

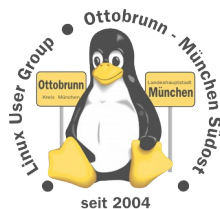
# Backup: rsync, SSH, LUKS im Team

„Niemand hat die Absicht eure Daten zu klauen“.  
von löschen war nicht die Rede ...

- **Richard Albrecht, Jahrgang 1949** Webseite: <http://rleofield.de/>
  - Physiker / Uni Halle-Wittenberg
  - 1988 - 2000 am MPI für Biochemie Martinsried, Bildverarbeitung, C/C++ Entwicklung
  - bis 2011: Middleware, Datenbanken, .NET, Webanwendungen
  - bis 2014: Software für CCD Kameras bei SVS-Vistek in Seefeld
  - jetzt: Rentner in Görlitz mit Linux
- **Warum der Vortrag?**
  - alle Datendesaster, die ich erlebt habe, waren menschliche Fehler
    - Kommunikation, mangelnde Tests, zu komplex
    - ungeeignete Mittel: RAID, NAS
  - Trojaner **locky** hat Erfolg, (noch in Windows)
    - fehlende Isolation, fehlende Historie
- **Diskussion der Möglichkeiten**
  - einfach, preiswert, sicher
  - mit Linux

# Linux User Group Ottobrunn - München SüdOst - LOMSO

[Webseite LOMSO](#)



Linuxtage Graz 2016  
Richard Albrecht, LUG-Ottobrunn  
Linux Stammtisch Görlitz

# Linux-Stammtisch in Görlitz



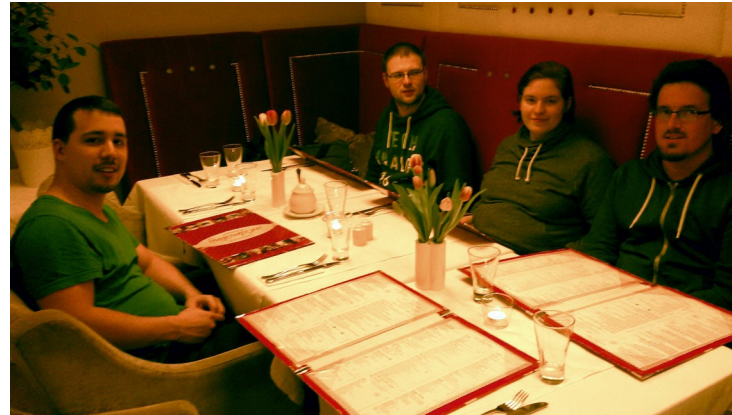
Biergarten auf dem Kraftwerk



Altstadtbrücke



3 Königsumzug von Polen nach DE



unser Stammtisch  
(diesmal in Polen)

[Linux Stammtisch Görlitz bei Google+](#)



Hauseingang



Jugendstil-Bahnhof

Linuxtage Graz 2016  
Richard Albrecht, LUG-Ottobrunn  
Linux Stammtisch Görlitz



# Backup Umgebungen

- **Hilfe für Linux Umsteiger (die ich betreue)**
  - Benutzer kann es nicht ohne Hilfe einrichten und pflegen
  - muss vom Helfenden gemanagt werden
  - mit eigenen Ressourcen
  - PC ist hinter Router
  - Datenmenge nicht hoch
    - einige GB
  - Datenschutz beachten, genaue Absprachen
- **kleine Firmen**
  - oft überfordert
  - automatisch, wartungsfrei
  - Brandschutz
  - Daten bleiben lokal
  - Administration per SSH (Datenschutz beachten)
  - LUKS
- **größere Firmen**
  - haben eine eigene IT
  - meist keine Probleme (?)
  - nicht in diesem Vortrag



# Backup Umgebungen

- Hilfe für Linux Umsteiger (die ich betreue)
  - Linux Wissen kann nicht vorausgesetzt werden
  - wird sich auch nicht angeeignet
  - Beruf, Familie
  - The Art of Unix Usability (Eric Raymond)
    - Allow your users the luxury of ignorance

Melanija (mitte) in Kroatien



man hat DSL, aber keinen Router, wo ist der denn ...  
kein Zugriff auf den Router (kein Portforwarding)  
Passwort vergessen, hat der Nachbar ...

Datenmengen 20-30 GB pro PC



Christoph in Neuried

Lore in Pasing



# Voraussetzungen für die folgenden Abschnitte

- **Installieren von Programmen**
  - Synaptic, apt-get
  - Hilfesystem (man, info, Wikis)
- **Terminal**
  - öffnen, einfache Kommandos absenden
  - Arbeiten als root, sudo -s
  - Editieren von Konfigurationen
  - einfache Skripte
- **Netzwerk**
  - Internetadressen, Namensauflösung,
  - Dienste, Ports (in /etc/services)
  - Router, Modem
  - lokales Netz

# Überlegungen

- Planung
  - Was will man erreichen?
- Software
  - einfach, komplex, teuer, preiswert, ...
- Hardware
  - NAS, RAID, Mini Server, externe HD, ...
- Verfahren
  - copy
  - inkrementell
  - Historie
- Sicherheit, phys.
  - Zugriffsrechte, Lagerung, Brandschutz

# Planung (gemeinsam)

- Warum ein Backup
  - Hardwaredefekte
  - Datenverluste
  - Brand
  - Archiv
- Linux Filesystem, FHS
  - /etc, /usr/local/bin, /home,
- Dateigrößen
  - Konsolidieren
    - wichtig, weniger wichtig
    - statische Daten von dynamischen Daten trennen
    - Links im Filesystem einsetzen
- Mengen abschätzen
  - ein Zweig im Filesystem
    - kleiner als 50% der Backup HD (< 1 TB)



# Planung2

- **Was**
  - Daten oder System
- **Wie oft**
  - stündlich, täglich, wöchentlich
  - Granularität der Zeiten
- **Historie**
  - Wie lange?
  - Granularität der Archivierung
- **Lagerung**
  - zu Hause
  - Brandabschnitt, Lagerung außer Haus
- **Kosten**
  - Wert der Daten
- **Wiederherstellung**
  - Kenntnisse (des Betreuers)
  - Zeitfenster (Stunden?, Tag(e), Woche)

# Planung, Technik

- einfache HD
  - preiswert
  - austauschbar
  - nicht sicher (s.u. bei copy/paste)
- NAS
  - wie HD
  - teurer
  - Protokoll dazwischen, meist SAMBA
- Filesysteme im Server
  - ext4, zfs
  - NTFS?
  - Standard Linux ist ok
- Zuverlässigkeit
  - RAM (memtest)
  - Server-PC, Board
  - Netzteil

# Varianten

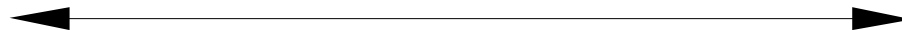
copy/paste

rsync

rsnapshot

rsync/rsnapshot  
(mit staging area)

lokal

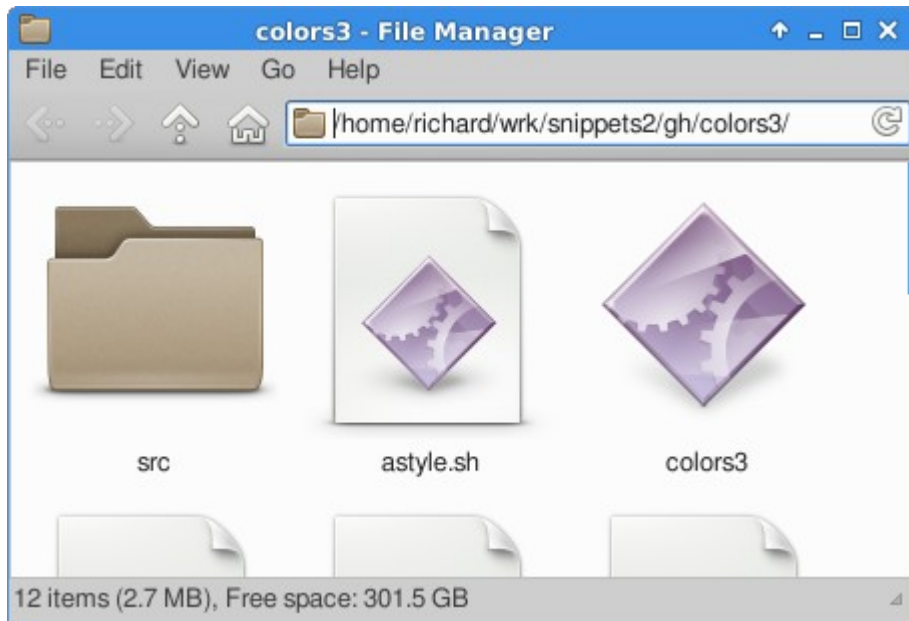


remote

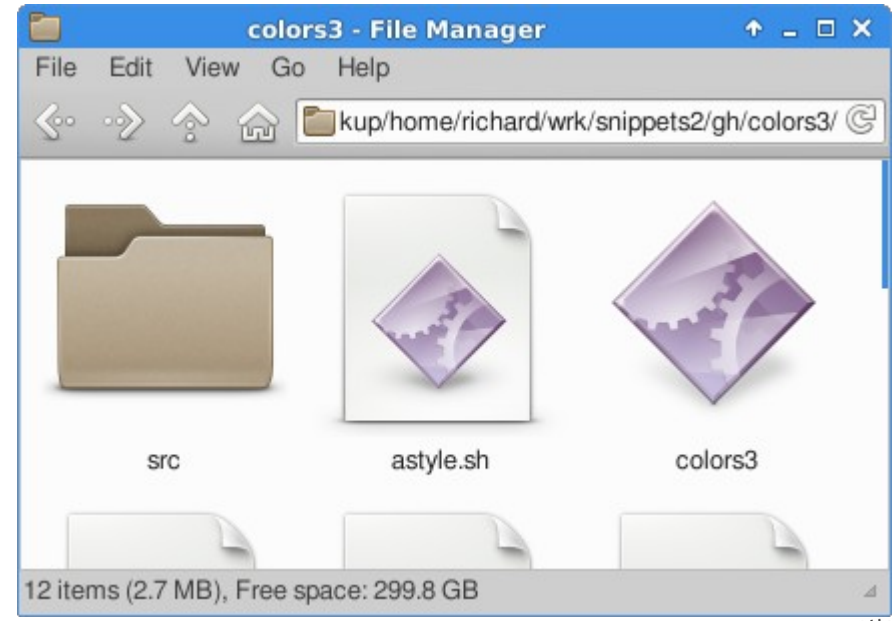
# einfach, mit cp

- **copy/paste**
  - einfach
  - mit GUI
  - normalerweise keine Skripte

Original



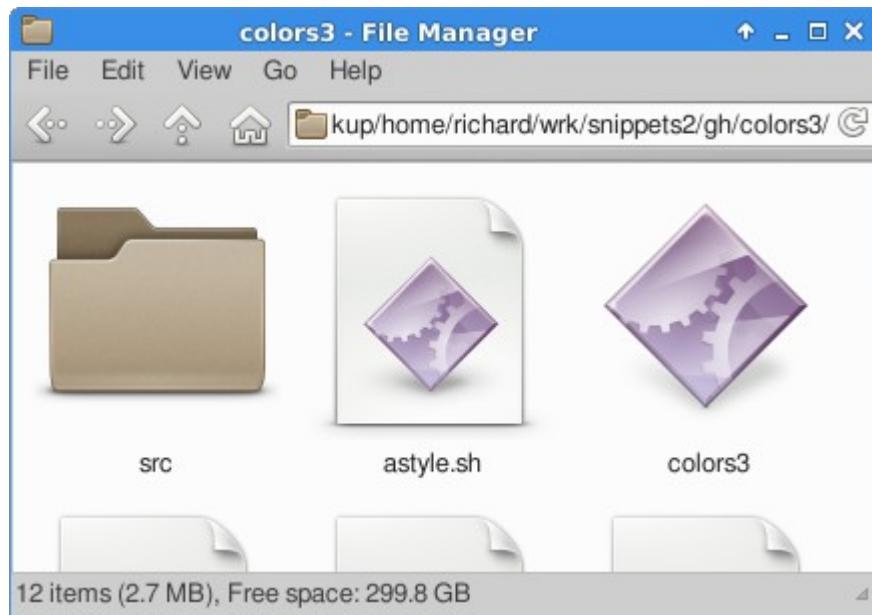
Backup



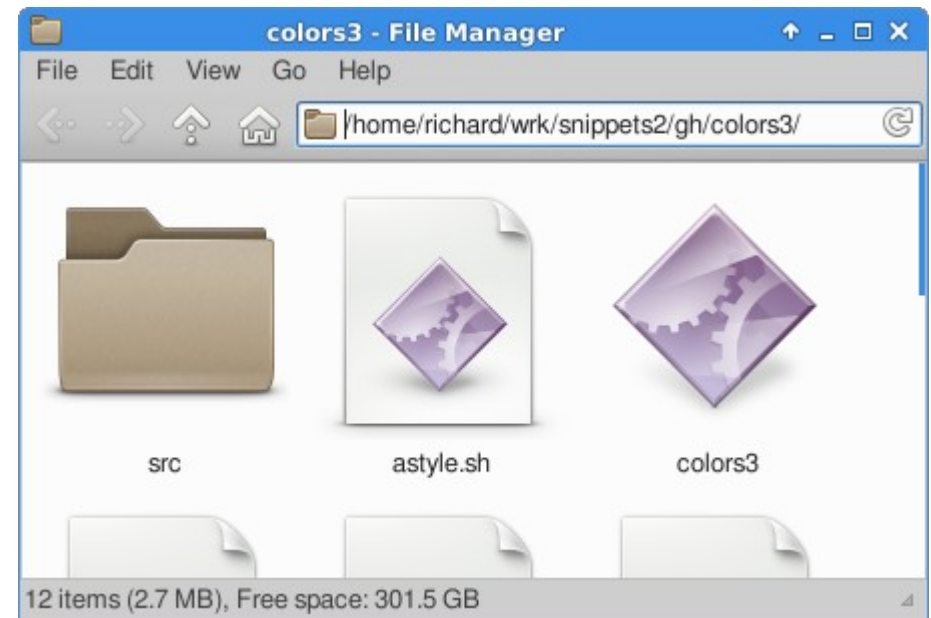
# Warum das keine gute Idee ist

- **copy/paste**
  - keine Sicherheit
  - keine Verifikation
  - vom Host erreichbar
  - Malware kann Backup löschen (locky!)
  - nur mit Tricks sicherer (RO, extern, ...)
  - „rm -rf /“ kann jedem als root passieren (**Bitte nicht ausprobieren**)

Original?



Backup?



Vertauschung Backup - Original ...



# rsync, das bessere cp



- Ersatz für copy/paste
  - mehr Prüfungen
  - inkrementell
  - SSH
  - viele Optionen
  - Entkopplung
- Optionen
  - man pages
    - sehr viele, unübersichtlich
- meine Auswahl, ohne SSH
  - \$ rsync **-avSAXH** source/ target/
    - alle Attribute, Rechte,
    - Hardlinks, Sparse Files
- Filesysteme
  - nur Linux, andere vermeiden, wie VFAT, NTFS
  - ext3, ext4
  - SAMBA?
    - kein Sparse, nicht alle Attribute

# rsync



## • rsync remote

- nicht komplexer als vorher
  - `$ rsync -avSAXH user@client:/source/ target/`
- SSH kommt hinzu (Option `-e`, kann weggelassen werden)
- in beide Richtungen möglich

## • Entkopplung

- **Server holt die Daten vom Client**
- **versehentliches Löschen nicht möglich**
- **Client hat keine Möglichkeit, die Daten zum manipulieren**
  - „`rm -rf /`“ löscht nur den Client, nie das Backup

## • SSH

- nur über Schlüssel
- Client hat **keinen** Zugriff auf Server
- Server hat Zugriff auf Client, je nach Rechten
- Server hat u.U. **root** Zugriff !
- Server im Netz **absichern**
- **headless**, ohne GUI, keine normalen Aufgaben

# Rsync, Probleme

## • Path Syntax

- `rsync -avSAXH source target/`
  - legt source in target an
- `rsync -avSAXH source/ target/`
  - source wird nicht in target angelegt (Unterschied ist der Slash nach Source, ohne Leerzeichen)
- Option **-n** (Dry Run)
  - Test, was rsync macht, ohne Ausführung

## • Überschreiben von Files

- keine Warnung
- kann man abschalten, ist nicht sinnvoll
- rsync soll automatisieren
- **testen, testen, testen**, 3 x hinschauen



## • Prüfungen vor copy

- Größe und Datum
  - kein Inhalt
  - kann erzwungen werden, dauert dann aber sehr lange
    - (-I, --ignore-times — don't skip files that match size and time)
    - (-c, --checksum — skip based on checksum, not mod-time & size)
- bei RAM Fehlern keine Kontrolle der Integrität
- eigene Prüfungen vorsehen
  - 2 x Backup, identisch, MD5, diff u.a.

# Backupsoftware?

- eine Schicht mehr
  - weniger Kontrolle
  - auf Hersteller angewiesen
  - Updates?
  - Datenformat?
  - Historie
  - Preis
  - TOC
  - Wartung
  - Verschlüsselung?
  - SSH?
  - Remote?
  - Entkopplung?
- wir haben Linux, wir brauchen das nicht
  - rsync, rsnapshot, diff, SSH, LUKS

# NAS?

- gleiche Probleme wie externe HD, aber
  - Linux?
  - kein Linux
    - kein Filesystem Standard
    - kastriert
  - Abhängigkeit vom Hersteller
  - alle Nachteile der externen HD
  - keine Trennung, kein Server
  - SSH, rsync
    - Versionen? umständlich
  - teuer
  - leistungsschwach
  - viele Linux Programme fehlen
  - eigenes Linux nicht installierbar.
- als Fileserver ok
- als Backup Server nicht



# Lösung: Mini Server

- preiswerter kleiner Server
  - Linux unserer Wahl
  - Open Source
  - austauschbar
  - SSH, rsync
    - Version identisch mit Client
  - Updates
  - LUKS
  - headless
    - ohne GUI
    - stabiler
  - Wartung nur über SSH
- es reicht ein alter PC
  - Stromverbrauch?
  - USB?
  - SATA?
  - kann auch eine VM sein

**Mini Server (Vorschlag)**  
2x2 TB HD, SSD, 8GB RAM,  
10-15 Watt, ohne Lüfter  
mit Ubuntu Server  
unter 450 Euro



# SSH, Installation

- **SSH installieren (auf allen beteiligten PCs)**
  - # apt-get install **ssh autossh**
    - **Server absichern**
    - **/etc/ssh/sshd\_config editieren**
    - Passwort-Login für alle Benutzer sperren
  - Schlüsselpaar erzeugen und sichern (\$ ssh-keygen)
    - für jeden Benutzer auf dem Server
  - öffentliche Schlüssel des Servers auf die Clients verteilen
    - privater Schlüssel verbleibt auf dem Server
    - öffentlicher Schlüssel kommt auf den Client (~/.ssh/authorized\_keys2)
- **eigenen Router am Server freischalten**
  - Port 22 (bzw. der für SSH gewählte Port) muss zum Server-PC im eigenen Netz weitergeleitet werden
  - Firewall im Router abschalten, bzw. den SSH Port freischalten
- **Remote Router freischalten?**
  - nicht nötig
  - wegen SSH -R



# SSH, remote Client, betreute Rechner

- Rechner meldet sich beim Start am Server an
  - für jeden remote PC einen Account im Server
  - für jeden remote PC einen Port im Server
  - Anmeldung an einem Login Account
  - 'rsync' auf User im spezifischen Account
  - mit **autossh** (überlebt reboot, Client oder Server)
- in /etc/rc.local des Clients
  - SSH -R zum Server, silent
  - Verbindung von **localhost** zu **localhost**



In rc.local des Client:

```
su -s /bin/sh user -c '/usr/bin/autossh -p 22 -q -f -N -R 14520:localhost:22 login@yy.xxxxxx.de'
```

im Server Account für 'rsync':

Fernwartung: **ssh -p 14520 -X -C user@localhost**

Backup: **rsync -av -e 'ssh -p 14520' user@localhost:/home/user/ /mnt/user/**

- rsync Client->Server kann jederzeit starten, vom Server aus
  - keine Portweiterleitung im Client Router
  - Ort des Client braucht nicht bekannt zu sein
- rsync kann unterbrochen werden
  - Shutdown des Client
  - Skript auf dem Server, Test mit Ping, ob der Client wieder aktiv ist
  - rsync neu starten

# LUKS

- **die Endstation unseres Backups**
  - Zugriffssicherheit
  - mit Keyfile, für rsync
  - mit Passwort für eigenen Zugriff
- **Brandschutz**
  - mit LUKS kann man eine HD an eine unsicheren Stelle auslagern
    - beim Nachbarn, im Schließfach
- **LUKS ist immer letzte Stufe**
  - einfacher Austausch der HD, kein Schreddern der Daten
- **man spart**
  - Tresor, bei Diebstahl ist nur der Neuwert der HD weg
- **Plan**
  - Austausch der Platten mit den externen Sicherungen



**LUKS**  
Linux Unified Key Setup

LUKS bei LOMSO in Ottobrunn: <https://www.lug-ottobrunn.de/wiki/LUKS>

# Alles zusammen

- **rsync**

- remote Daten abholen,
  - erstes rsync braucht u.U. lange (einige Wochen, langsame DSL Verbindung)
- wegen **rsync** und **autossh** stabil

- **SSH -R**

- Remote Zugriff per localhost

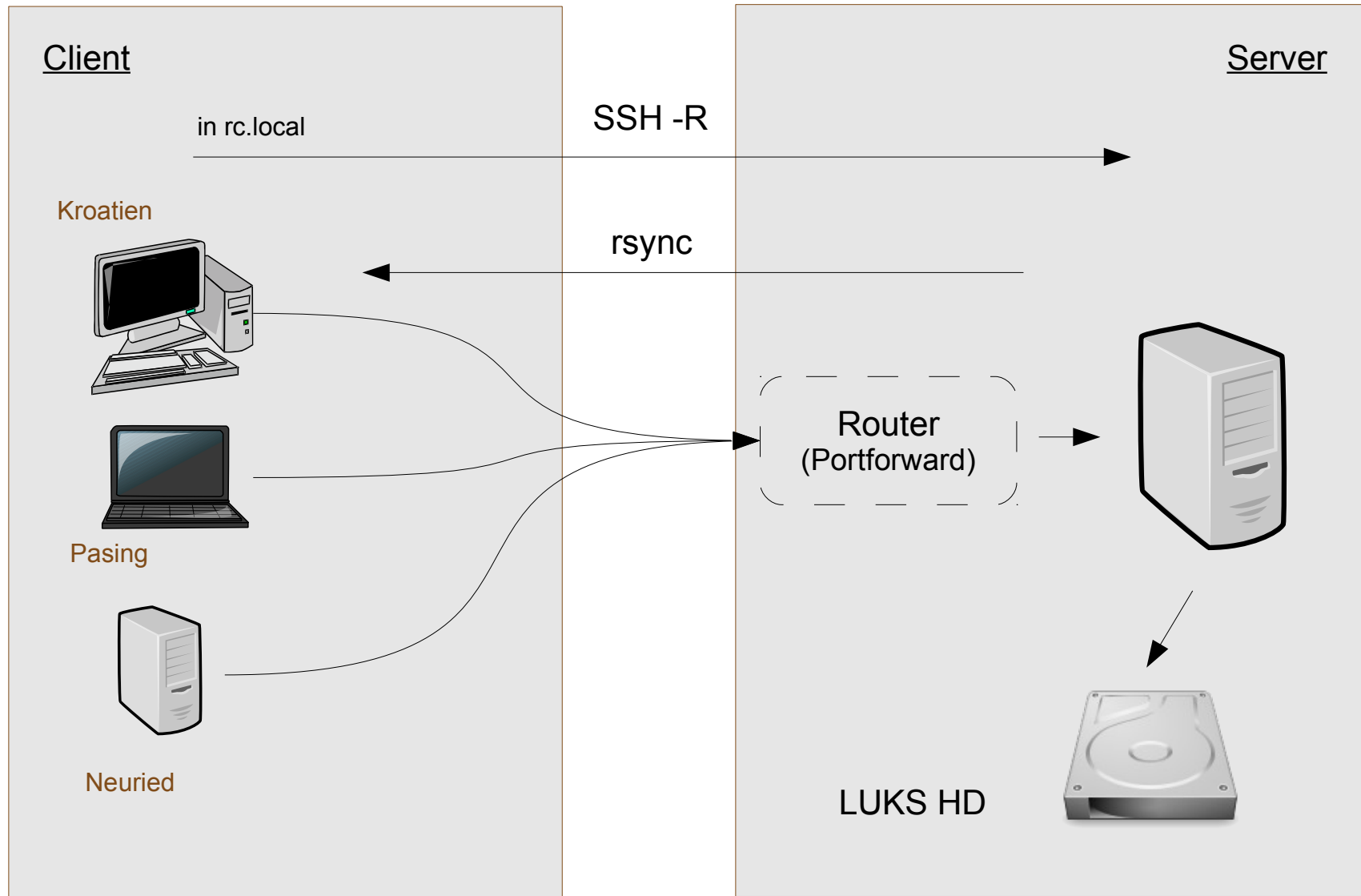
- **LUKS**

- als finale HD im Server
- austauschbar
- 2x, besser 3x
- vergleichbar





so schaut das aus



# rsnapshot

- rsync der Oberklasse ;-)
  - Historie
  - über cron gesteuert
  - flexibel
- Einstellungen
  - 3 Dinge, die aufeinander abgestimmt sind
  - rsnapshot.conf, (**retain**, logfile, backup Zeilen)
  - source folder / target folder
  - cronjobs, mit **retain** Werten aufrufen

30 min



# Rsnapshot.conf

- Retain values

- Intervalle für Backup
  - hourly, daily, weekly, ...
- müssen aufeinander in cron abgestimmt sein
- Namen beliebig, aber eindeutig
- keine Leerzeichen im File

- Cron

- rsnapshot aufrufen mit genau den gleichen Namen für die Intervalle
- d.h. hourly, weekly, ..
- oder per Skript in Kombination mit cron

- Test

- rsnapshot -c rsnapshot.conf configtest

# Rsnapshot.conf

## common retain values

retain	hourly	4
retain	daily	7
retain	weekly	4
retain	monthly	3

## angelegte Folder:

daily.0  
daily.1  
daily.2  
daily.3  
daily.4  
daily.5  
daily.6  
hourly.0  
hourly.1  
hourly.2  
hourly.3  
monthly.0  
monthly.1  
weekly.0  
weekly.1  
weekly.2  
weekly.3

## strange retain values ;-)

retain	maus	4	hund.0
retain	katze	3	hund.1
retain	hund	3	hund.2
retain	wolf	5	katze.0
			katze.1
			katze.2
			maus.0
			maus.1
			maus.2
			maus.3
			wolf.0
			wolf.1

## cron

/etc/cron.d/rsnapshot			
3	*/6	* * *	root /home/rleo/bin/rsnapshot/hourly
3	1	1 1 *	root /home/rleo/bin/rsnapshot/yearly
33	1	1 * *	root /home/rleo/bin/rsnapshot/monthly
11	2	* * 1	root /home/rleo/bin/rsnapshot/weekly
11	3	* * *	root /home/rleo/bin/rsnapshot/daily

# Rsnapshot.conf

## Backup Values

### mit SSH

backup	root@nc.xxxxx.de:/bin	target/	ssh_args=-p11022 -i /root/.ssh/id_rsa_nc
backup	root@nc.xxxxx.de:/root	target/	ssh_args=-p11022 -i /root/.ssh/id_rsa_nc
backup	root@nc.xxxxx.de:/etc	target/	ssh_args=-p11022 -i /root/.ssh/id_rsa_nc
backup	root@nc.xxxxx.de:/home	target/	ssh_args=-p11022 -i /root/.ssh/id_rsa_nc
backup	root@nc.xxxxx.de:/opt	target/	ssh_args=-p11022 -i /root/.ssh/id_rsa_nc
backup	root@nc.xxxxx.de:/usr/local/bin	target/	ssh_args=-p11022 -i /root/.ssh/id_rsa_nc

### ohne SSH

backup	/media/red/backup2/test_bilder/	test_bilder/
--------	---------------------------------	--------------

### Rsync args

rsync_short_args	-aSAXHv
rsync_long_args	--delete --numeric-ids --relative --delete-excluded



# Alles zusammen mit rsnapshot

- **rsync**
  - remote Daten abholen,
    - erstes rsync braucht u.U. lange (einige Wochen, langsame DSL Verbindung)
- **SSH -R**
  - remote Zugriff per **localhost**
- **rsnapshot**
  - ungeeignet für Rechner, die nicht immer laufen
  - einen **Arbeitsbereich**, Staging-Area, auf dem Server anlegen, der mit **rsync** gefüllt wird
  - **rsnapshot** holt die Daten dort ab
  - ca. 30GB pro remote PC
- **LUKS**
  - als finale HD im Server
  - austauschbar
  - **2x**, um zu vergleichen

# Skripte

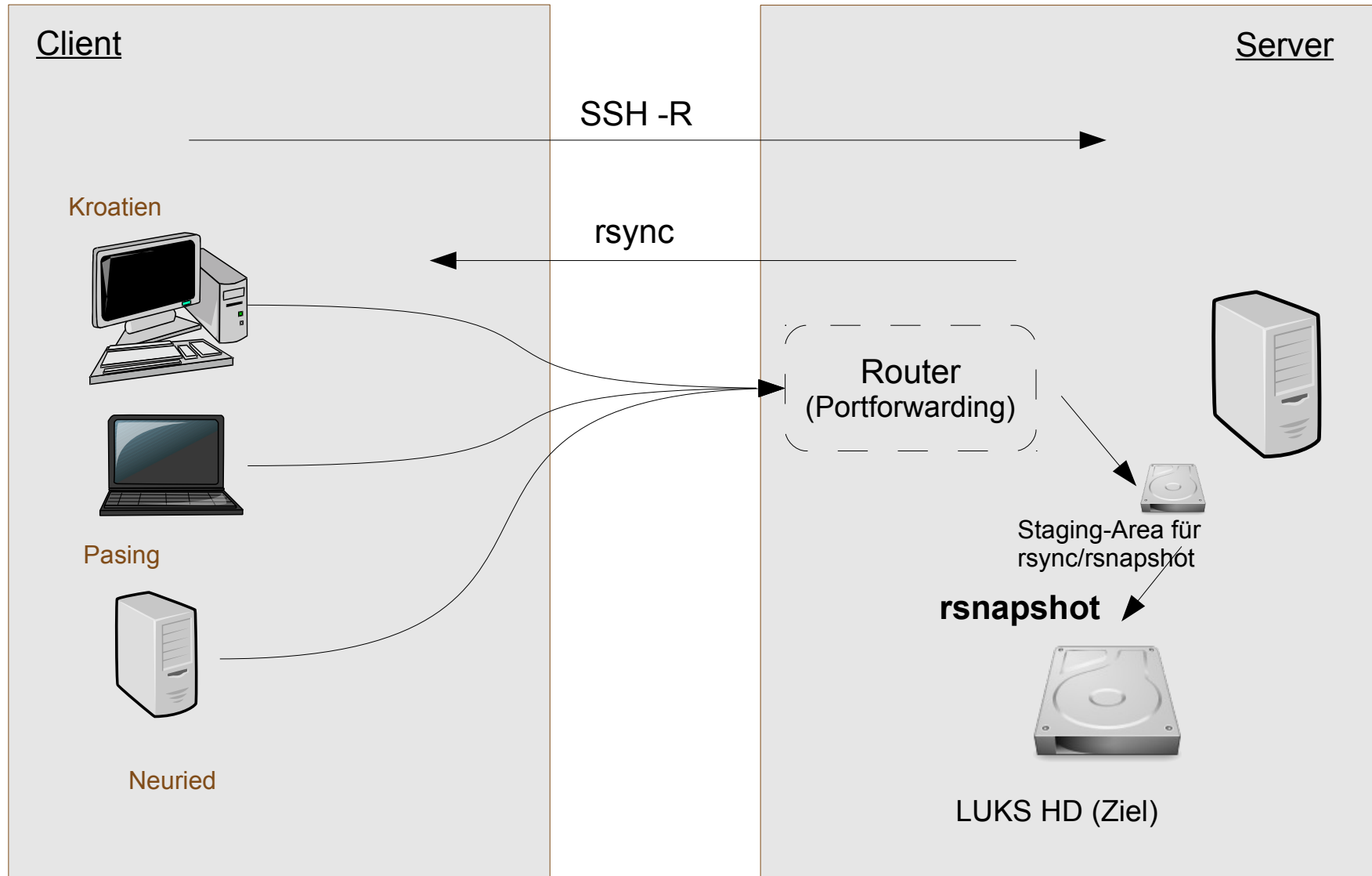
- **mount externe USB**
  - Cronjob, daily, hourly
  - HD ist leicht austauschbar
- **remote rsync**
  - alle 15 min poll, bei Erfolg, wait 3 h
- **rsnapshot**
  - mehr logging
  - Erfolgskontrolle
  - Kopie auf 2. HD
- **Schutz gegen bit rot**
  - Bitfehler ('rotten bits') sind heute nicht mehr selten, weil die Platten sehr groß sind
  - **diff -r --no-dereference rs/ /mnt/dluks/rs2/**
  - MD5 Vergleich ist besser
    - unterbrechbar
    - bei Fehler, HD tauschen, **rs** Folder neu anlegen, ok

Demo bei Github:

[https://github.com/rleofield/rsync\\_scripts](https://github.com/rleofield/rsync_scripts)

- Bit Fehlerrate  $< 1 \text{ pro } 10^{14}$
- HD Size: 4 Terabyte =  $32 * 10^{12}$  Bits
- auf Austauschbarkeit und kleine Preise setzen.

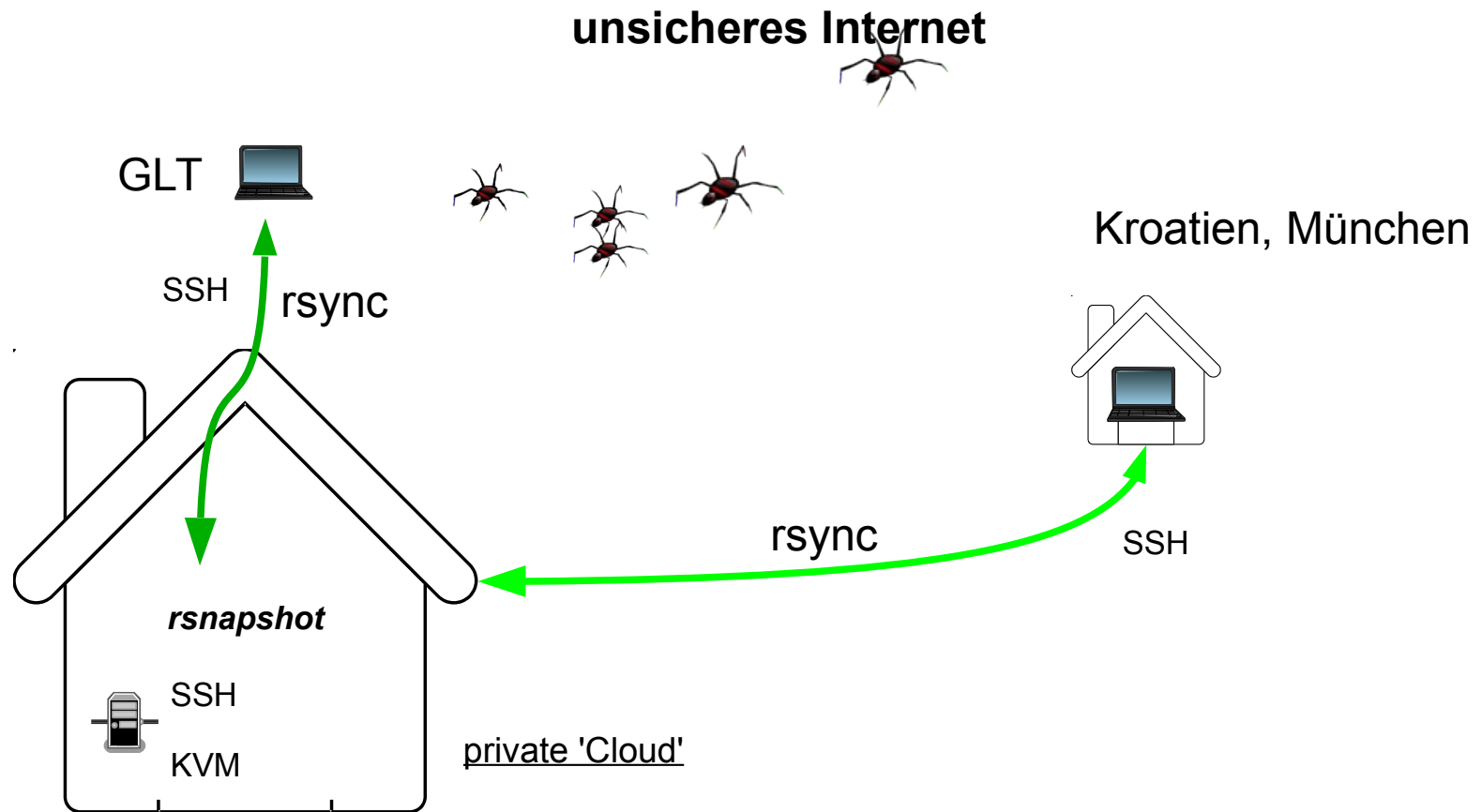
so schaut das aus, mit 'Staging-Area' für rsnapshot



# Demos

## • Demonstration

- nach Hause zum Server
- remote rsync aus Kroatien
- rsnapshot Konfiguration



# Ende des Vortrages

## • Take Home Message

- ein gutes Backup mit Linux Bordmitteln ist nicht schwer
- räumliche Trennung und Sicherheit ist mit **rsync, rsnapshot, SSH und LUKS** erreichbar
- mit SSH -R kann man jeden (konfigurierten) Rechner remote erreichen



*Vielen Dank für Eure Aufmerksamkeit.*

*Noch einen schönen Grazer Linuxtag 2016  
wünscht Euch*

*Richard Albrecht  
LUG-Ottobrunn  
Linux-Stammtisch-Görlitz*



richard albrecht  
physiker