

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

Die Integration von Cloud-Speichern wie Google Drive direkt in das macOS-Dateisystem bietet einen erheblichen Komfortgewinn. Werden Dateien jedoch unverschlüsselt übertragen und gespeichert, gibt man die Souveränität über die eigenen Daten vollständig an den Provider ab. Die Kombination aus **Rclone** und einer **FUSE-Schicht** löst dieses Problem, indem sie eine lokale Verschlüsselungsebene einzieht.

Bei der Wahl der technischen Schnittstelle stehen Nutzer heute jedoch vor einer Richtungsentscheidung: dem bewährten, aber alternden **macFUSE** oder dem modernen **FUSE-T**.

Warum FUSE-T die notwendige Wahl für moderne Systeme ist

Lange Zeit war **macFUSE** der unangefochtene Standard. Es operiert jedoch als **Kernel-Extension (KEXT)**. Da Apple die Architektur von macOS konsequent in Richtung eines abgesicherten Kernels entwickelt, bringt dieser Ansatz auf aktuellen Systemen (insbesondere mit Apple Silicon M1/M2/M3-Chips) gravierende Nachteile mit sich:

Eingriff in die Systemintegrität (SIP): Um macFUSE zu nutzen, muss die Sicherheit des Macs im Wiederherstellungsmodus auf „**Reduzierte Sicherheit**“ herabgestuft werden. Dies öffnet Tür und Tor für potenzielle Angriffe auf Systemebene, da die strikte Trennung zwischen Drittanbieter-Software und dem Betriebssystemkern aufgeweicht wird.

Systemstabilität: Da macFUSE im Kernel-Space läuft, führt ein Programmierfehler oder ein Konflikt nicht nur zum Absturz der App, sondern oft zu einem **Kernel Panic** – der gesamte Mac stürzt ab.

Zukunftssicherheit: Apple hat den Support für Kernel-Extensions faktisch abgekündigt. In kommenden macOS-Versionen wird die Nutzung dieser Treiber immer hürdenreicher oder gänzlich unmöglich.

FUSE-T hingegen geht einen fundamental anderen Weg. Es arbeitet vollständig im **User-Space**. Statt sich tief in das System einzugraben, nutzt es moderne Protokolle wie NFS oder das neue Apple FSKit, um dem Finder vorzugaukeln, es handele sich um ein lokales Netzwerk-Laufwerk.

Die Vorteile von FUSE-T auf einen Blick:

- **Volle Sicherheit:** Dein Mac bleibt im Modus „Volle Sicherheit“. SIP (System Integrity Protection) muss nicht angetastet werden.
- **Stabilität:** Ein Fehler im Dateisystem-Prozess beendet lediglich den Mount, reißt aber niemals das gesamte Betriebssystem mit in den Abgrund.
- **Einfachheit:** Keine komplizierten Neustarts im Recovery-Modus nötig. Die Installation erfolgt sauber und ohne tiefgreifende Systemänderungen.

Mit diesem Setup schaffst du eine Umgebung, die deine Daten schützt (durch Rclone-Verschlüsselung) und gleichzeitig die Integrität deines Macs bewahrt (durch FUSE-T).

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

Schritte der Installation

1. Die Installation der Software unter (macOS)

- a) FUSE-T installieren (via Brew)
- b) Rclone (Offizielle Binary) installieren:
- c) Prüfung

2. Konfiguration

- Schritt 1: Basis-Remote (gdrive_secure)
- Schritt 2: Crypt-Remote (gdrive_crypt)

3. Die dauerhafte Einbindung (macOS)

- a) Den Ordner erstellen
- b) Die LaunchAgent-Datei erstellen
- c) Den Dienst aktivieren

4. Wiederherstellung (Time Machine)

5. Überwachung von rclone

- a) Status-Abfrage des Dienstes
- b) Echtzeit-Statistiken via API
- c) Überprüfung des lokalen Zwischenspeichers
- d) Validierung des Cloud-Bestandes

6. Grafische Benutzeroberfläche (Web-GUI) via Terminal

7. Hinweise & Haftungsausschluss

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

1. Die Installation der Software unter (macOS)

Wichtig: Installiere Rclone **nicht** über Homebrew, da die Brew-Version unter macOS den mount-Befehl nicht unterstützt.

a) FUSE-T installieren (via Brew)

```
Bash
brew tap macos-fuse-t/homebrew-cask
brew install --cask fuse-t
```

b) Rclone (Offizielle Binary) installieren:

```
Bash
sudo -v ; curl https://rclone.org/install.sh | sudo bash
```

c) Prüfung

```
Bash
rclone version
```

Dies sollte eine aktuelle Version 1.73.x für Apple Silicon anzeigen:

```
rclone v1.73.3
- os/version: darwin 26.3 (64 bit)
- os/kernel: 25.3.0 (arm64)
- os/type: darwin
- os/arch: arm64 (ARMv8 compatible)
- go/version: go1.25.8
- go/linking: dynamic
- go/tags: cmount
```

Bild: Terminalausgabe von *rclone version* nach erfolgreicher installation von rclone.

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

2. Konfiguration

Führe den folgenden Befehl aus und folge dann den Schritten:

```
Bash
rclone config
```

Schritt 1: Basis-Remote (gdrive_secure)

- n (New Remote) -> Name: gdrive_secure -> Typ: drive.
- Alle Abfragen (Client-ID, Service Account, Advanced Config) mit **Enter** (leer lassen) überspringen.
- Use web browser to automatically authenticate? -> **y**.
- Im Browser einloggen und Zugriff erlauben.
- Configure this as a Shared Drive? -> **n**.
- Mit y bestätigen und speichern.

Schritt 2: Crypt-Remote (gdrive_crypt)

- n -> Name: gdrive_crypt -> Typ: crypt.
- Remote to encrypt: gdrive_secure:geheim
- filename_encryption: **1** (standard).
- directory_name_encryption: **1** (true).
- **Passwörter:** Wähle y, um dein eigenes **Secret** und danach dein eigenes **Salt** einzugeben. **Sichere beide Passwörter extern!**

```
Current remotes:

Name                Type
====                ==
gdrive_crypt        crypt
gdrive_secure       drive

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> █
```

Bild: Wenn die Konfiguration erfolgreich war, sollte ein Start *rclone config* diese Ausgabe liefern.

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

3. Die dauerhafte Einbindung (macOS)

a) Den Ordner erstellen

```
Bash
mkdir -p ~/Cloud/GoogleDrive
```

b) Die LaunchAgent-Datei erstellen

Erstelle die Datei plist-Datei mit nano

```
Bash
nano ~/Library/LaunchAgents/com.user.rclone.gdrive.plist
```

und füge diesen Inhalt ein (ersetze DEIN_USER durch deinen echten Namen, ggf. cache von 15 GB auf mehr oder weniger setzen):

```
XML

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.user.rclone.gdrive</string>
  <key>ProgramArguments</key>
  <array>
    <string>/usr/local/bin/rclone</string>
    <string>mount</string>
    <string>gdrive_crypt:</string>
    <string>/Users/frankwoithe/Cloud/GoogleDrive</string>
    <string>--vfs-cache-mode</string>
    <string>full</string>
    <string>--vfs-cache-max-size</string>
    <string>15G</string>
    <string>--rc</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
```

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

```
<key>KeepAlive</key>
<true/>
<key>StandardErrorPath</key>
<string>/tmp/rclone.err</string>
<key>StandardOutPath</key>
<string>/tmp/rclone.out</string>
</dict>
</plist>
```

c) Den Dienst aktivieren

Damit macOS die Datei registriert und den Dienst startet:

```
Bash
launchctl load ~/Library/LaunchAgents/com.user.rclone.gdrive.plist
```

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

4. Wiederherstellung (Time Machine)

Falls du deinen Mac neu aufsetzt:

1. **Installiere FUSE-T und Rclone** (Schritt 1 dieser Anleitung).
2. **Hole die Config zurück:** Kopiere `rclone.conf` aus dem Backup (versteckter Pfad `~/.config/rclone/`) zurück an den Platz, falls sie nicht mehr da sein sollte
3. **OAuth-Token erneuern:**

```
Bash
rclone config reconnect gdrive_secure:
```

4. **Dienst laden:** Führe die Schritte unter 3. *Die dauerhafte Einbindung* noch einmal aus.

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

5. Überwachung von rclone

Um die Integrität des Hintergrunddienstes sowie den Fortschritt umfangreicher Datentransfers (Migrationen) zu überwachen, stehen folgende Diagnosewerkzeuge zur Verfügung:

a) Status-Abfrage des Dienstes

```
Bash
launchctl list | grep rclone
```

Verifiziert, ob der LaunchAgent aktiv vom System verwaltet wird. Eine Ziffer in der ersten Spalte repräsentiert die Prozess-ID (PID) und signalisiert einen aktiven Dienst. Ein Bindestrich (-) mit einem Exit-Code ungleich 0 in der zweiten Spalte deutet auf einen Dienstabbruch hin.

b) Echtzeit-Statistiken via API

```
Bash
rclone rc core/stats
```

Liefert eine detaillierte JSON-Ausgabe zur aktuellen Performance. Hier lassen sich die Übertragungsrate (speed), die verbleibende Zeit (eta) sowie die aktuell im Transfer befindlichen Dateien einsehen. *Voraussetzung: Das --rc Flag ist in der .plist aktiv.*

c) Überprüfung des lokalen Zwischenspeichers

```
Bash
du -sh ~/Library/Caches/rclone/vfs/gdrive_crypt
```

Ermittelt die aktuelle physische Belegung des VFS-Caches auf der lokalen Festplatte. Nähert sich dieser Wert dem definierten Maximum (z. B. 15 GB), tritt ein Drosselungseffekt ein, um die Systemstabilität zu gewährleisten.

d) Validierung des Cloud-Bestandes

```
Bash
rclone size gdrive_crypt:
```

Führt eine rekursive Inventur des verschlüsselten Zielverzeichnisses durch. Dieses Werkzeug ist essentiell für den abschließenden Abgleich, ob die lokal initiierten Datenmengen vollständig in der Cloud registriert wurden.

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

6. Grafische Benutzeroberfläche (Web-GUI) via Terminal

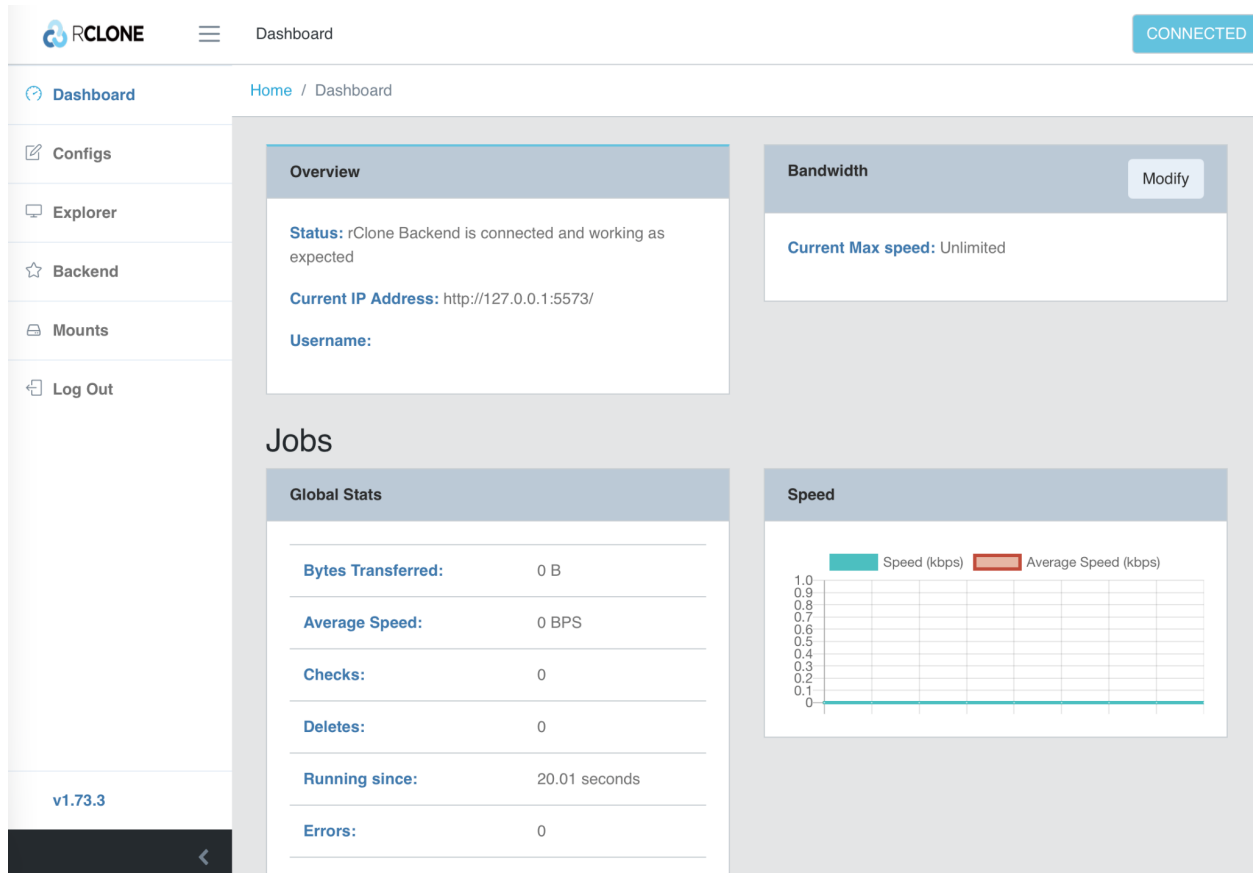


Bild: Weboberfläche von rclone

Um eine detaillierte visuelle Analyse der Transferraten und Verzeichnisstrukturen vorzunehmen, kann die rclone Web-GUI manuell über das Terminal gestartet werden. Diese Methode wird explizit dem permanenten Betrieb im Hintergrund vorgezogen, um **Systemressourcen zu schonen** und potenzielle **Sicherheitsrisiken** durch einen ständig erreichbaren lokalen Webserver zu minimieren.

Nutze für den Start exakt diesen Befehl:

```
Bash
rclone rcd --rc-web-gui --rc-addr localhost:5573 --rc-user User --rc-pass
Passwort
```

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

Erläuterungen zum Betrieb:

- **Modul-Initialisierung:** Bei der ersten Ausführung wird das benötigte Web-GUI-Modul automatisiert von rclone heruntergeladen und im lokalen Cache installiert. Dieser Vorgang erfolgt im Hintergrund und ist nur einmalig erforderlich.
- **Vermeidung von Port-Konflikten:** Durch die Zuweisung von Port **5573** (`--rc-addr localhost:5573`) wird sichergestellt, dass keine Kollision mit dem Standard-Port des im Hintergrund laufenden Mount-Dienstes auftritt. Die Oberfläche ist nach dem Start im Browser unter `http://localhost:5573` aufrufbar.
- **Authentifizierung:** Der Zugriff wird über die Parameter User und Passwort geschützt. Es ist ratsam, diese Zugangsdaten auch im lokalen Betrieb zu verwenden, um die Schnittstelle gegen unbefugte Zugriffe anderer Prozesse abzusichern.
- **Beenden des Dienstes:** Um den Webserver nach Beendigung der Überwachung wieder vollständig zu schließen und die Ressourcen freizugeben, wird die Tastenkombination **Ctrl + C** im Terminal verwendet.

Leitfaden: Verschlüsseltes Google Drive als natives Laufwerk unter macOS

7. Hinweise & Haftungsausschluss

Die vorliegende Anleitung wurde unter maßgeblicher Zuhilfenahme künstlicher Intelligenz erstellt. Die technische Konfiguration wurde in einem iterativen Verfahren schrittweise entwickelt, in der Praxis getestet und basierend auf den Ergebnissen fortlaufend optimiert. Zur Sicherstellung der Fehlerfreiheit und zur Validierung der Befehlsketten wurde ein strukturiertes Crosschecking unter Einbeziehung von drei Sprachmodellen (**Gemini**, **Claude** und **ChatGPT**) durchgeführt. Bei Unklarheiten wurde zudem eine Internetrecherche durchgeführt.

Haftungsausschluss:

Die Anwendung der hier beschriebenen Schritte erfolgt ausdrücklich auf eigenes Risiko. Trotz sorgfältiger Erstellung und mehrstufiger Testphasen übernimmt der Autor keinerlei Haftung für den Verlust von Daten, Beschädigungen an Hard- oder Software oder sonstige Folgeschäden, die durch die Nutzung dieser Anleitung entstehen könnten. Eine eigenständige Sicherung der Quelldaten (Backup) vor Beginn der Migration wird zwingend vorausgesetzt.