

Datensicherung über das Internet mit BorgBackup

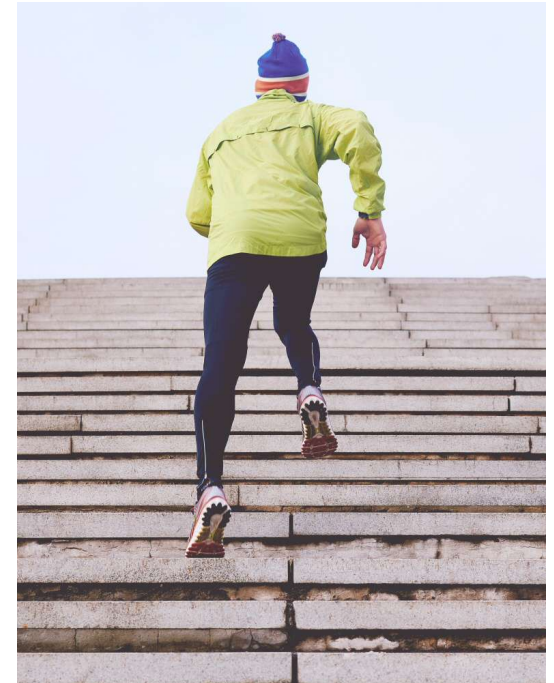


Herbert Oppmann

KNF Kongress, 24.11.2019

<http://www.franken.de/kongress>

- Persönliches fällt immer mehr digital (und nur digital) an ⇒ wech ist wech
- Backup wird daher immer wichtiger (siehe Beitrag von Max Riegel)
- Es gibt jetzt storage.franken.de (siehe Beitrag von Martin Bokämper)
- BorgBackup ist eine mögliche Lösung



Was euch erwartet



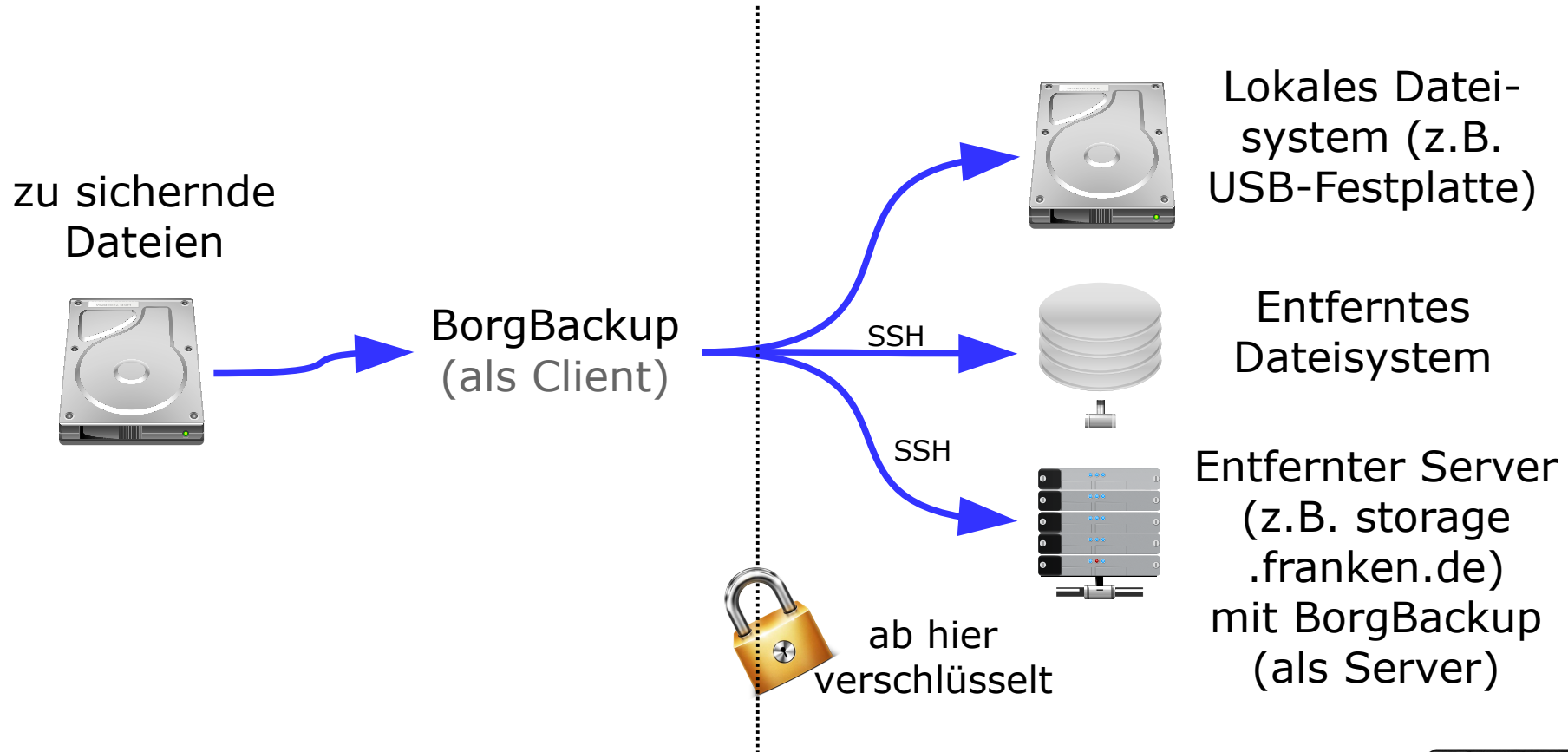
➔ Wie funktioniert BorgBackup?

- Was sind die Ziele?
- Welche Komponenten sind beteiligt und wie wirken sie zusammen?
- Wie und in welchem Maße werden die Ziele erreicht?
- Wie wende ich BorgBackup an?
 - Meine Konstellation
 - Installation
 - Betrieb

- **Vertraulichkeit der Daten**
⇒ Verschlüsselung
- **Geringer Platzbedarf für die archivierten Daten**
⇒ Deduplizierung und Datenkompression
- **Robustheit**
⇒ Transaktionsmodell, Fehlererkennung
- **Flexibilität**
⇒ Quelle und Ziel, Einstellmöglichkeiten



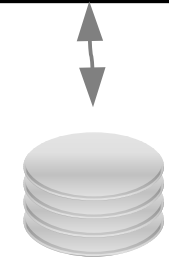
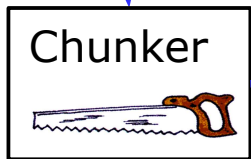
Flexibilität: Backup-Quelle und Ziel



Ein Blick unter die Haube



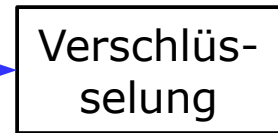
zu sichernde Dateien,
Verzeichnisse und
Meta-Informationen



Lokaler
Cache



Chunks:
0 - 512 KiB → direkt
Sonst Stücke von
512 KiB bis 8 MiB



BorgBackup
als Server



Segments
~ 500 MiB



Ziel-
Dateisystem

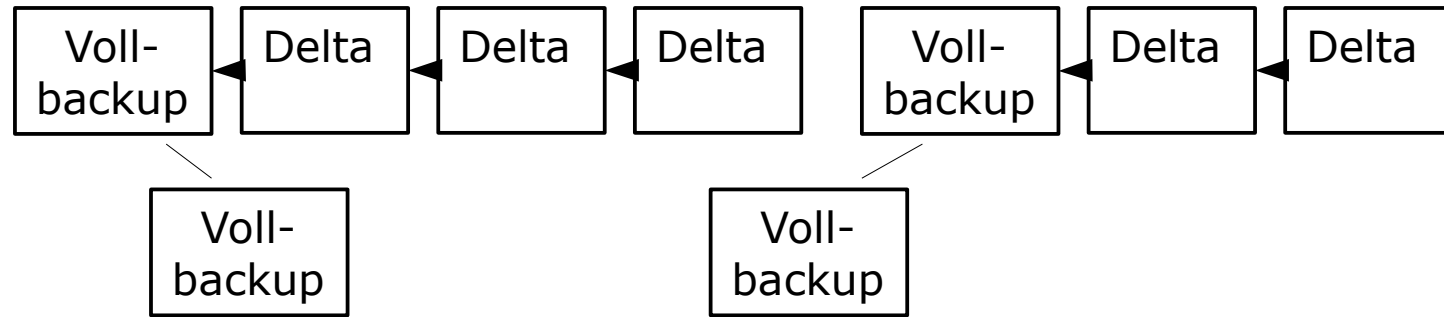
- Hilft der Deduplizierung:
 - Haben sich nur Teile einer Datei geändert, muss nicht die ganze Datei gesichert werden
- Verwischt die tatsächlichen Dateilängen
 - "buzhash" Algorithmus:
Die (dem Angreifer unbekannt) Daten bestimmen den Punkt, wo geschnitten wird! Zusätzlich startet man mit einem für das Repository zufällig ermittelten Seed-Wert.



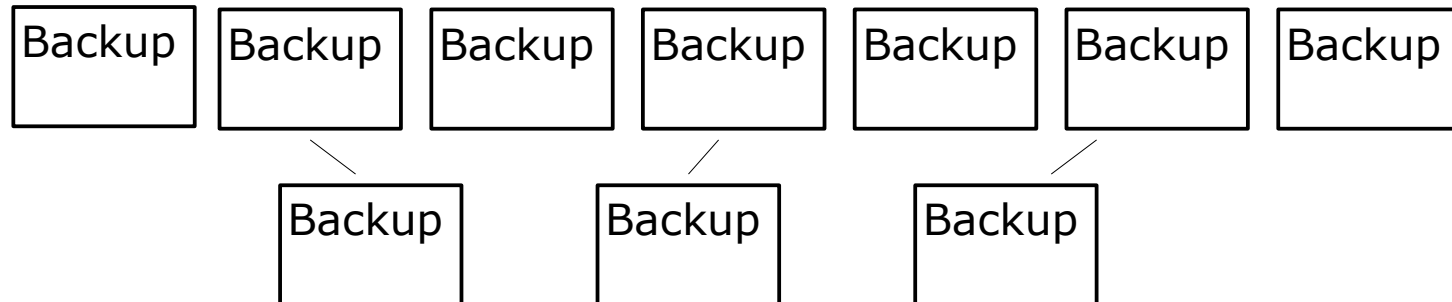
- Zu jedem Chunk wird ein kryptografischer Hash berechnet; dies ist seine ID
- Gibt es im Repository bereits einen Chunk mit so einer ID, dann wird der Chunk nicht erneut komprimiert, verschlüsselt und in das Repository geschrieben
- Es ist unerheblich, wo der Chunk herkam

Wirkung der Deduplizierung

Bekannte
Backupprogramme:



BorgBackup:



- Verschiedene (übliche) Verfahren implementiert
- Verwendetes Verfahren wird in jedem Chunk vermerkt



- Vorgegeben:
 - AES-256 in counter mode: der Initialisierungsvektor ist eine Zahl (Nonce), die für jeden Verschlüsselungsvorgang hochgezählt wird
 - Authentifizierung über HMAC-SHA256 in keyed mode (gleiches Verfahren wie bei Deduplizierung, aber anderer Key)
- Aber auch andere möglich. Welches Verfahren verwendet wird, wird am Chunk vermerkt.

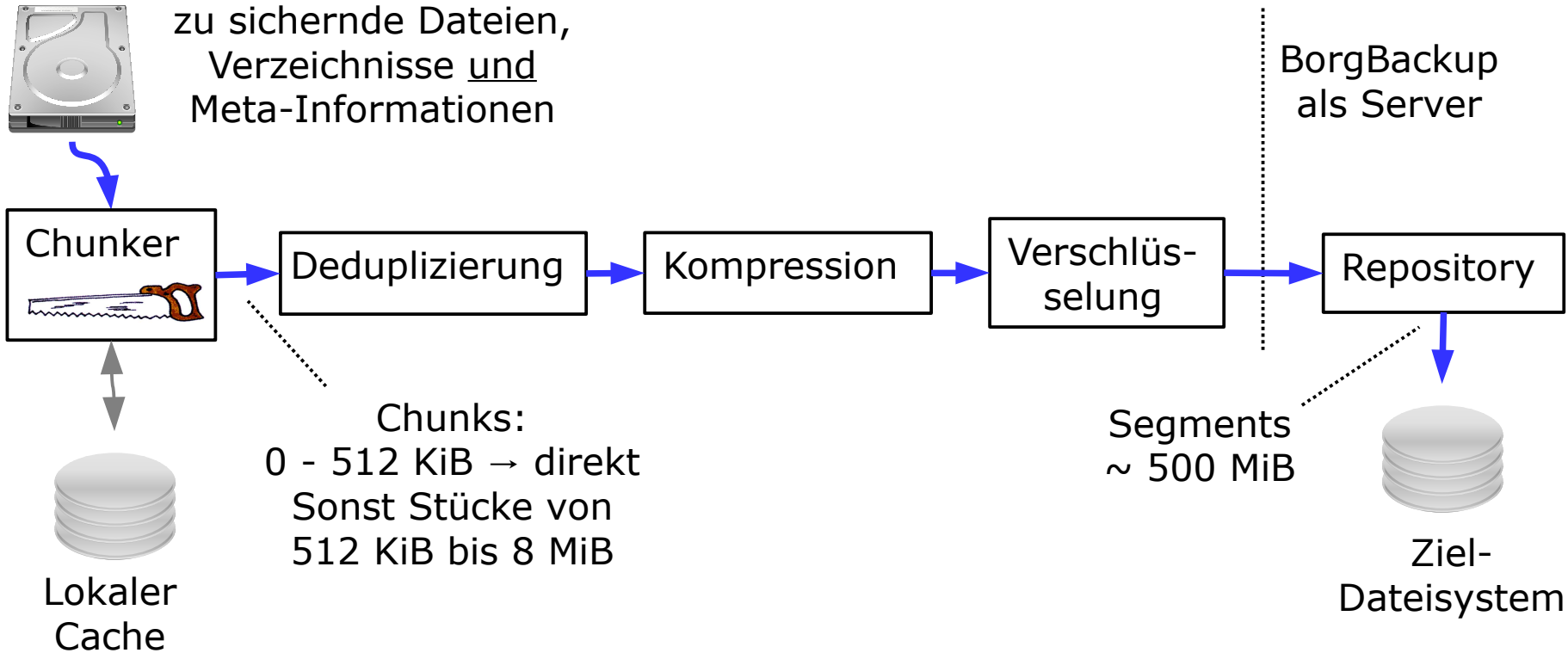


- Datenbank für Chunks mit den Befehlen PUT, DELETE, COMMIT
- Erst werden alle neuen Chunks und alle Löschaufträge übertragen und je nach Platz an vorhandene oder neue Segmente angehängt.



- Am Ende folgt COMMIT, der alle Aufträge umsetzt. Dabei müssen nur wenige Byte geändert werden. Dadurch stark reduzierte Gefahr einer Korruption.
- Ab und zu wird "aufgeräumt", d.h. nicht mehr referenzierte Chunks tatsächlich gelöscht
- Checkpoints werden nach Vorgabe alle 30 Minuten erstellt

Ein Blick unter die Haube



Was euch erwartet

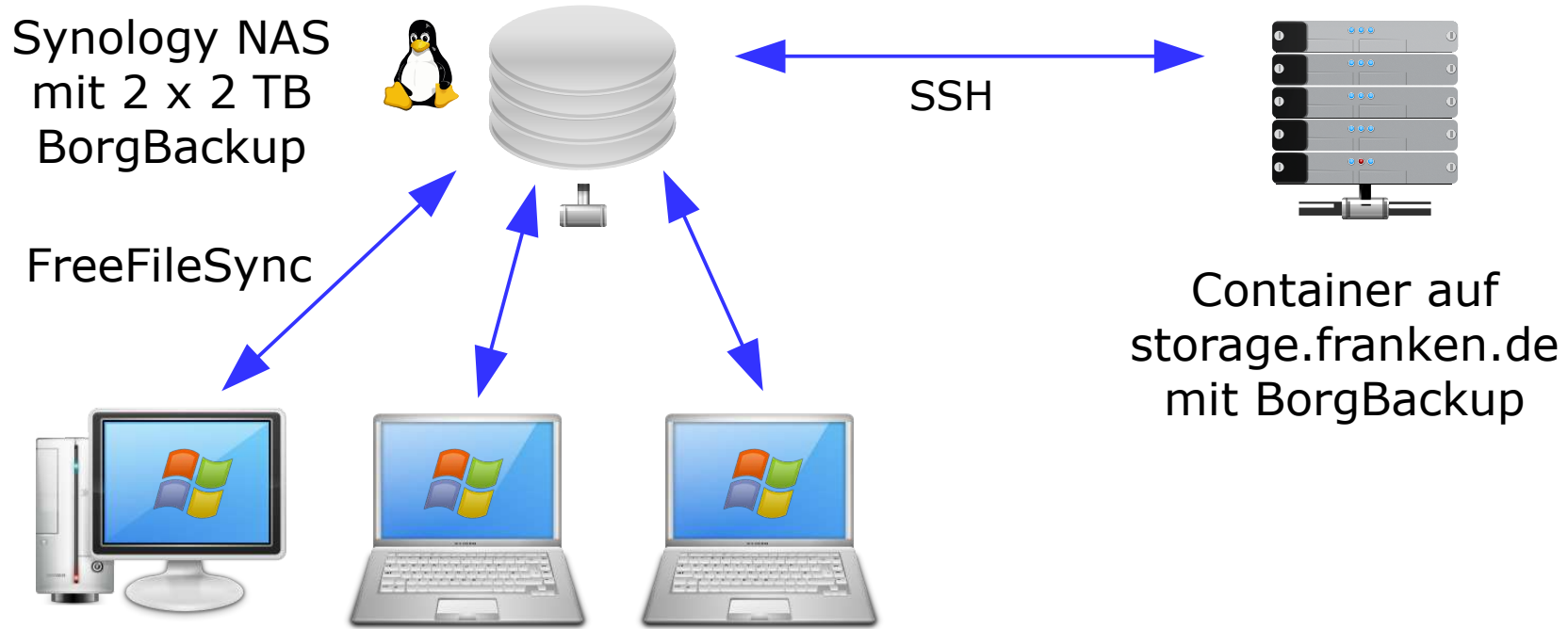


- Wie funktioniert BorgBackup?
 - Was sind die Ziele?
 - Welche Komponenten sind beteiligt und wie wirken sie zusammen?
 - Wie und in welchem Maße werden die Ziele erreicht?

➔ Wie wende ich BorgBackup an?

- Meine Konstellation
- Installation
- Betrieb

Meine Konstellation



Container auf storage.franken.de

- Schlüsselpaar für Container erzeugen
- Beim KNF einen eigenen Container beantragen. Dabei den öffentlichen Schlüssel mitgeben.
- Danach ist Einloggen über SSH mit dem privatem Schlüssel möglich
- Installation von BorgBackup mit dem Paketmanager
- Anlegen eines leeren Repositories mit Verschlüsselung. Dabei wird ein Schlüsselpaar erzeugt.
- Export des (Passwort-gesicherten) privaten Schlüssels



Synology NAS

- Den "Benutzer-Home-Dienst" aktivieren, damit home-Verzeichnisse angelegt werden
- Im Paket-Zentrum bei den Paket-Quellen die synocommunity eintragen. Von dort BorgBackup installieren.
- Mit SSH einloggen. Schlüssel für den Container hinterlegen, Schlüssel für das Repository hinterlegen.
- Shell-Skript erzeugen für die Sicherung
- Im Aufgabenplaner eine regelmäßige Aufgabe dafür einrichten

Shell-Skript



```
#!/bin/sh
# Damit das automatisch laeuft:
export BORG_REPO=ssh://Nutzer@storage.franken.de:221nn/data/Repository
export BORG_PASSPHRASE='Passwort'
```

```
# Starte Backup:
borg create --compression zstd,16 \
--checkpoint-interval 1200 \
--exclude '*/@eaDir' --stats \
:::{now:%Y-%m-%d} /volume1/NAS-Nutzer
```

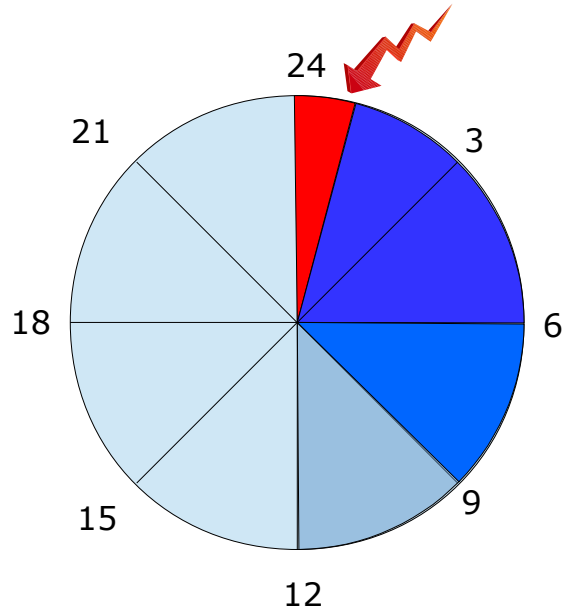
```
# Duenne Backups aus:
borg prune --list \
--keep-daily 7 \
--keep-weekly 4 \
--keep-monthly 6
```



Wann laufen lassen?

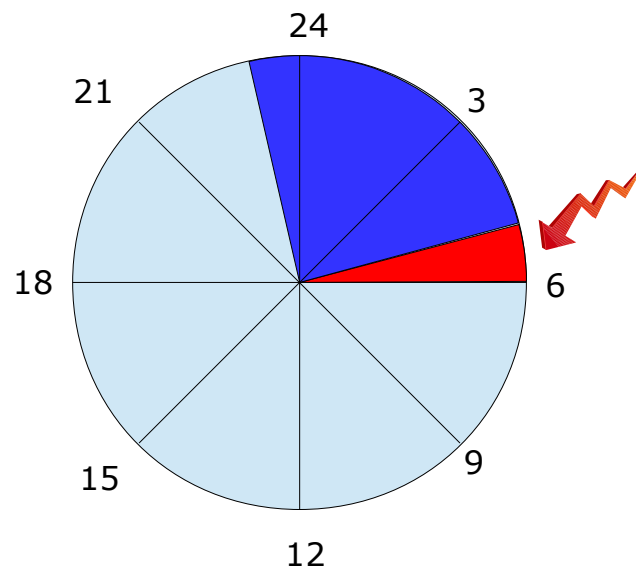


Variante A:
Direkt **nach** DSL-Zwangstrennung



Fast unbegrenzte Laufzeit.
Ggf. mit Bandbreitenbeschränkung

Variante B:
Bis DSL-Zwangstrennung



Wenn noch zu viele Daten ausstehen.

Ende



Vielen Dank für Ihre Aufmerksamkeit!



Herbert Oppmann

herby@memotech.franken.de

- BorgBackup <https://www.borgbackup.org/>
Deduplicating archiver with compression and encryption
- FreeFileSync <https://freefilesync.org/>
Folder comparison and synchronization software
- Synocommunity <https://synocommunity.com/>
Packages for Synology NAS

Bildnachweise (1)



Folie 1:

Meine Dokumente: Icon-Pack "Pleasant" von [Harven Zhang](#):

https://findicons.com/icon/51294/my_documents

BORG Logo: <https://www.borgbackup.org/>

Wolke: By The people from the Tango! project - The Tango! Desktop Project, Public Domain,

<https://commons.wikimedia.org/w/index.php?curid=675005>

Folie 2:

Person auf Treppe: von Clique Images auf Unsplash <https://unsplash.com/photos/hSB2HmJYaTo>

BOR-G Nummernschild: Eigenes Bild

Folie 4:

Golf-Fahnen: von Victor Ritsvall auf Unsplash <https://unsplash.com/photos/BIMH6-TIAMQ>

Folie 5:

Festplatte: By The Oxygen Team, KDE; - KDE github;, LGPL,

<https://commons.wikimedia.org/w/index.php?curid=18610807>

Netzwerk-Datenbank: By The Oxygen Team, KDE; - KDE github;, LGPL,

<https://commons.wikimedia.org/w/index.php?curid=18623088>

Server: By The Oxygen Team, KDE; - KDE github;, LGPL,

<https://commons.wikimedia.org/w/index.php?curid=18623089>

Bildnachweise (2)



Folie 5 (Fortsetz.):

Schloss: By Svengraph - <http://svengraph.deviantart.com/art/I-Love-Icons-178980215>, CC BY 3.0,
<https://commons.wikimedia.org/w/index.php?curid=47333860>

Folie 6:

Datenbank: By The Oxygen Team, KDE; - KDE github;, LGPL,
<https://commons.wikimedia.org/w/index.php?curid=18623099>

Folie 7:

Schokolade: von Charisse Kenion auf Unsplash <https://unsplash.com/photos/tWe8ib-cnXY>

Folie 10:

Schraubzwinge: By BESSEY Tool GmbH & Co. KG - BESSEY Tool GmbH & Co. KG, CC BY-SA 3.0 de,
<https://commons.wikimedia.org/w/index.php?curid=20568112>

Folie 11:

Schloss: By The Oxygen Team, KDE; - KDE github;, LGPL,
<https://commons.wikimedia.org/w/index.php?curid=40399947>

Folie 13:

Schubfächer: von Jan Kolar / VUI Designer auf Unsplash <https://unsplash.com/photos/IRoX0shwjUQ>

Bildnachweise (3)



Folie 16:

Computer: By The Oxygen Team, KDE; - KDE github;, LGPL,
<https://commons.wikimedia.org/w/index.php?curid=18611001>

Laptop: By The Oxygen Team, KDE; - KDE github;, LGPL,
<https://commons.wikimedia.org/w/index.php?curid=18610796>

Tux: By Larry Ewing, Simon Budig, Garrett LeSage - <http://www.home.unix-ag.org/simon/penguin/>,
garrett/Tux on GitHub, CC0, <https://commons.wikimedia.org/w/index.php?curid=753970>

Windows: Icon-Pack "Quiet" von Tidav: <https://findicons.com/icon/89007/windows>

Folie 17:

Werkzeug: By The Oxygen Team, KDE; - KDE github;, LGPL,
<https://commons.wikimedia.org/w/index.php?curid=18610596>

Folie 19:

Muschel: von Javardh auf Unsplash <https://unsplash.com/photos/dZ4Ylj91F2M>