

Update: 26.2.2026 - 00.32

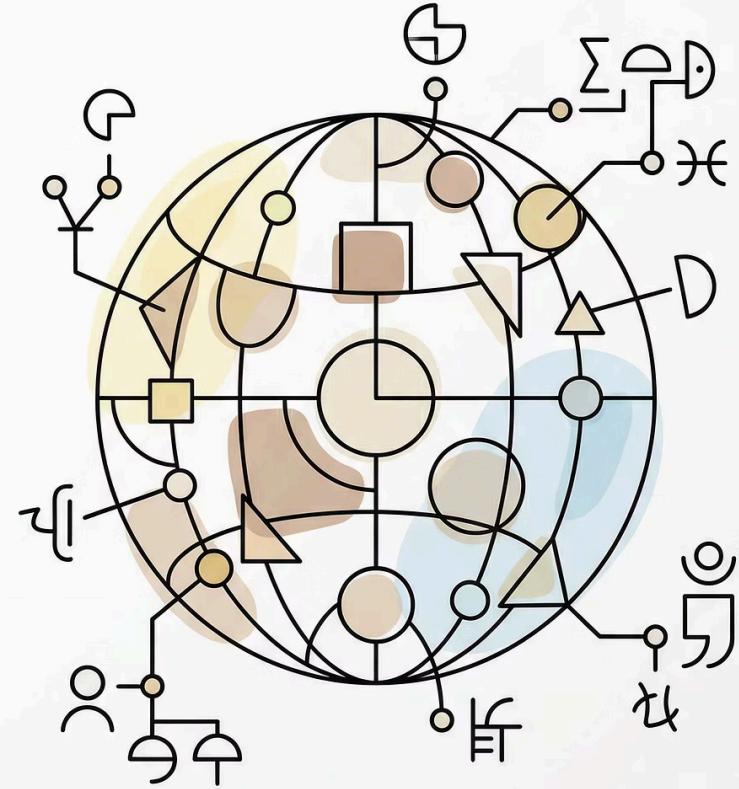
# Guidetranslator v6.

**guidetranslator – Von V.6 bis V.6.b+:** Security-Fixes, App Store Blocker behoben, Rebranding, iOS STT-Pipeline neu gebaut, satzweise Segmentierung mit Hybrid-Toggle. **Alexanders Tests trieben jeden Fix. Launch-ready.** 🚀

## ☐ Für Ulrich: 🎉 Deine App funktioniert – und sie wird jeden Tag besser!

Ich habe eine Übersetzungs-App gebaut, die auf deinem Handy läuft, Sprache erkennt, sofort übersetzt und vorliest. Auf Knopfdruck, ohne Anmeldung, in 45 Sprachen.

In den letzten Stunden habe ich alles sicherer, schneller und stabiler gemacht – und alle Fehler, die ich bei ausgiebigen Testen gefunden habe, sind behoben. Die App ist jetzt bereit für den App Store.





# Diagnose: Warum Echtzeit-Übersetzung stockt

Alexanders detaillierter Test hat 5 Kernprobleme in der STT-Pipeline aufgedeckt. Alle haben eine gemeinsame Lösung: satzweise Segmentierung.

5

Kernprobleme

2

STT-Engines betroffen

1

Lösung: Segmentierung

2

Modi (Toggle)

# Satzweise Segmentierung – Hybrid mit Toggle

## Technische Lösung

- Automatische Satz-Erkennung in der STT-Schicht (. ! ? erkennen)
- Jeder erkannte Satz → sofort als eigenes Segment übersetzen + vorlesen
- Interim-Text live anzeigen (grau/kursiv unter dem Textfeld)
- "Senden"-Button als Escape-Hatch: erzwingt sofortige Übersetzung
- Manuelles Tippen bleibt wie bisher (600ms Debounce)

## Hybrid Toggle

### Satz-Modus (Default)

Automatische Segmentierung bei . ! ? – sofortige Übersetzung pro Satz

### Absatz-Modus

Manueller "Senden"-Button – User kontrolliert wann übersetzt wird

- Beide Modi zeigen Interim-Text live an. Der Senden-Button ist immer als Escape-Hatch verfügbar.

# iOS Safari: Web Speech API ersetzt durch Google Cloud STT

## 🔍 Problem

- iOS Safari hat webkitSpeechRecognition im Window, aber .start() schlägt sofort mit service-not-allowed fehl
- Betrifft ALLE iOS-Browser (alle nutzen WebKit)
- Wichtige Erkenntnis: iOS Safari ignoriert sampleRate: 16000 und nutzt stattdessen 48kHz → sampleRate muss dynamisch gelesen werden



## 🔧 Lösung: createGoogleCloudSTTEngine()

- Neuer Engine in `stt.ts` – nutzt denselben Google API Key wie TTS/Translate
- Audio-Aufnahme via Web Audio API (wie Whisper-Engine)
- Sample Rate dynamisch via `audioContext.sampleRate` (nicht hardcoded!)
- Sendet alle 3s PCM-Audio (LINEAR16/Base64) an `speech.googleapis.com`
- Auf iOS wird die kaputte Web Speech API automatisch übersprungen

- ❑ Doppelte Absicherung: `isIOS()` erkennt iOS upfront + Auto-Fallback wenn `service-not-allowed` feuert → Hook wechselt automatisch, ohne User-Aktion.

## ENGINE-PRIORITÄT

# 5-stufige STT Engine-Priorität

01

## Apple Native Bridge

Zukunft: native iOS Integration (noch nicht implementiert)

02

## Google Cloud STT (iOS)

Aktiv auf iOS: Web Speech API wird übersprungen, direkt Google Cloud

03

## Web Speech API

Chrome / Edge / Android: wie bisher, schnell und zuverlässig

04

## Google Cloud STT (Fallback)

Genereller Fallback: greift wenn Web Speech service-not-allowed feuert

05

## Whisper Offline

Letzter Ausweg: vollständig offline, kein Internet nötig

- ❑ Alexanders Testergebnisse haben direkt zu dieser Architektur geführt. Danke für die präzisen Beobachtungen! 🙌

# Die 'Sprünge' im Text – was wirklich passiert

## 🔍 Alexanders Beobachtung

- Text springt von 362 → 725 → 996 → 2238 Zeichen
- Kurze Sätze → sofortige Übersetzung ✓
- Schnell hintereinander sprechen → sehr lange Wartezeit ✗
- Kein Feedback während des Sprechens

## ⚙ Ursache: Snowball-Effekt

- Google Cloud STT: isFinal nur beim Stop → kein Output während Sprechen
- Web Speech API: wartet auf Stille → bei schnellem Sprechen lange Pause
- GESAMTER akkumulierter Text wird jedes Mal neu übersetzt (2238 Zeichen = langsam)
- 600ms Debounce obendrauf
- Keine Interim-Anzeige → User sieht "nichts"

❑ Alexanders präzise Protokollierung der Zeichensprünge (362→725→996→2238) hat die Diagnose erst möglich gemacht.

# Update: iOS STT Fix & Pipeline-Optimierung

iOS Safari Web Speech API komplett ersetzt. Neue Google Cloud STT Engine, Auto-Fallback, satzweise Segmentierung und Hybrid-Toggle implementiert.

5

Dateien geändert

+409

Zeilen hinzugefügt

5

Engine-Prioritäten

2

Modi (Toggle)

# Was geändert wurde – 5 Dateien, +409 Zeilen

## stt.ts

- detectSentenceBoundary(): erkennt Satzenden (. !? ; + CJK/Arabisch)
- Google Cloud STT: Polling alle 3s mit Satzgrenzenerkennung → synthetische isFinal: true auf iOS
- Audio-Buffer nach jedem Final getrimmt (nur letzte 5s behalten)
- Web Speech API: Interim auf Satzgrenzen geprüft, Dedup-Logik

## useSpeechRecognition.ts

- Neuer onInterimResult-Callback
- interimTranscript State
- Auto-Fallback: service-not-allowed → automatisch Google Cloud STT

## TranslationPanel.tsx

- Segment-Architektur statt monolithischem Textblock
- Hybrid-Toggle: ⚡ Satz-Modus vs. ¶ Absatz-Modus
- Senden-Button während Aufnahme sichtbar
- Interim-Text grau/kursiv unter Textarea

## i18n.ts

- 5 neue Keys (sentence, paragraph, sentenceMode, paragraphMode, send) in allen 9 UI-Sprachen