

WhispSkrid — Benutzerhandbuch

Offline-Push-to-talk-Diktat installieren und verwenden

Version v1.0.1

2026-09-18

Inhaltsverzeichnis

Kapitel 1 — Installation	2
In zwei Minuten	2
Voraussetzungen	2
Zwei Möglichkeiten zur Installation	3
Ein Modell wiederherstellen	4
Erster Start	4
Eine erste Diktierübung	5
Deinstallieren und aktualisieren	5
Kapitel 2 — Verwenden	6
Für wen ist dieses Werkzeug?	6
Das Interaktionsmodell: strikter Sprech-und-Hör-Modus	6
Zwei Wege, um eine Diktierübung zu starten	7
Persistente Sitzung und Client	7
Konfiguration	8
Modelle verwalten	9
Zusammenfassung der Befehlszeile	10
Kapitel 3 — Fehlerbehebung	11
Wie gelangt der Text in die Anwendung?	11
Das Mikrofon	11
Die Einspritzwerkzeuge und ihre automatische Auswahl	11
Abkürzungen: X11 und Wayland sind nicht gleichwertig	12
--diagnose	12
Symptome → wahrscheinliche Ursache → Maßnahmen	13
Wo kann ich eine Fehlermeldung finden?	14
Wenn das Problem weiterhin besteht	14

Kapitel 1 — Installation

In zwei Minuten

Für diejenigen, die sofort diktieren möchten, ohne den Rest zu lesen.

Installieren Sie das native Paket Ihrer Distribution.

```
# Debian 12 et 13, Ubuntu 22.04 et 24.04 et leurs dérivées
sudo apt install ./whispskrid_1.0.1_all.deb
```

```
# Fedora 42
sudo dnf install ./whispskrid-1.0.1-1.fc42.noarch.rpm
```

```
# openSUSE Leap 15.6
sudo zypper install ./whispskrid-1.0.1-1.leap156.noarch.rpm
```

```
# Arch Linux
sudo pacman -U whispskrid-1.0.1-1-any.pkg.tar.zst
```

2. Ein Whisper-Modell herunterladen, falls noch keines vorhanden ist (das Paket enthält keines – es wäre zu groß):

```
whispskrid --download-model base
```

Das Modell base wird heruntergeladen, überprüft und dann in `~/.local/share/whispskrid/whisper-models/` gespeichert. Beim ersten Start ohne ein vorhandenes Modell schlägt das Tool diesen Download interaktiv vor.

3. Starten Sie das Tool.

```
whispskrid
```

Er zeigt eine “bereit”-Leiste und wartet. Halten Sie die rechte **Umschalttaste**, sprechen Sie und lassen Sie los: Der transkribierte Text wird in das aktive Fenster geschrieben.

Das war’s für den Anfang. Der nächste Abschnitt erklärt, wie man von den Quellcodes installiert, ein anderes Modell oder eine andere Sprache auswählt und das Programm mit der Desktop-Tastatur bedient (Kapitel 2).

Voraussetzungen

- **Ein Linux-System** mit einer grafischen X11- oder Wayland-Session. Ohne einen Anzeigeserver gibt es kein aktives Fenster, in das der transkribierte Text geschrieben werden kann.
- **Python 3** und eine virtuelle Umgebung (`python3-venv`).
- **Ein Text-Injection-Tool:** `ydotool` oder `xdotool` zum Tippen, `wl-clipboard` unter Wayland oder `xclip` unter X11 für die Zwischenablage. Wenn ein funktionelles Eingabetool nicht verfügbar ist, startet das

Tool zwar, verweigert aber nicht den Start: es wechselt in einen reduzierten Modus (der transkribierte Text bleibt im Terminal angezeigt, ohne Eingabe – Kapitel 3).

- **Ein Mikrofon, das vom System** als Standardeingabegerät erkannt wird (Kapitel 3).
- **Festplattenspeicher für das Modell:** ungefähr 145 MB für das Modell base, mehr für größere Modelle (Kapitel 2). Während des Betriebs hält eine Session das Modell im Speicher – ungefähr 150 bis 300 MB RAM für base.

Das native Paket deklariert und installiert alle seine Abhängigkeiten, sowohl System- als auch Python-Abhängigkeiten, unabhängig von der Distribution. Im Git-Clone-Modus installiert `.venv/bin/pip install -e .` die Python-Bibliotheken; das Injektionswerkzeug (entweder `ydotoool` oder `xdotoool`, oder `wl-clipboard` oder `xclip`) muss dann separat installiert werden.

Zwei Möglichkeiten zur Installation

WhispSkrid bietet zwei Kanäle an, wobei der eine den anderen nicht ersetzt.

Das native Paket (empfohlen für den normalen Gebrauch)

```
# Debian 12 et 13, Ubuntu 22.04 et 24.04 et leurs dérivées
sudo apt install ./whispskrid_1.0.1_all.deb
```

```
# Fedora 42
sudo dnf install ./whispskrid-1.0.1-1.fc42.noarch.rpm
```

```
# openSUSE Leap 15.6
sudo zypper install ./whispskrid-1.0.1-1.leap156.noarch.rpm
```

```
# Arch Linux
sudo pacman -U whispskrid-1.0.1-1-any.pkg.tar.zst
```

Das Paket installiert einen einzelnen Befehl, `whispskrid`, in `/usr/bin/`. Es erstellt auch eine isolierte Python-Umgebung und installiert darin die Transkriptions-Engine (`faster-whisper`) – dieser Schritt benötigt während der Installation eine Netzwerkverbindung, nur einmal. Danach sind keine Superuser-Rechte erforderlich.

Jedes Paket wurde installiert, von `whispskrid --diagnose` überprüft und anschließend sauber in einem Container jeder der oben genannten Versionen (Debian 12 und 13, Ubuntu 22.04 und 24.04, Fedora 42, openSUSE Leap 15.6, Arch Linux) deinstalliert. Das Debian-Paket wurde zusätzlich in realen Nutzungsszenarien, einschließlich Spracherkennung, auf mehreren physischen Maschinen validiert, darunter ein System mit Debian 13; die anderen Pakete wurden dies noch nicht in einer echten grafischen Umgebung. Die anderen Systeme werden nicht abgedeckt – nicht aus Prinzip, sondern einfach aufgrund des Testumfangs.

Die Installation selbst ändert niemals persönliche Dateien: keine bestehende Konfiguration wird überschrieben, kein Benutzerordner wird erstellt oder verändert. Erst beim ersten Start und beim Herunterladen eines Modells erscheinen Dateien im persönlichen Ordner – niemals vorher und niemals ohne eine explizite Aktion des Benutzers.

Der Git-Clone (zum Verfolgen der Entwicklung oder zur Mitarbeit)

```
git clone https://github.com/RonanDavalan/whispskrid.git
cd whispskrid
python3 -m venv .venv
.venv/bin/pip install -e .
.venv/bin/whispskrid --diagnose
```

In diesem Modus wird das Werkzeug über `.venv/bin/whispskrid` anstelle des Befehls `whispskrid` des Pakets gestartet, und seine Konfiguration befindet sich in `config/config.yaml`, neben dem Code – niemals im persönlichen Verzeichnis. Die beiden Modi werden nicht miteinander vermischt: ein Git-Clone ignoriert die Konfiguration des Pakets, und umgekehrt.

Ein Modell wiederherstellen

Keiner der beiden Kanäle enthält ein Transkriptionsmodell – diese haben eine Größe von ungefähr 75 MB (`tiny`) bis zu mehreren Gigabyte (`large-v3`), was zu viel ist, um es mit dem Tool zu liefern.

Die einfachste Methode bei der Installation über ein Paket ist, dem Tool die Arbeit zu überlassen:

```
whispskrid --download-model base
```

Dieser Befehl lädt das angeforderte Modell – standardmäßig `base`, oder eines von `tiny`, `small`, `medium`, `large-v3` (und deren Varianten `.en`, spezialisiert für Englisch) – prüft, ob es geladen wird, installiert es in `~/.local/share/whispskrid/whisper-models/`, und wird dann beendet. Beim allerersten Start ohne ein Modell wird die gleiche Option interaktiv angeboten.

Das Modell und die Sprache sind zwei getrennte Optionen: Ein Whisper-Modell ist mehrsprachig, es ist die Option `-l` (oder `default_language` in der Konfiguration, Kapitel 2), die die Erkennungssprache festlegt. Ohne sie erkennt Whisper die Sprache selbstständig.

Wenn beim Start kein Modell gefunden wird, zeigt das Tool die durchsuchten Orte und die Vorgehensweise an – es kommt niemals zu einem stillen Fehler.

Erster Start

```
whispskrid
```

(Im Klon-Git-Modus ist der gleiche Befehl `.venv/bin/whispskrid`, ausgehend vom Root-Verzeichnis des Repositories.)

Beim Start lädt das Tool die Konfiguration, lädt das Modell einmalig und öffnet das Mikrofon. Anschließend wird eine “Bereit”-Meldung angezeigt und das Programm wartet. Dieser Prozess läuft im Vordergrund und stellt die persistente Sitzung dar. Das Modell bleibt im Speicher, sodass jede Diktieraufgabe ohne Ladezeit gestartet werden kann (Kapitel 2). Nur eine Sitzung kann gleichzeitig aktiv sein.

Um zu beenden: **Strg+C** im Startterminal, oder `whispskrid --stop` von einem anderen Terminal aus.

Eine erste Diktierübung

WhispSkrid funktioniert mit “Push-to-Talk”: Man hält eine Taste gedrückt, während man spricht, und lässt sie los, wenn man fertig ist.

1. Den Tastaturfokus an die Zielanwendung (Editor, Browser, Terminal) übertragen.
2. **Die rechte Umschalttaste gedrückt halten.** Ein kurzer Signalton bestätigt den Beginn der Aufnahme.
3. **Normal sprechen**, solange die Taste gedrückt gehalten wird.
4. **Loslassen.** Der aufgenommene Audio wird zur Transkription übergeben; der resultierende Text wird in das aktive Fenster eingefügt.

Es gibt keine Aufwachfunktion, keinen Wartezustand und keinen Timer: Die Aufnahme dauert genau so lange, wie die Taste gedrückt gehalten wird. Jeder Tastendruck erzeugt einen unabhängigen Schnappschuss.

Unter einer **Wayland**-Sitzung wird dieser Tastendruck nur für Fenster erfasst, die über XWayland laufen. Für ein natives Wayland-Fenster muss ein Steuerungscode an eine Desktop-Verknüpfung gebunden werden (Kapitel 2).

Deinstallieren und aktualisieren

Natives Installation. `sudo apt remove whispSkrid` (Debian), `sudo dnf remove whispSkrid` (Fedora), `sudo zypper remove whispSkrid` (openSUSE) ou `sudo pacman -R whispSkrid` (Arch) entfernt den Befehl und die isolierte Python-Umgebung (`/usr/lib/whispSkrid/venv`). Unter Debian löscht `sudo apt purge whispSkrid` zusätzlich die Werkseinstellungen. Die persönliche Konfigurationsdatei (`~/.config/whispSkrid/config.yaml`) und die Modelle (`~/.local/share/whispSkrid/whisper-models/`) werden bei der Deinstallation auf keiner Distribution verändert; diese können manuell gelöscht werden, falls gewünscht.

Git-Repository klonen. Das geklonte Verzeichnis löschen: es enthält alles (Code, virtuelle Umgebung `venv` /, eventuelle Modelle in `whisper-models/`). Nichts wird anderswo in diesem Modus gespeichert.

Update. Mit dem Paket: Installieren Sie die neueste Datei mit dem gleichen Befehl wie bei der Installation (`apt install`, `dnf install`, `zypper install` ou `pacman -U`): Ersetzen Sie die vorhandene Version, ohne die persönlichen Einstellungen zu verändern. Mit dem Git-Clone: Führen Sie `git pull` aus und starten Sie anschließend `.venv/bin/pip install -e .`, um eventuelle neue Abhängigkeiten zu berücksichtigen.

Kapitel 2 — Verwenden

Für wen ist dieses Werkzeug?

WhispSkrid ist ein Kommandozeilen- und Offline-Spracherkennungstool, das auf Whisper basiert. Es zielt auf eine einfache und vorhersehbare Nutzung ab: Man hält eine Taste, spricht, lässt die Taste los und der Text wird in das aktive Fenster geschrieben. Nichts weiter – keine Aktivierungsphrase, keine Sprachbefehle, kein Status, der verwaltet werden muss.

Die Transkription wird durch `faster-whisper`, eine Implementierung von Whisper, bereitgestellt, die auf CTranslate2 basiert und für die Ausführung auf der CPU optimiert ist. Das Standardmodell ist `base`, ein guter Kompromiss zwischen Geschwindigkeit und Genauigkeit für die meisten Systeme; es kann jederzeit geändert werden (siehe unten). Die gesamte Spracherkennung findet lokal statt: kein Ton und kein Text verlassen den Computer, zu keinem Zeitpunkt.

Das Interaktionsmodell: strikter Sprech-und-Hör-Modus

Eine Diktierübung ist ein isolierter Abschnitt, ohne Bezugnahme von einem Abschnitt zum nächsten.

1. **Aktivierung** – Drücken und Halten der Aktivierungstaste für die Spracherkennung. Die Audioaufnahme beginnt; ein kurzer Signalton bestätigt dies.
2. **Aufnahme** – Das Audio wird so lange aufgezeichnet, wie die Taste gedrückt gehalten wird. Es gibt keine automatische Unterbrechung, keinen Timer oder eine Erkennung des Satzendes.
3. **Loslassen** – Die Aufnahme stoppt. Der gesamte Audio-Puffer wird in einem Durchgang transkribiert.
4. **Einfügen** – Der transkribierte Text wird nach einer leichten Nachbearbeitung in das aktive Fenster eingefügt (Kapitel 3 für den Mechanismus).

Es gibt keinen “gespeicherten” Text zwischen zwei Diktaten, daher gibt es keinen Befehl zum “Kopieren der Sitzung”: jede Ausführung erzeugt ihre eigene Injektion.

Die Zeitbegrenzung

Nur automatische Abschaltung: `capture.max_seconds` (standardmäßig 300 Sekunden). Wenn die Taste länger gedrückt gehalten wird, stoppt die Aufnahme automatisch, was aufgezeichnet wurde, wird transkribiert und normal eingefügt, und eine Warnung wird protokolliert. Dies ist kein Komfort-Timer – es ist ein Schutz vor einer blockierten Taste oder einem Audiopuffer, der endlos anschwillt.

Die Stummschaltung ist nicht aktiviert

Die automatische Erkennung von Sprachbeginn und -ende (VAD) ist noch nicht verfügbar. Die Schlüssel `vad.enabled` (Standardwert: `false`) und `vad.silence_ms` sind bereits in der Konfiguration vorhanden; `vad.enabled: true` gibt eine Warnung “noch nicht implementiert” aus, und das Tool setzt den Betrieb im Modus “nur bei gedrücktem Knopf sprechen” fort.

Zwei Wege, um eine Diktierübung zu starten

Die lokale Taste

Ein Hotkey-Listener (pynput) überwacht die Tastatur. Der Standardtastencode ist **rechte Shift-Taste** (`shift_r`), ohne einen globalen Standard-Shortcut für die üblichen Desktops, und ist einfach zu halten. Er kann über `hotkeys.push_to_talk` in der Konfiguration geändert werden. Auf diese Weise ist das Halten standardmäßig aktiviert: die Aufnahme dauert genau von der Betätigung bis zur Freigabe.

pynput Beobachten Sie den X-Server. Unter Wayland werden nur Fenster erfasst, die über XWayland laufen, niemals ein natives Wayland-Fenster – dies ist eine Best-Effort-Lösung. Um diesen Listener niemals zu starten, setzen Sie `hotkeys.pynput_enabled: false` und steuern Sie ihn über den unten beschriebenen Pfad.

Die Unterbefehle zur Steuerung

Universeller, vom Sitzungstyp unabhängiger Weg: Diese Unterbefehle werden den globalen Desktop-Verknüpfungen zugeordnet. Da eine Desktop-Verknüpfung das Konzept “Taste gedrückt / Taste losgelassen” nicht unterstützt, funktioniert dieser Weg im Umschaltmodus:

Aktion	Unterbefehl
Start der Aufnahme	<code>whispskrid --dictate</code>
Die Aufnahme stoppen und injizieren	<code>whispskrid --dictate-stop</code>
Starten oder Stoppen, je nach Status	<code>whispskrid --toggle</code>
Den aktuellen Snapshot ohne Injektion verwerfen	<code>whispskrid --cancel</code>
Den aktuellen Status kennen	<code>whispskrid --status</code>
Aktuelle Sitzung beenden	<code>whispskrid --stop</code>

Beispiel unter KDE Plasma: In *Systemeinstellungen* → *Tastenkombinationen* → *Benutzerdefinierte Tastenkombinationen*, erstellen Sie einen Eintrag mit der Bezeichnung “Befehl/URL”, dessen Befehl `whispskrid --toggle` ist, und weisen Sie ihm eine Tastenkombination zu. Einmaliges Drücken startet die Aufnahme, ein zweites Mal stoppt sie und injiziert die Daten.

Die beiden Wege steuern den gleichen internen Zustand. Eine gestartete Aufnahme, die durch eine Taste ausgelöst wurde, kann durch `--cancel` abgebrochen werden, und umgekehrt.

Persistente Sitzung und Client

`whispskrid` Der Befehl ohne Kontrollargumente startet eine **persistente Sitzung** im Vordergrund: er lädt die Konfiguration und das Modell, öffnet das Mikrofon, startet den Tastendruck-Listener und den Kontroll-Socket und wartet dann.

`whispskrid` wird mit einem Kontrollargument (`--dictate`, `--status`, `--stop...`) als **Client** ausgeführt: Es verbindet sich mit der laufenden Sitzung über eine Unix-Socket (in `$XDG_RUNTIME_DIR`), sendet den Befehl, zeigt die Antwort (OK oder ERR) an und beendet sich. Wenn keine Sitzung läuft, wird dies angezeigt und das Programm beendet sich mit einem Fehler.

Nur eine einzige persistente Sitzung kann gleichzeitig laufen: wenn der Socket bereits antwortet, verweigert eine neue Instanz den Start.

Konfiguration

Das gesamte Verhalten wird über eine einzige YAML-Datei, `config.yaml`, gesteuert.

Wo befindet sich die Datei?

Die Reihenfolge der Einträge ist von Bedeutung – der erste Eintrag hat Vorrang, die folgenden werden ignoriert:

1. Die Umgebungsvariable `WHISPSKRID_CONFIG`, falls definiert – wird unverändert verwendet, wird niemals automatisch erstellt.
2. `config/config.yaml` des Repositories, im Git-Clone-Modus.
3. `~/.config/whispskrid/config.yaml`, bei Paketinstallation – wird beim ersten Start automatisch aus einer Vorlage erstellt, mit einer Meldung, die angibt, wo die Datei erstellt wurde.
4. Die Vorlage selbst, schreibgeschützt.

In der Praxis, unabhängig vom installierten nativen Paket, wird `~/.config/whispskrid/config.yaml` – eine persönliche Datei, die niemals durch ein Paket-Update überschrieben wird.

Modell wechseln

```
models:
  default: "base"
  device: "auto"
  compute_type: "auto"
```

`models.default` ist ein kurzer Name, der im Modellordner aufgelöst wird, oder ein absoluter Pfad. `device` entspricht `cpu`, `cuda` oder `auto`.

`compute_type` akzeptiert vier Werte: `int8`, `int8_float16`, `float16` und `auto`. Mit `auto`, erfolgt die Auswahl automatisch: `int8` auf Prozessor, `float16` auf Grafikkarte.

`--model NOM` Auf der Kommandozeile überschreibt `models.default` für eine Session. Ein größeres Modell transkribiert feiner, allerdings mit einem höheren Speicherbedarf und einer höheren Latenz.

Sprache erzwingen

```
default_language: ""
```

Vide, die Benutzeroberfläche verwendet die System-Sprache, und die Spracherkennung erkennt die Sprache automatisch. `-l fr` (oder `--lang fr`) in der Kommandozeile erzwingt die Sprache der Benutzeroberfläche und der Spracherkennung für eine Sitzung, ohne die Konfigurationsdatei zu verändern.

Die Sprechtaete

```
hotkeys:
```

```
pynput_enabled: true
push_to_talk: ["shift_r"]
min_hold_ms: 250
mode: "hold"
```

Eine Funktionstaste ist ebenfalls geeignet (["f4"], geschützt vor Tastenkombinationen des Desktops). Die Auflistung mehrerer Tasten ergibt eine **Kombination**: alle müssen gleichzeitig gedrückt gehalten werden, um die Diktierfunktion zu starten, und im Modus `hold` löst das Loslassen einer Taste die Funktion ab und fügt sie ein. `min_hold_ms` bricht die Aktion ab, ohne einen kurzen Druck zu transkribieren – schützt vor einer versehentlichen Berührung, die eine Aufnahme auf Stille startet, die Whisper interpretieren könnte; hat außerhalb des Modus `hold` keine Wirkung. `mode: "toggle"` ändert die Funktionalität: ein Druck startet die Aufnahme, der nächste beendet sie und fügt sie ein, das Loslassen hat keine weitere Wirkung. `mode: "armed"` (Sprach- Phrasen-Erkennung) erfordert nicht mitgelieferte Modelle: das Aktivieren schlägt fehl, solange diese fehlen.

Die zeitliche Begrenzung und die Nachbearbeitung

```
capture:
  max_seconds: 300

post_processing:
  trim: true
  capitalize_sentence_start: true
```

`trim` entfernt die weißen Ränder des transkribierten Textes. `capitalize_sentence_start` erzwingt einen Großbuchstaben für den ersten Buchstaben, der eingefügt wird. Whisper setzt selbst die Zeichensetzung und die Großschreibung am Satzanfang; die Nachbearbeitung beschränkt sich auf diese beiden Einstellungen; es gibt weder eine Ersetzungstabelle noch eine Regel pro Sprache noch eine Einfügung von dünnen Leerzeichen.

Die Zwischenablage unter Wayland

```
clipboard:
  defer_restore: false
```

Mit `true` wird die Zwischenablage erst am Ende der Sitzung wiederhergestellt, anstatt nach jeder Diktierung. Dies ist nützlich mit bestimmten GTK-Anwendungen unter Wayland (Kapitel 3).

Modelle verwalten

Die Modelle werden in einem verwalteten Ordner gespeichert:

- installiert über Paket: `/usr/share/whispkrid/whisper-models/` (schreibgeschützt), ergänzt durch `~/local/share/whispkrid/whisper-models/` für die Benutzer-Downloads;
- aus dem Quellcode: `whisper-models/` im Stammverzeichnis des Repositories.

`whispkrid --download-model [NOM]` Lädt ein Modell in den Benutzerordner herunter und prüft, ob es

geladen wird. Größenordnung des Speicherplatzbedarfs bei Quantisierung int8 :

tiny	75 Mo
base	145 Mo
small	480 Mo
medium	1,5 Go
large-v3	3 Go

Im RAM-Speicher, beim Spitzenverbrauch: auf der CPU tiny ~0,4 GB, base ~0,5–0,6 GB, small ~1,3 GB, medium ~3,5–3,7 GB, large-v3 ~6,8 GB; auf der GPU (x86) tiny ~0,2 GB, base ~0,3 GB, small ~0,7 GB, medium ~1,7–2 GB, large-v3 ~3,2–3,9 GB VRAM. Werte wurden auf fünf Maschinen mit vier Sprachen gemessen – Werte pro Modell. In der Praxis: tiny und base laufen überall in Echtzeit, einschließlich auf ARM; small hinkt auf ARM hinterher (2- bis 3-mal langsamer); medium benötigt x86; large-v3 benötigt x86 mit ~8 GB RAM und läuft nur auf der GPU (auf der CPU, selbst auf x86 wird die Echtzeit überschritten).

Das Tool funktioniert offline, sobald das Modell vorhanden ist: der Hugging Face-Cache ist nur eine Notlösung, nicht der normale Weg.

Zusammenfassung der Befehlszeile

```
whispskrid [-l LANG] [--model NOM]
whispskrid --dictate | --dictate-stop | --toggle | --cancel
whispskrid --status | --stop
whispskrid --diagnose
whispskrid --download-model [NOM]
whispskrid --help | --version
```

Kapitel 3 – Fehlerbehebung

Wie gelangt der Text in die Anwendung?

WhispSkrid hat keine grafische Benutzeroberfläche – es schreibt den transkribierten Text direkt in die Anwendung, die gerade den Fokus der Tastatur hat, unabhängig davon, welche Anwendung das ist.

Der Text wird **nicht Zeichen für Zeichen eingegeben**. Der tatsächliche Mechanismus: der transkribierte Text wird in die Zwischenablage kopiert, und dann wird ein Tastenkürzel zum “Einfügen” ((Ctrl+V) simuliert. Das ist zuverlässiger und schneller als eine vollständige Simulation der Eingabe, insbesondere bei längeren Textabschnitten.

Die ursprüngliche Zwischenablage wird direkt nach dem Einfügen wiederhergestellt – das Diktieren überschreibt also nicht dauerhaft, was dort kopiert wurde. Unter Wayland mit einer GTK-Anwendung kann diese Wiederherstellung erfolgen, bevor das Fenster das Lesen der Zwischenablage abgeschlossen hat, wodurch der alte Inhalt eingefügt wird; die Einstellung `clipboard.defer_restore: true` verzögert die Wiederherstellung bis zum Ende der Sitzung und beseitigt dieses Risiko.

Das Mikrofon

WhispSkrid nimmt den Ton vom **Standard-Eingabegerät des Systems** auf (PulseAudio oder PipeWire). Es gibt keine Geräteauswahl in `config.yaml`: Um ein anderes Mikrofon zu verwenden, muss das gewünschte Gerät als Standardeingabe festgelegt werden – in den Soundeinstellungen der Desktopumgebung, mit `pavucontrol`, oder über die Kommandozeile (`pactl set-default-source <nom>`, wobei die Liste mit `pactl list sources short` oder `arecord -l` angezeigt wird).

`whispSkrid --diagnose` öffnet einen Test-Stream auf diesem Gerät und meldet, ob die Öffnung fehlschlägt – z. B. weil das Mikrofon fehlt, getrennt ist oder bereits von einer anderen Anwendung verwendet wird.

Die Einspritzwerkzeuge und ihre automatische Auswahl

Die Injektion erfordert **zwei** Werkzeuge: eine Schreibmaschine und ein Textbaustein-Werkzeug. Die Schreibmaschine wird nur einmal beim Start ausgewählt und direkt abgefragt – sie wird nie vor jedem Einfügen erneut geprüft:

- **Erste Wahl:** `ydotoool` (Eingabe über den Treiber `ydotoold`, Tastenkombinationen abhängig von der Tastaturbelegung).
- **Alternativ:** `xdotool` (Standardmodus mit Fenstererkennung).
- **Alternativ:** Degradierter Modus – der transkribierte Text bleibt im Terminal angezeigt, und die Eingabe ist für die Sitzung deaktiviert, anstatt bei jeder Diktierversuch einen stillen Fehler zu erzeugen.

Die Zwischenablage berücksichtigt den Sitzungstyp: `wl-clipboard` unter Wayland (immer erforderlich, auch mit `ydotool`), `xclip` unter X11. Ohne eine Zwischenablage gelangt der transkribierte Text niemals in die Zwischenablage: Nichts wird in der Anwendung gespeichert, und ein Ctrl+V fügt den zuletzt manuell kopierten Inhalt wieder ein.

Aktivieren Sie ydotool (Notfalllösung)

Dieser Abschnitt bezieht sich nur auf Systeme, bei denen der verwendete Motor ydotool ist. Wenn xdotool die Eingabe übernimmt, sind die folgenden Abschnitte nicht erforderlich – insbesondere ist es unnötig, der Gruppe input beizutreten.

Drei Bedingungen, die in der angegebenen Reihenfolge überprüft werden müssen, falls der Text nicht geschrieben werden kann:

1. **Das Paket ydotool ist installiert.** Unter Debian befindet es sich in trixie-backports, nicht im stabilen Zweig – aktivieren Sie dieses Repository, falls erforderlich.
2. **Der Dienst ydotoold läuft.** ydotool benötigt diesen Daemon im Hintergrund.
3. **Der Benutzer gehört zur Gruppe input.** Diese Gruppe gewährt Zugriff auf /dev/uinput, andernfalls schlägt ydotool stillschweigend fehl:

```
sudo usermod -aG input $USER
```

Es ist erforderlich, sich ab- und wieder anzumelden, damit diese Gruppenänderung wirksam wird.

Abkürzungen: X11 und Wayland sind nicht gleichwertig

Die integrierte Push-to-Talk-Funktion wird vom Tool selbst über pynput überwacht, das den X-Server beobachtet. Unter Wayland hört pynput nur die Fenster, die über XWayland laufen (viele Electron-Anwendungen und ältere Anwendungen), niemals ein natives Wayland-Fenster. Für diese letzteren muss ein Unterbefehl zur Steuerung (--toggle, --dictate, --dictate-stop) an eine Verknüpfung der Desktop-Umgebung gebunden werden (Kapitel 2).

--diagnose

whispkrid --diagnose Überprüft die Umgebung und gibt einen Wert zurück – der Rückgabewert ist 0, wenn alles in Ordnung ist, andernfalls ein anderer Wert. Jeder Punkt wird in einer eigenen Zeile ok / !! dargestellt:

- **Umgebung** – Sitzungstyp (X11 / Wayland), verwendetes Injektions-Backend.
- **Werkzeuge** – Vorhandensein von ydotool, ydotool, xdotool, wl-copy / wl-paste oder xclip, paplay ; Lese- und Schreibzugriff auf /dev/uinput.
- **Backend** – Import von faster_whisper und ctranslate2 ; device und compute_type aktiv; CUDA verfügbar oder nicht (informativ).
- **Modelle** – Pfad zum gelösten Modellordner; Vorhandensein des Standardmodells; Test laden (Speicherbelegung und anschließende Freigabe).
- **Audioeingang** – Öffnen eines Capture-Streams im erwarteten Format.
- **Zwischenablage** – Lese-/Schreibzugriff.

- **Kontroll-Socket** — Pfad, zugängliches Ausführungsverzeichnis (mit Schreibrechten), aktive Sitzung oder nicht.
- **Konfiguration** — Pfad zur tatsächlich geladenen `config.yaml`.

Symptome → wahrscheinliche Ursache → Maßnahmen

Symptome	Wahrscheinliche Ursache	Aktion
Der transkribierte Text bleibt im Terminal, es wird nichts injiziert	Keine funktionierenden Eingabemethoden (weder <code>ydotool</code> noch <code>xdotool</code>): reduzierter Modus	Installer <code>ydotool</code> oder <code>xdotool</code> , dann die Sitzung neu starten
Die Meldung “kein Modell gefunden” beim Start	Kein Modell an den erwarteten Stellen gefunden	<code>whispskrid --download-model base</code> — siehe Kapitel 1
Nichts wird transkribiert, das Mikrofon scheint stumm zu sein	Schlechte Standard-Eingabegeräte oder Mikrofon, das auf Systemebene deaktiviert ist	Überprüfen / Ändern der Standardeinstellung (<code>pavucontrol</code>); <code>whispskrid --diagnose</code>
<code>--diagnose</code> signalisiert den Fehler beim Öffnen des Audio-Streams	Ein anderes Programm verwendet dieses Gerät, oder Sie haben nicht die erforderlichen Zugriffsrechte	Schließen Sie die andere Anwendung; überprüfen Sie die Berechtigungen des Geräts
Nichts wird in der App geschrieben, es werden keine Fehler angezeigt	Das Werkzeug hat sich auf <code>ydotool</code> gefaltet, aber der Benutzer ist nicht in der Gruppe <code>input</code> enthalten	Der Gruppe beitreten, wieder in Kontakt treten – siehe oben
Die App fügt immer den zuletzt manuell kopierten Inhalt wieder ein	Kein Clipboard-Tool installiert (<code>wl-clipboard</code> unter Wayland, <code>xclip</code> unter X11)	Installieren Sie es
Unter Wayland zeigt eine GTK-Anwendung kurzzeitig den alten Inhalt, bevor der neue Inhalt angezeigt wird	Die Wiederherstellung des Ausschnitts erfolgt, bevor das Fenster die Auswahl angezeigt hat	<code>clipboard.defer_restore: true</code> in der Konfiguration
Die rechte Umschalttaste funktioniert nicht	Fenster im nativen Wayland-Modus werden in den Vordergrund geholt — <code>pynput</code> zeigt nur X11- und XWayland-Fenster	Lier <code>--toggle</code> / <code>--dictate</code> ist eine Verknüpfung zum Desktop – siehe Kapitel 2
Kein Ton zu Beginn der Aufnahme	<code>paplay</code> nicht vorhanden oder kein Audio-Datei konfiguriert	Wichtiger Hinweis: Installieren Sie <code>pulseaudio-utils</code> falls Sie Ton wünschen

Symptome	Wahrscheinliche Ursache	Aktion
Die Aufnahme stoppt automatisch nach fünf Minuten <code>vad.enabled: true</code> ohne Wirkung	Die Schutzvorrichtung <code>capture.max_seconds</code> ist erreicht Die Erkennung von Stille ist in dieser Version nicht implementiert	Gewünschtes Verhalten; ggf. Wert in der Konfiguration anpassen – siehe Kapitel 2 Nur für fortgeschrittene Benutzer; das Gerät verwendet weiterhin die Push-to-Talk-Funktion – siehe Kapitel 2
Die Transkription ist langsam	Das Modell ist zu groß für die Maschine, oder <code>compute_type</code> es ist ungeeignet	Wechseln zu <code>base</code> oder <code>small</code> ; beibehalten von <code>compute_type: auto</code> – siehe Kapitel 2
Die geänderte Konfiguration scheint nicht berücksichtigt zu werden “Eine Sitzung läuft bereits.”	Die bearbeitete Datei ist nicht die tatsächlich verwendete Datei Eine persistente Sitzung läuft bereits (es kann nur eine gleichzeitig aktiv sein)	Überprüfen der Lösungsreihenfolge – siehe Kapitel 2 <code>whispSkrid --stop</code> , oder die bestehende Sitzung verwenden

Wo kann ich eine Fehlermeldung finden?

WhispSkrid zeigt seine Fehler direkt im Terminal an, in dem die persistente Sitzung gestartet wurde. Das erneute Starten von einem sichtbaren Terminal ist der erste Schritt zur Diagnose eines Problems, das hier nicht aufgeführt ist. Im Zweifelsfall `whispSkrid --diagnose` überprüft alles und zeigt den Status jedes einzelnen Punktes an.

Wenn das Problem weiterhin besteht

Der Projekt-Repository hat eine öffentliche Tracking-Seite: <https://github.com/RonanDavalan/whispSkrid/issues>