfr — kfr@strukturpunkt.de — ssh — 88x34
kfr@kopenhagen:~/tmp/knf$ ls -la
total 8
drwxr-xr-x 2 kfr users 4096 Nov 10 21:52 .
drwxr-xr-x 6 kfr root 4096 Nov 10 21:44 ..
-rw-r--r-- 1 kfr users 0 Nov 10 21:52 blöd.txt
kfr@kopenhagen:~/tmp/knf$ touch blöd.txt
kfr@kopenhagen:~/tmp/knf$ ls -la
total 8
drwxr-xr-x 2 kfr users 4096 Nov 10 21:53 .
drwxr-xr-x 6 kfr root 4096 Nov 10 21:44 ..
-rw-r--r-- 1 kfr users 0 Nov 10 21:52 blöd.txt
-rw-r--r-- 1 kfr users 0 Nov 10 21:53 blöd.txt
kfr@kopenhagen:~/tmp/knf$ LANG=C ls -lai
total 8
12107779 drwxr-xr-x 2 kfr users 4096 Nov 10 21:53 .
7168003 drwxr-xr-x 6 kfr root 4096 Nov 10 21:44 ..
12107780 -rw-r--r-- 1 kfr users 0 Nov 10 21:52 blo??d.txt
12107781 -rw-r--r-- 1 kfr users 0 Nov 10 21:53 bl??d.txt
kfr@kopenhagen:~/tmp/knf$
```



# Was kann schief gehen?

- Umlaute & Sonderzeichen im Dateinamen
- Zeitstempel
- Permissions / erweiterte Dateiattribute
- aus Dokument-Datei wird ein Ordner
- Soft-Links / Hard-Links
- Synchronisation bricht ab



# Was kann schief gehen?

- Teste mit DEINEN Daten
  - Dateianzahl
  - Dateigröße
  - Dateimenge
  - Änderungsfrequenz
- Beispiel-Test: Linux-Kernelquellen inkl. Binärfiles



# r2sync

- Versuch rsync bidirektional zu erweitern
- Perl-Script ruft rsync mit `--dry-run` auf und gleicht mit Datenbank ab
- Autor ist an Umlauten verzweifelt
- [r2sync.org](http://r2sync.org)





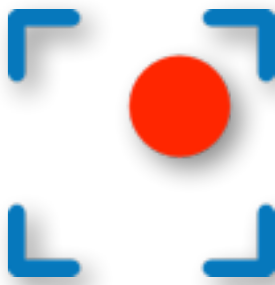
# Unison

- bi-direktionale Dateisynchronisation per Kommandozeile und GUI
- <http://www.cis.upenn.edu/~bcpierce/unison/>
- unübliche Programmiersprache: Objective CAML
- seltene Updates
- Server: SSH-Zugang und /usr/bin/unison
- Standard-Linux Paket „unison“ (Version muss passen)
  - 2.40 nicht für Debian Squeeze
- Win&OS X: <http://alan.petitepomme.net/unison/index.html>
  - Menü: Install command-line tool



# Unison

- sicher (ssh), schnell, gut bei grossen Datenmengen
- gute Konflikterkennung, manuelle Lösung
- nicht vollständig automatisierbar
  - hängt manchmal (2.32)
- `~/Library/Application\ Support/Unison/`
  - Status-Datenbank
  - Profile



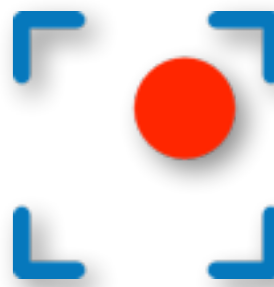
# Unison

- Beispiel ~/Library/Application Support/Unison/KNF.prf

```
root = /Users/kfr/tmp
root = ssh://kfr@ccloud.franken.de//home/kfr/unison-tmp
times = true
#silent = true # => -auto -batch

ignore = Name {geheim*}
ignore = Name {CVS,* .svn}

path = KNF1
path = KNF2
```



# Unison

Unison: Wikipedia (local <-> proliant)

Quit Restart Go Rescan Right to left Merge Left to right Skip Diff Layout

Check and/or adjust selected actions; then press Go

Path	Size	<	Action	>
▼ All Changes... (6 files)	8.5 MB		←?→	
▼	--	~	←?→	~
TheCowAtTheSteeringWheel.jpg	27 KB	+	→	○
iperf	76 KB	○	←	+
logs.tar.gz	< 1KB	○	←	+
mail-20080701.bz2	5.7 MB	○	←	+
ocaml-3.10.2.tar.gz	2.7 MB	+	→	○

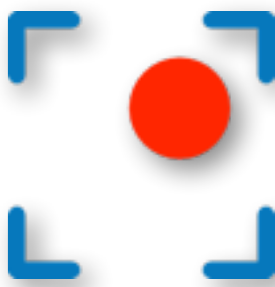
```

local      : dir props changed  modified on 2009-12-31 at 14:47:05  size 2814071  rwxr-xr-x
proliant   : dir props changed  modified on 2009-12-31 at 14:51:41  size 6086828  rwxr-x---
    
```



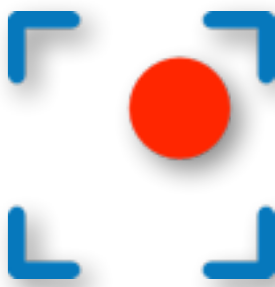
# Unison

- Automation:
  - `unison -ui text -auto -batch KNF`
  - Exit 0: OK, alles übertragen
  - Exit 1: Konflikte, sonst alles übertragen
  - Logdatei: `~/unison.log`
- Grafisch: `unison -ui graphic KNF`



# Unison

- das ist schon ein bisschen Cloud
- Test mit >10GB, >10.000 Files OK
- aber:
  - Konflikte beachten
  - mehrere Clients „Handarbeit“
  - Gruppenfreigaben noch mehr „Handarbeit“

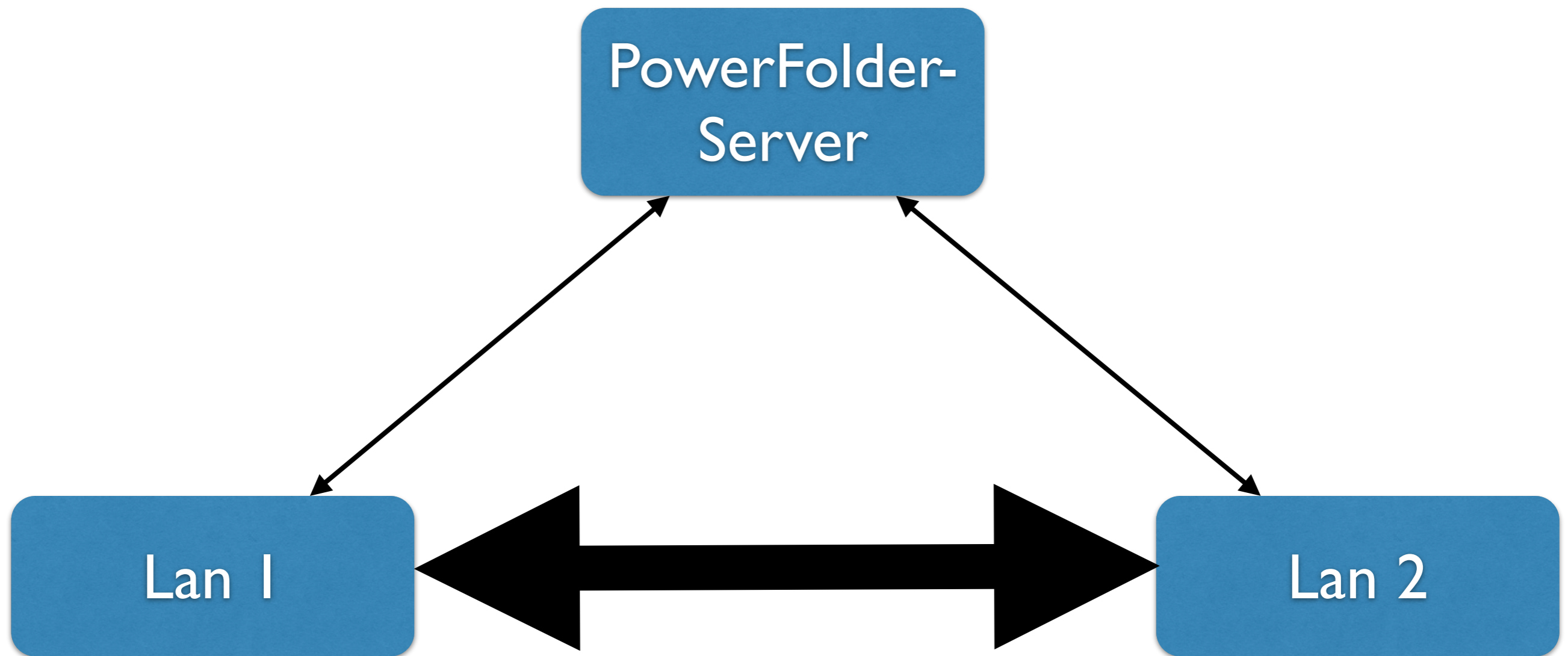


# PowerFolder

- kommerziell, Open-Source-Version
- Java-Server und Java-Client
- unübersichtliche Lizenz, hohe Kosten
- professionell (Stabilität, Features)
- synct Dateien sofort (inotify, fsevents?)



# Multi-source Download

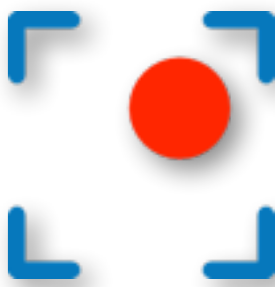






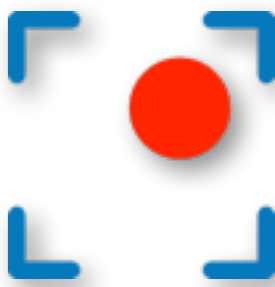
# PowerFolder

- Konflikterkennung:
  - letzter gewinnt
  - Warnungs-Anzeige im Client (nicht Server)
  - alte Dateien im Archiv
  - funktioniert im Test nicht wie beschrieben :-)



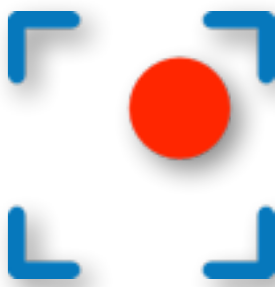
# PowerFolder

- SSL: [https://wiki.powerfolder.com/display/PFS/Using+Apache+with+mod\\_proxy+and+mod\\_ssl](https://wiki.powerfolder.com/display/PFS/Using+Apache+with+mod_proxy+and+mod_ssl)
- keine Sonderzeichen im Dateinamen: % :
- Client: Memory-Leak bei grossen Uploads?
- auch WebDAV-Server (per Option)
- WebGUI nicht so schön, die meisten Dateien nur Download, keine Anzeige (nicht mal Bilder).



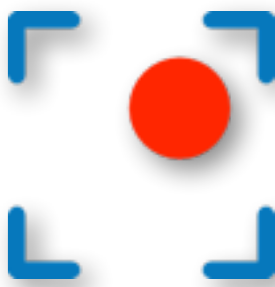
# PowerFolder

- Fazit:
  - Test mit 13GB, >30.000 Files OK
  - private Gruppen-Cloud
  - weniger Web-Lösung
  - „professionell“



# Seafiler

- Open Source
- China
- Python Server, sqlite3 (und andere DBs)
- SSL schwierig
- basiert auf Git



# Seafile

- wichtigste Unterschiede zu git
  - Automatische Synchronisation
  - Umgang mit grösseren / binären Dateien
  - Konfliktbehandlung
    - `x.txt`
    - `x (klaus.franken@strukturpunkt.de 2013-11-23-11-27-55).txt`



# Seafiler

- arbeitet immer transaktionsorientiert
- Vorteil oder Nachteil?
- andere Bibliotheken werden aber gesyncnt



# Seafiler

- Plus:
  - Wiki, Markdown-Syntax
  - client-seitige Verschlüsselung
  - Web-GUI: Text-Edit, Anzeige vieler Dateitypen
  - Freigaben an Mitglieder/Gruppen oder als Link
  - Test mit 8 GB, > 20.000 Dateien OK



# Seafiler

- Negativ:
  - Dateinamen auf Server nicht im Klartext
    - aber: Linux-Client als CLI
  - kein WebDAV, Kalender, Adressbuch
  - Zeitstempel der Dateien gehen verloren
    - es wird diskutiert ...





# Seafiler

- Fazit:
  - sehr viel Cloud - für Dateien
  - Showstopper: Zeitstempel