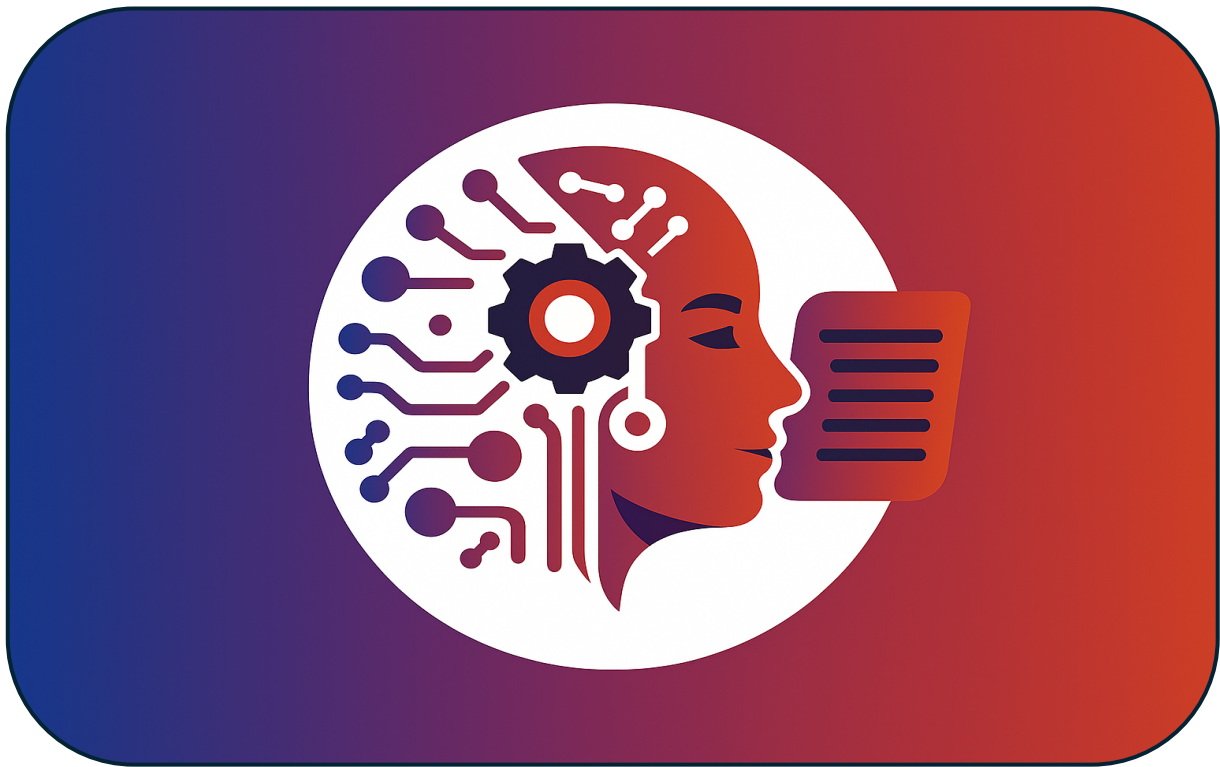


AILO

ai-powered communication and logging tool for iOS

Projekt-Dokumentation

Version 1.0



Inhaltsverzeichnis

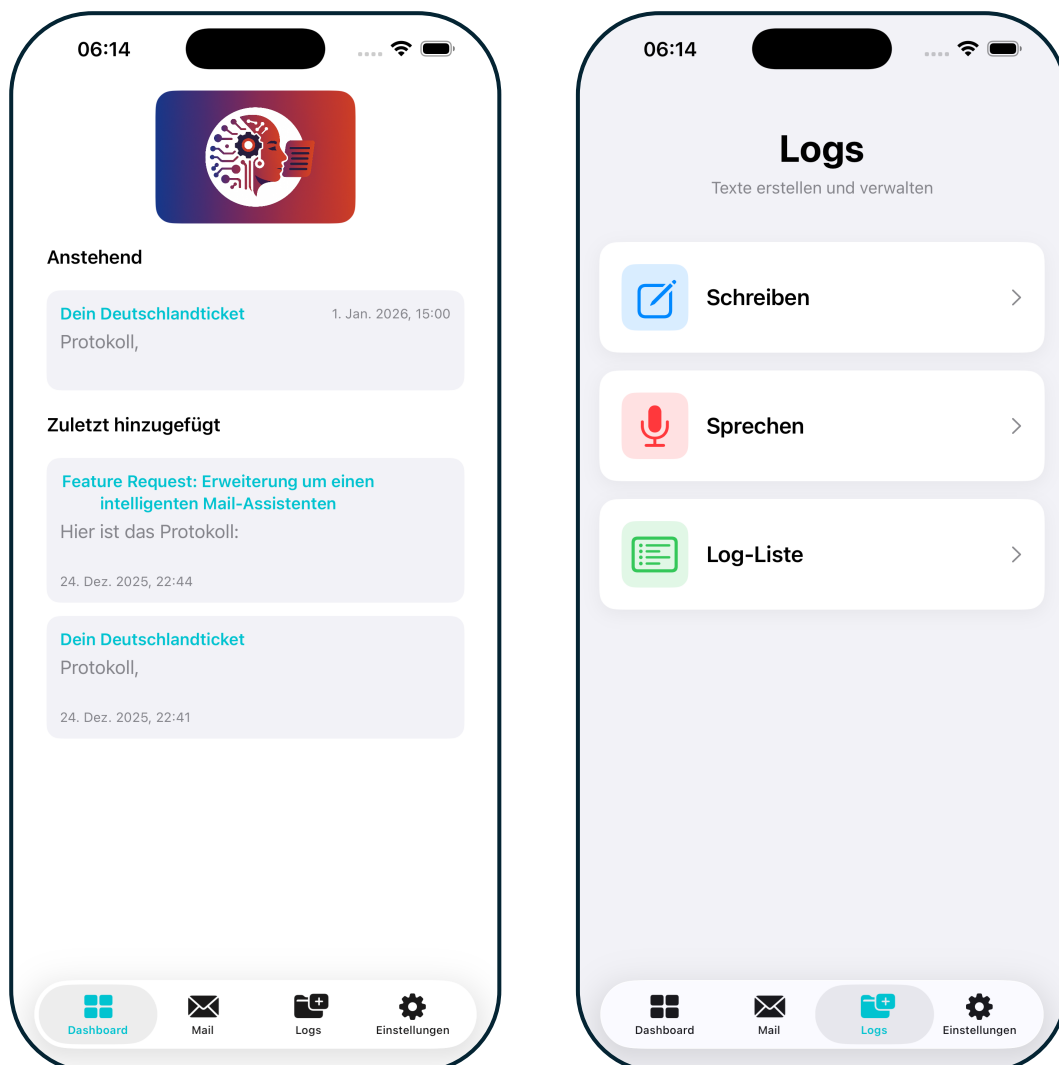
1. Projektübersicht
2. Technologie-Stack
3. Pre-Prompt Katalog System
4. Log-System
5. E-Mail Integration
6. KI-Integration & Prompt Manager
7. Architektur-Übersicht

1. Projektübersicht

AIILO ist eine native iOS-Applikation, die KI-Unterstützung für die Verwaltung von Notizen, E-Mails und Audio-Logs bereitstellt. Die App kombiniert klassische Produktivitätsfunktionen mit intelligenter Textverarbeitung und Spracherkennung.

1.1 Hauptfunktionen

- **Schreiben (Write):** Schnelle Texteingabe mit Kategorisierung, Tags, Erinnerungen und Markdown-Unterstützung
- **Sprechen (Speak):** Audio-Aufnahme mit Live-Transkription und automatischer Spracherkennung
- **Logs:** Zentrale Übersicht aller Einträge mit Volltextsuche und Export-Funktionen
- **E-Mail Integration:** IMAP/SMTP-Unterstützung mit Multi-Account-Verwaltung
- **KI-Integration:** OpenAI und Ollama Support mit konfigurierbaren Pre-Prompts



2. Technologie-Stack

- **Plattform:** iOS 16+, macOS 13+ (Catalyst)
- **Sprache:** Swift 5.9+
- **UI Framework:** SwiftUI
- **Datenbank:** SQLite mit DAO-Pattern
- **Audio:** AVFoundation, Speech Framework
- **Netzwerk:** URLSession, SwiftNIO (SMTP/IMAP)
- **Sicherheit:** Keychain für sensible Daten

2.1 Projektstruktur

AIILO_APP/

└─ App/	# App-Entry & Navigation
└─ Features/	# Feature-Module
└─ Services/	# Business Logic (AI, Audio, Mail)
└─ Database/	# SwiftData Models & DAOs
└─ Configuration/	# Settings & Language
└─ Helpers/	# Utilities & Parsers
└─ Views/	# Shared UI-Components

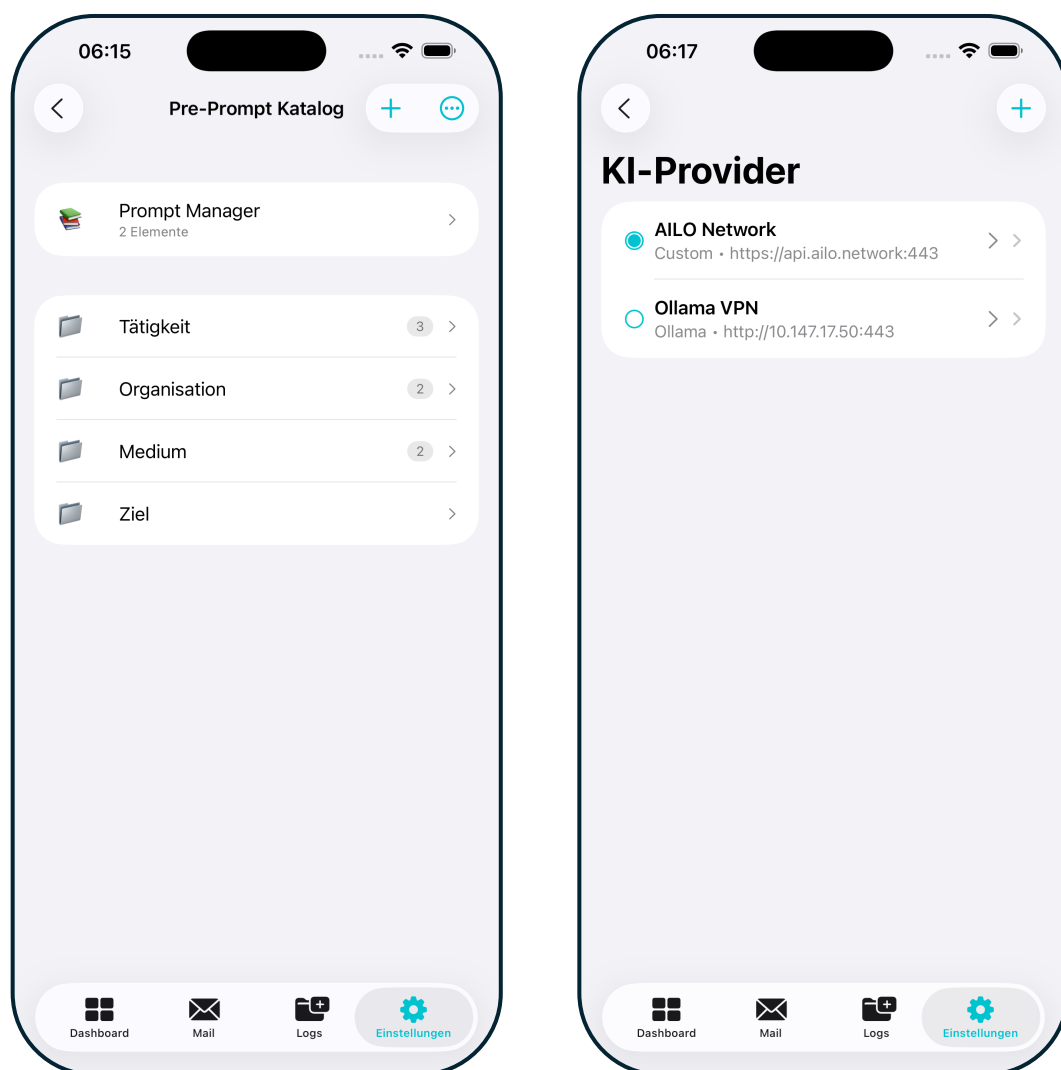
3. Pre-Prompt Katalog System

Das Pre-Prompt Katalog System ist ein zentrales Feature von AIILO, das eine hierarchische Verwaltung von KI-Anweisungen ermöglicht. Es bietet eine flexible Struktur für die Organisation und Wiederverwendung von Prompts.

3.1 Architektur des PrePromptCatalogManager




Der PrePromptCatalogManager ist als Singleton implementiert und verwaltet folgende Datenstrukturen:

1. **menuitems:** Hierarchische Menüstruktur mit Ordnern und Preset-Referenzen
2. **presets:** Die eigentlichen Pre-Prompt Inhalte (AIPrePromptPreset)
3. **recipes:** Kombinationen aus mehreren Presets (PrePromptRecipe)
4. **cookbooks:** Sammlungen von Rezepten für komplexe Workflows



3.2 Standard-Kategorien

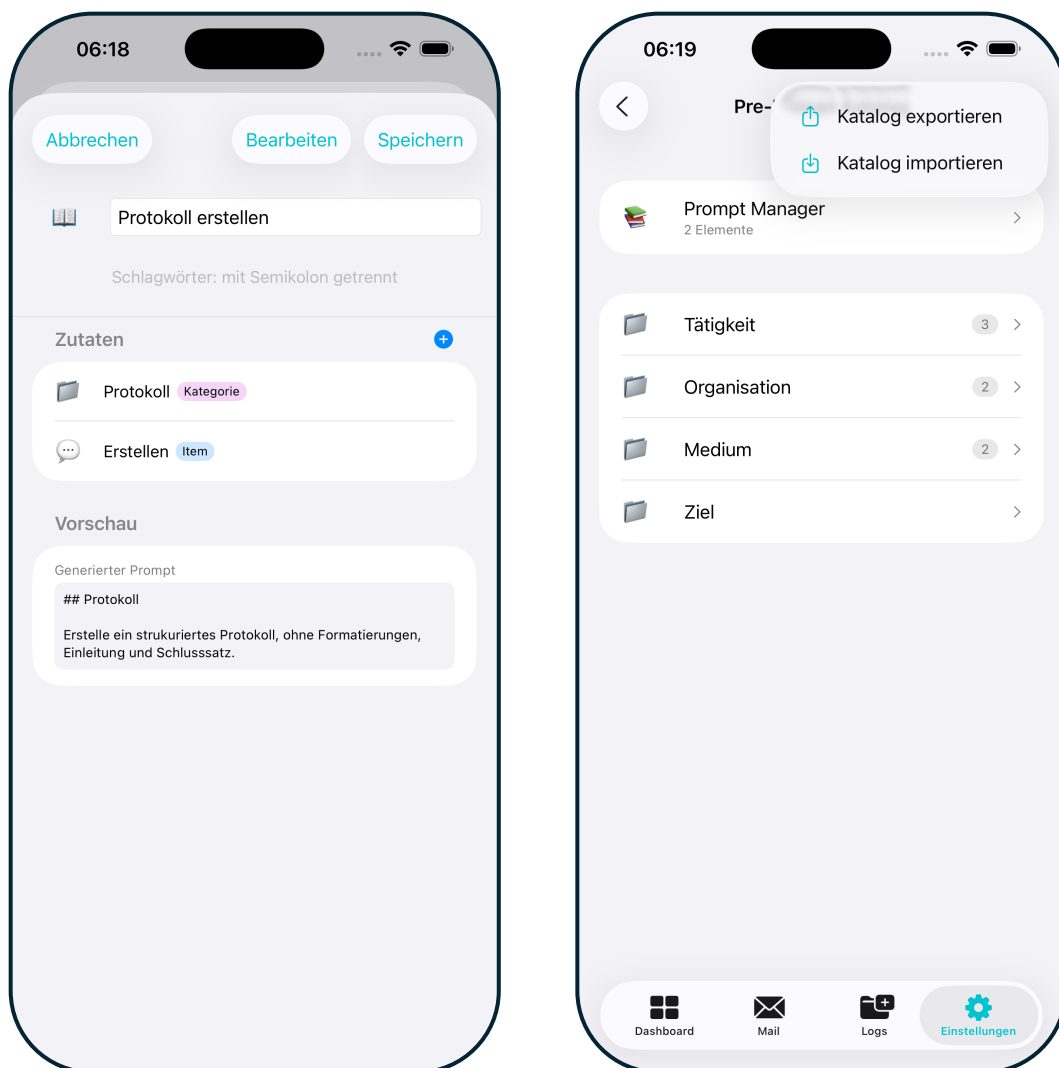
Bei der Erstinstallation werden automatisch folgende Kategorien angelegt:

-  **E-Mail:** Antworten, Weiterleiten, Analyse
-  **Notizen:** Allgemeine Textverarbeitung
-  **Protokoll:** Standard-Preset für Textstrukturierung

3.3 Rezept-System

Rezepte ermöglichen die Kombination mehrerer Pre-Prompts zu komplexen Anweisungsketten. Ein Rezept kann:

- Mehrere Kategorien referenzieren
- Schlagwörter kombinieren
- Einen generierten Prompt aus allen Elementen erzeugen

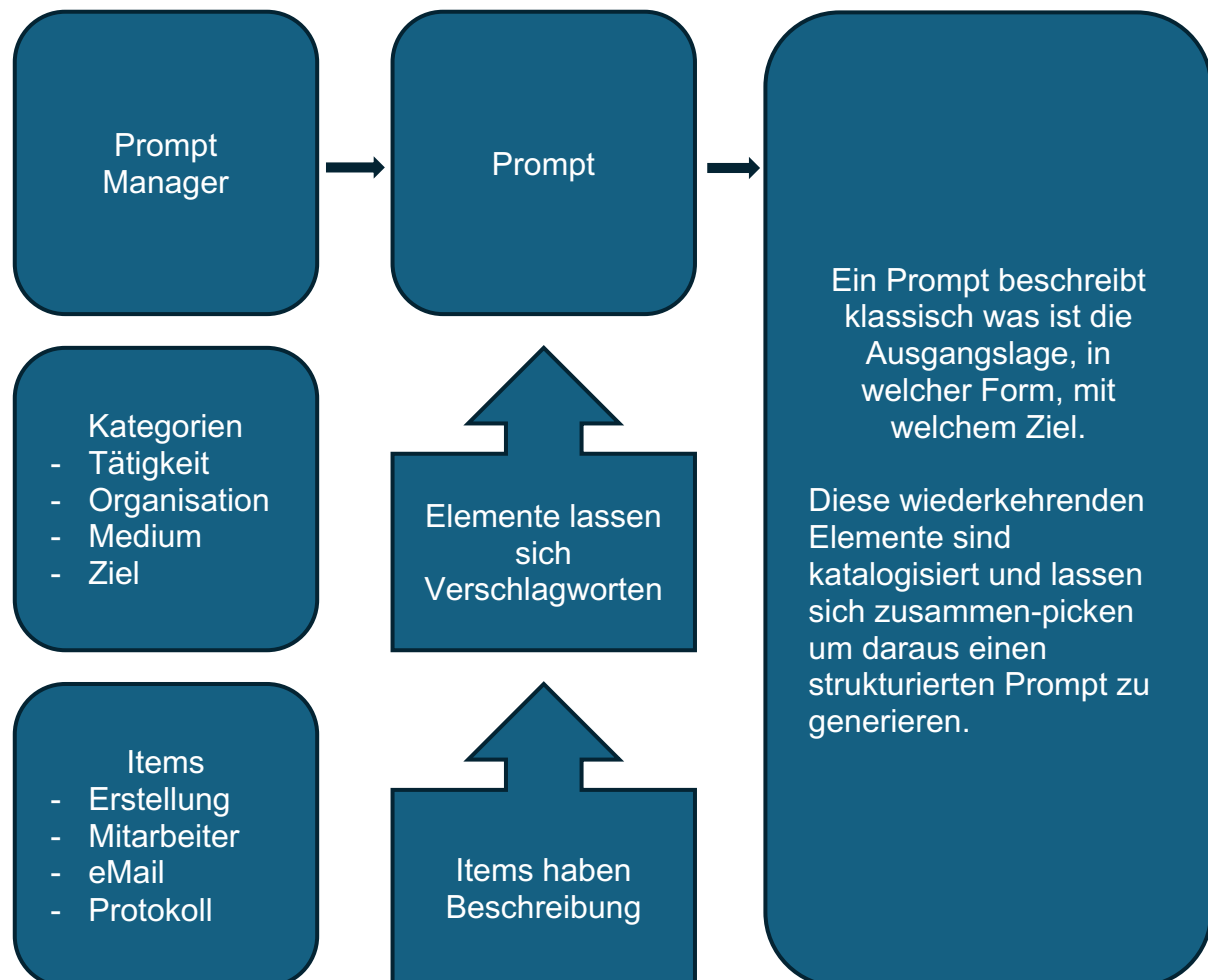


3.4 Export/Import Funktionalität

Der gesamte Katalog kann exportiert und importiert werden, um Konfigurationen zwischen Geräten zu teilen oder Backups zu erstellen.

3.5 Prompt Katalog Logik

Der Katalog ist so aufgebaut das auf 2 Ebenen damit gearbeitet werden kann, es gibt Kategorien und Items in der einen Ebene und den Prompt Manager mit den Prompts in der anderen Ebene. Das ganz ist vergleichbar eines Kochbuches, Kategorien und Items sind Arbeitsabläufe und Zutaten und der Prompt Manager das Kochbuch mit den Rezepten.



4. Log-System

Das Log-System ist das Herzstück der Datenverwaltung in AIILO. Es speichert alle Text- und Audio-Einträge zentral und ermöglicht deren Verarbeitung.

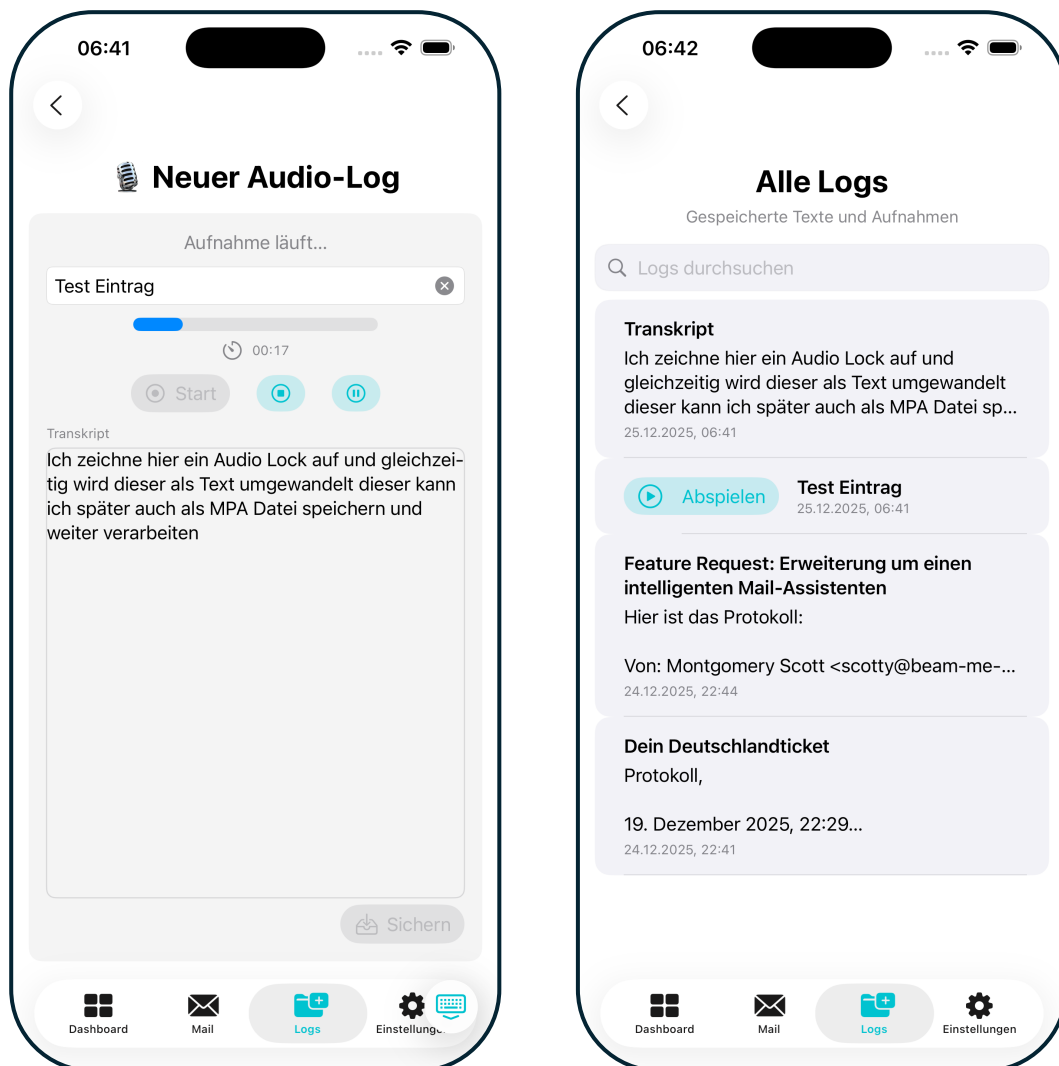
4.1 Datenmodell (LogEntry)

Jeder Log-Eintrag enthält folgende Eigenschaften:

- **id**: Eindeutige UUID
- **date**: Erstellungsdatum
- **type**: Text oder Audio
- **title**: Optionaler Titel
- **text**: Originaltext oder Transkription
- **aiText**: KI-überarbeiteter Text
- **category**: Kategoriezuordnung
- **tags**: Array von Schlagwörtern
- **reminderDate**: Optionale Erinnerung

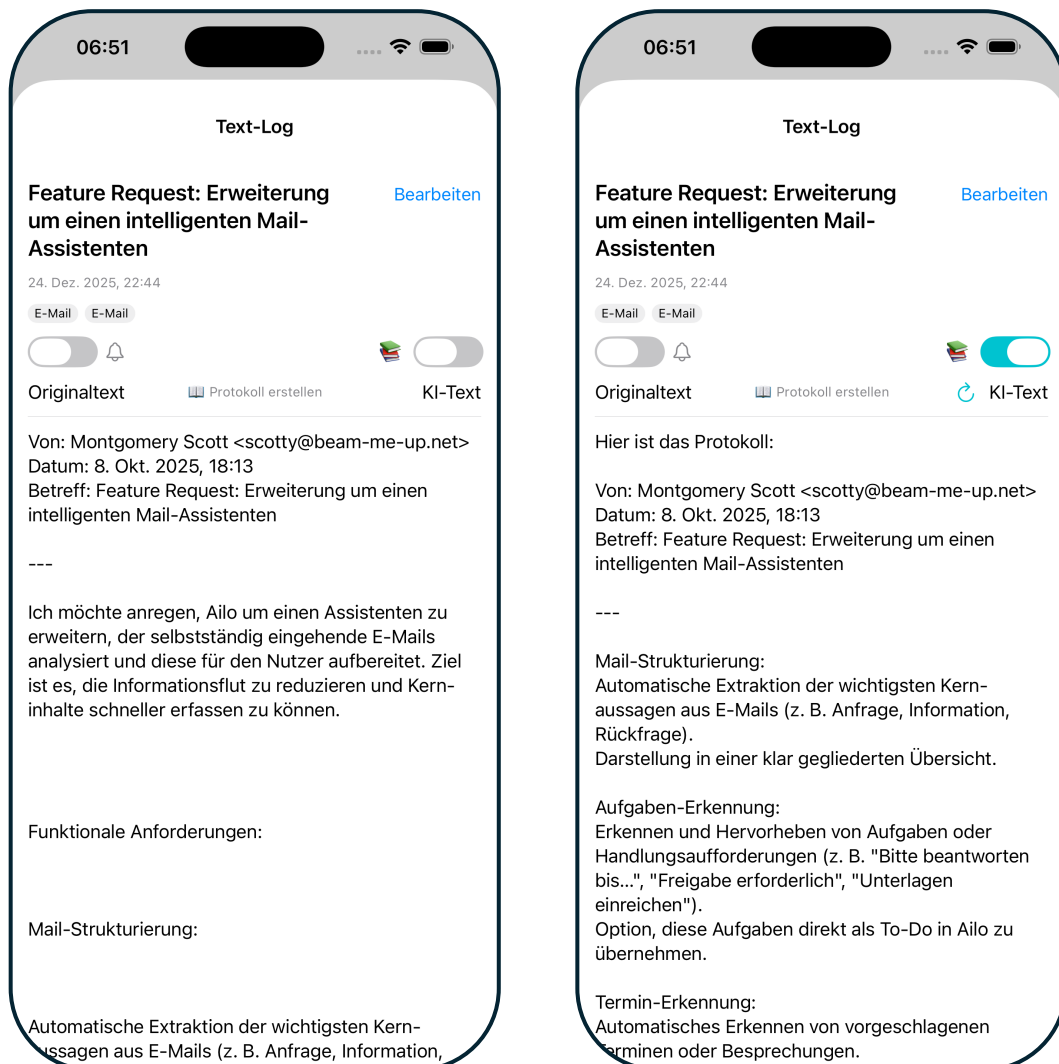
4.2 DataStore

Der DataStore verwaltet alle Log-Einträge persistent als JSON-Datei. Er bietet CRUD-Operationen und publiziert Änderungen via Combine für reaktive UI-Updates.



4.3 KI-Überarbeitung

Jeder Text-Log kann durch die KI überarbeitet werden. Der Originaltext bleibt dabei erhalten, während der KI-Text separat gespeichert wird. Der Benutzer kann zwischen Original und KI-Version umschalten.



5. E-Mail Integration

AILO bietet eine vollständige E-Mail-Integration mit IMAP/SMTP-Unterstützung. Die Implementierung folgt einer modernen, phasenbasierten Synchronisationsarchitektur.

5.1 Synchronisations-Architektur

Die Mail-Synchronisation ist in 5 Phasen unterteilt:

1. **Header-Only Sync:** Schnelle Übersicht durch minimalen Datentransfer
2. **Body-On-Demand:** Nachrichteninhalte werden bei Bedarf geladen
3. **Central-Processing:** Zentrale Verarbeitung aller E-Mail-Daten
4. **Structured-Storage:** SQLite-basierte Persistierung
5. **Bidirectional-Sync:** Zwei-Wege-Synchronisation mit dem Server

5.2 MailRepository

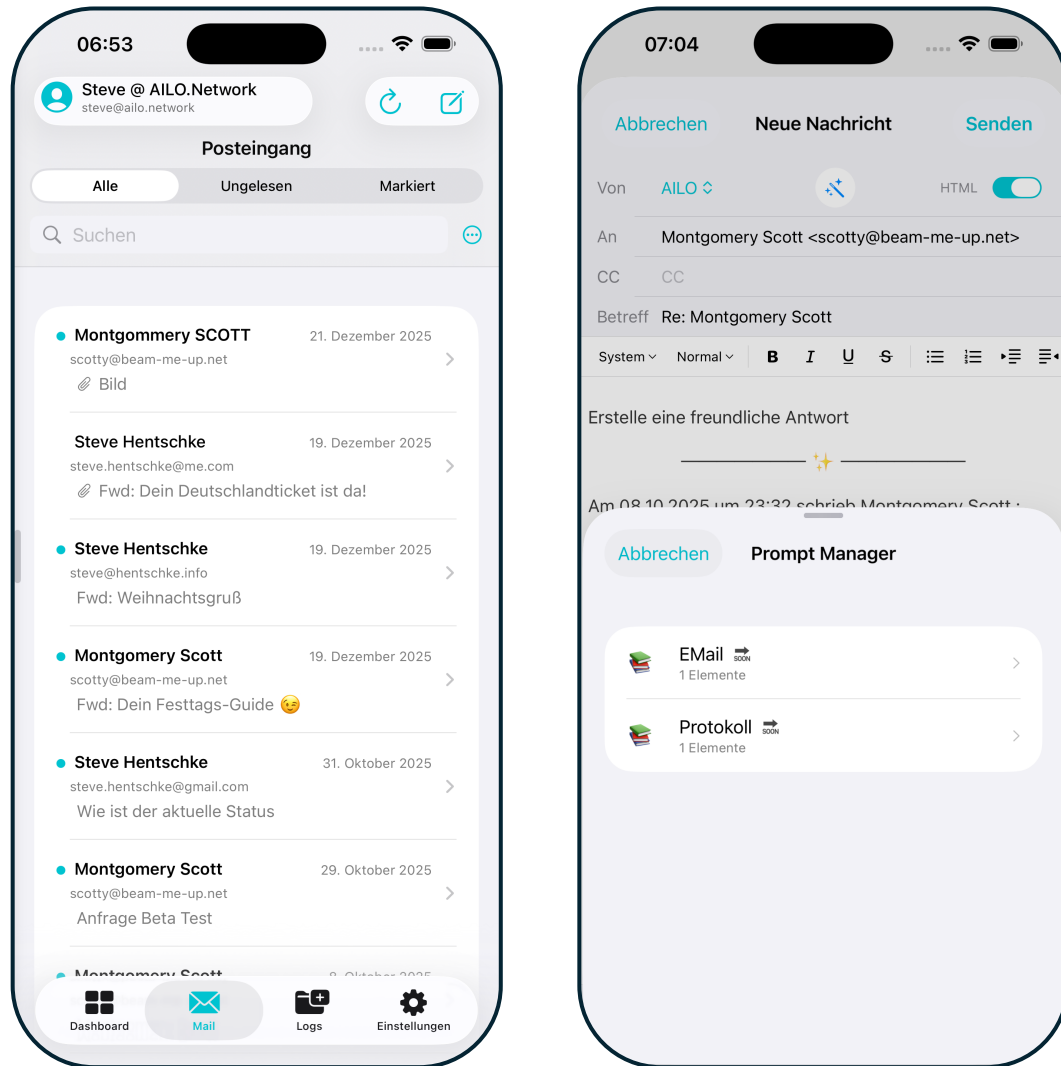
Der MailRepository ist die zentrale Schnittstelle für alle E-Mail-Operationen. Er koordiniert die Synchronisation, verwaltet den Cache und publiziert Änderungen an die UI.

Kernfunktionen:

- Initial Sync mit konfigurierbarem Limit
- Incremental Sync für neue Nachrichten
- Full Refresh Sync für vollständige Aktualisierung
- Folder Discovery und Special-Folder-Mapping

5.3 KI-Generator Antwort und Weiterleitungsmail

Der Prompt Manager findet hier seine zweite zentrale Anwendung: Als eine Art „Kochbuch“ wählt man eine E-Mail-Vorlage aus und erhält daraufhin verschiedene Antwortvarianten, die der eigenen Definition entsprechen. Die Kernaussage lässt sich bereits im Voraus formulieren. Beim Aufruf des Prompt Managers wird unter der eigenen Antwort eine „Magic-Line“ platziert. Die vorformulierte Aussage wird in den Prompt an das Sprachmodell integriert, während die Rahmenbedingungen der Antwort durch das „Rezept“ des Prompt Managers ergänzt werden.



5.4 IMAP-Implementierung

Die IMAP-Funktionalität wird durch eigene Parser und Commands realisiert:

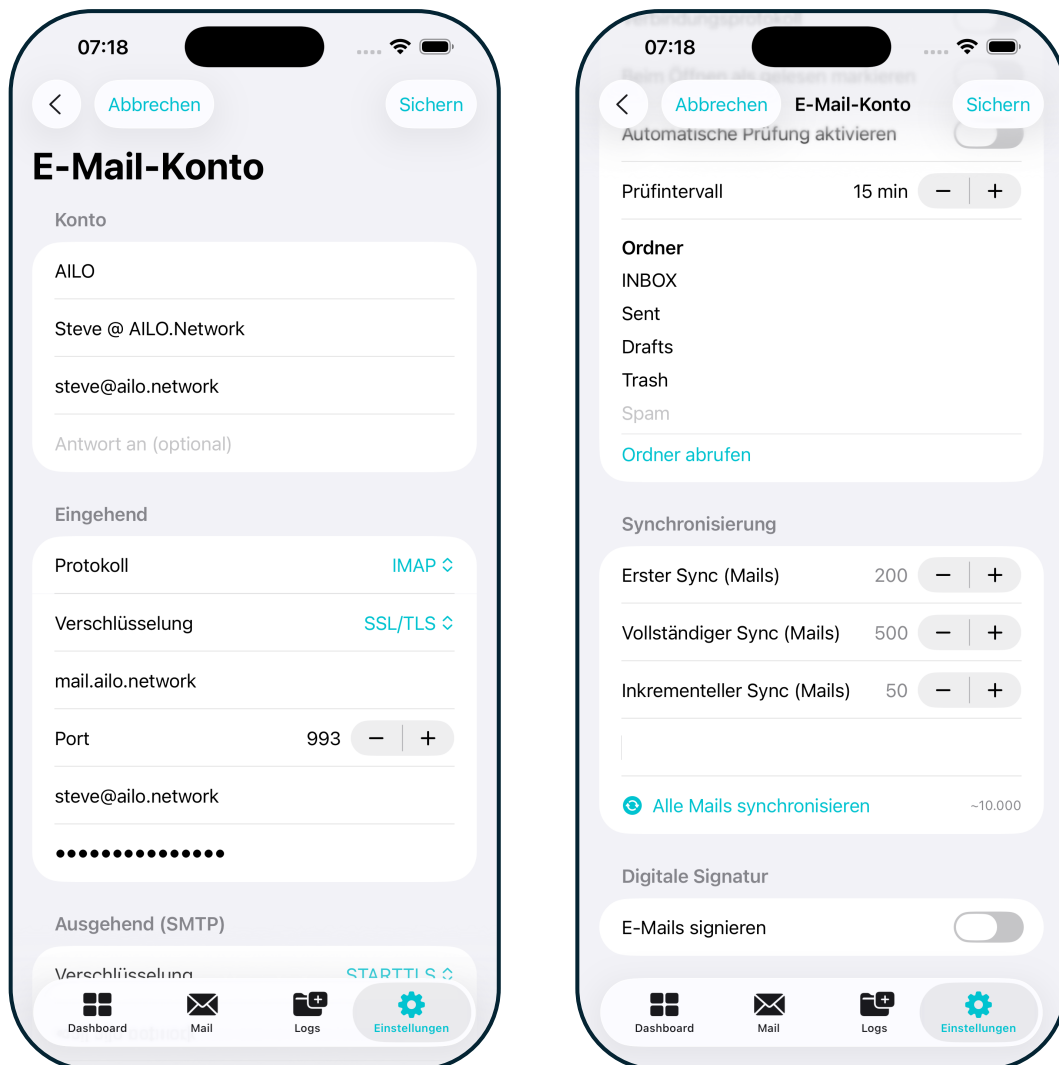
- **IMAPConnection:** Verbindungsmanagement mit TLS/StartTLS
- **IMAPCommands:** LOGIN, SELECT, FETCH, SEARCH, LIST
- **IMAPParsers:** Envelope, BODYSTRUCTURE, Flags Parsing

5.5 Konfigurierbare Sync-Limits

Anstatt hartcodierter Werte verwendet AIO konfigurierbare Sync-Limits pro Account:

- **syncLimitInitial:** Anzahl E-Mails bei Erstinstallation (Standard: 100)
- **syncLimitRefresh:** Anzahl E-Mails bei manuellem Refresh (Standard: 50)

Es lassen sich .p12 Zertifikatsdateien hinter jeden Account hinterlegen um digital signierte eMails zu versenden.



6. KI-Integration & Prompt Manager

Die KI-Integration ermöglicht die Anbindung verschiedener LLM-Provider für Textverarbeitung und -verbesserung.

6.1 Unterstützte Provider

AIILO unterstützt mehrere KI-Provider:

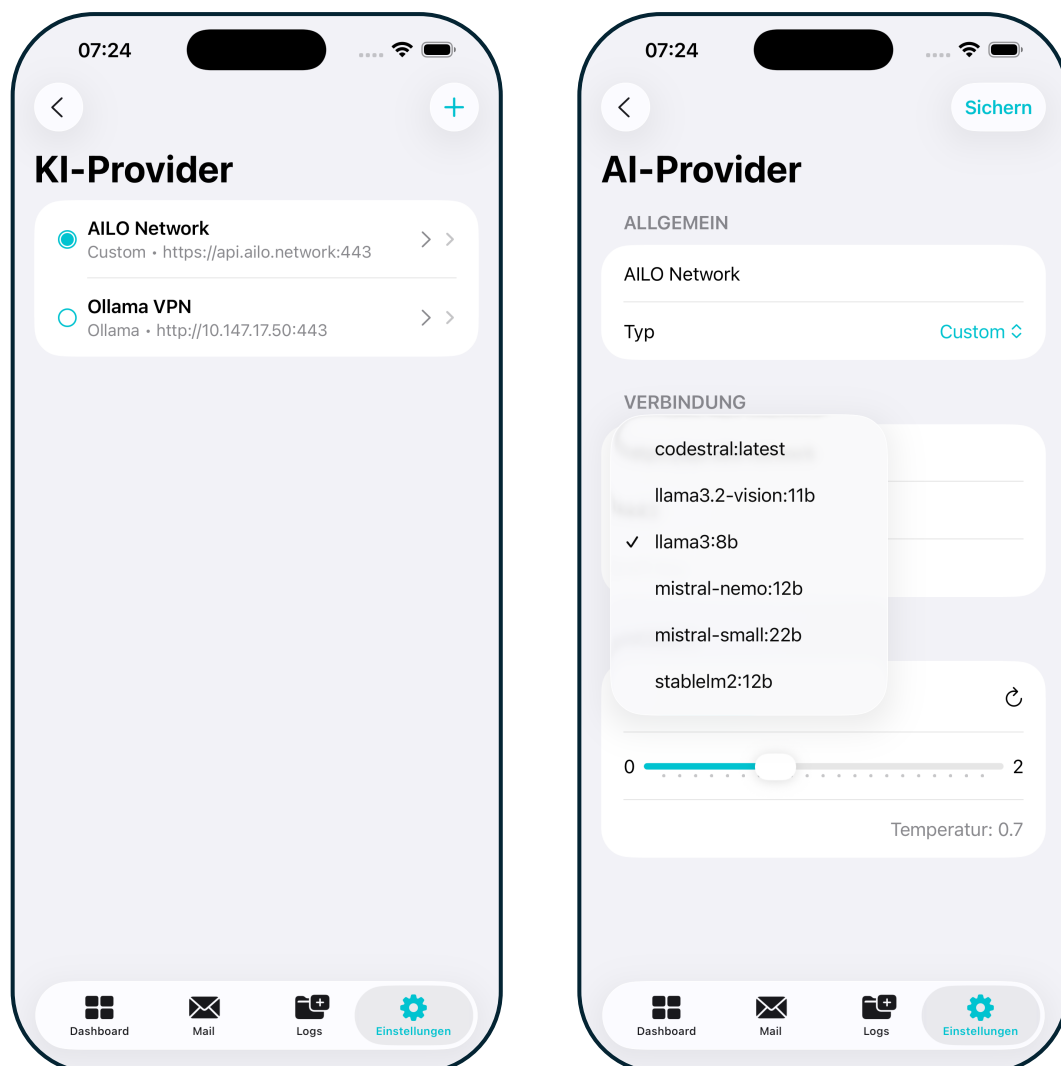
- **OpenAI:** GPT-4, GPT-3.5 via /v1/chat/completions API
- **Mistral:** Mistral Large via /v1/chat/completions API
- **Ollama:** Lokale LLMs via /api/chat und /api/generate
- **Custom:** Beliebige OpenAI-kompatible Endpoints

6.2 AIClient Implementierung

Der AIClient ist als zentrale HTTP-Schnittstelle implementiert und erkennt automatisch den Provider-Typ anhand der API-Struktur. Bei fehlgeschlagenen OpenAI-Anfragen erfolgt ein automatischer Fallback auf Ollama-Endpoints.

Workflow einer KI-Anfrage:

1. Provider-Konfiguration aus UserDefaults laden
2. Pre-Prompt aus Katalog oder Rezept auflösen
3. OpenAI-kompatible Anfrage versuchen
4. Bei Fehler: Fallback auf Ollama-Format
5. Antwort parsen und an UI zurückgeben

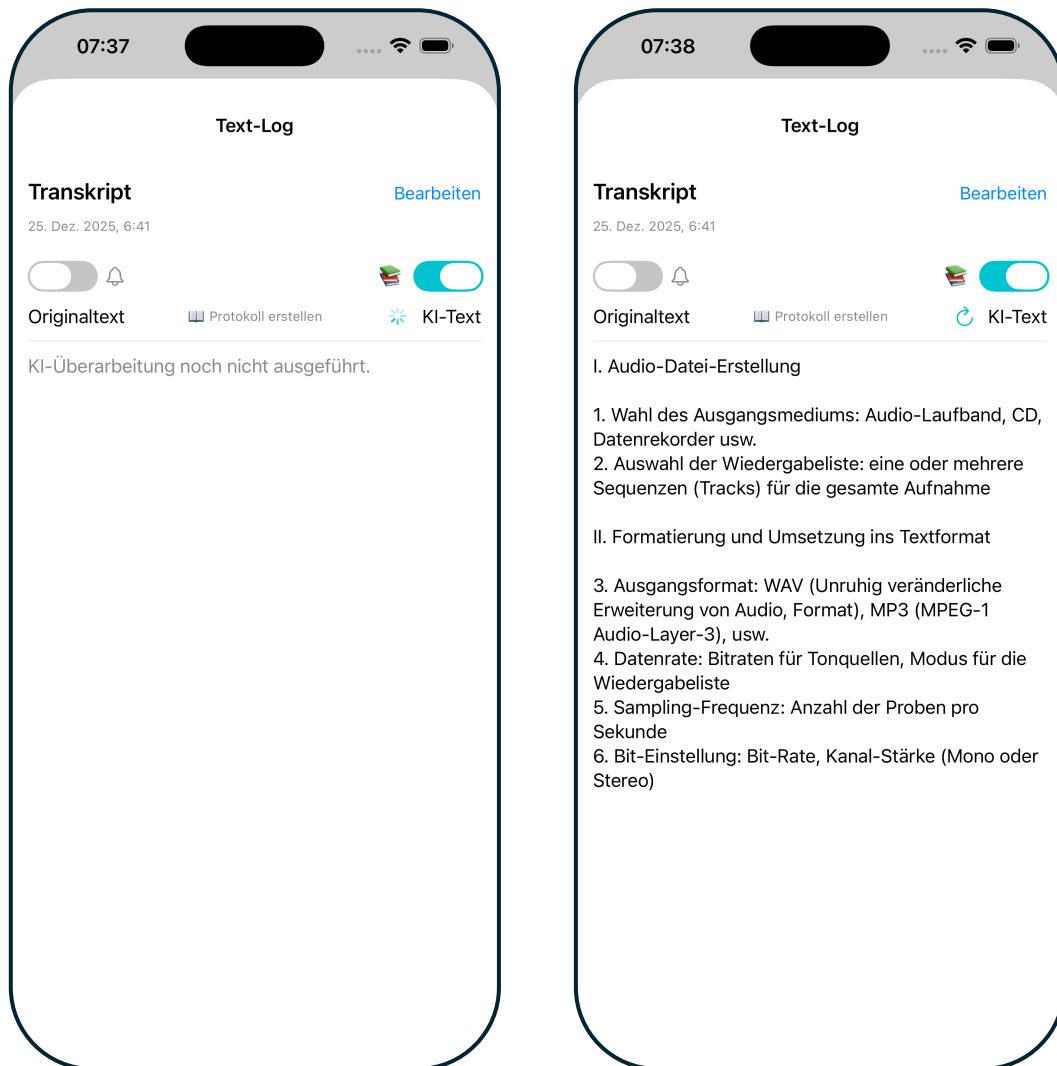


6.3 Pre-Prompt Integration in TextLogDetailView

In der Log-Detailansicht können Benutzer Pre-Prompts aus dem Katalog oder Rezepte auswählen, bevor sie die KI-Überarbeitung starten. Der ausgewählte Prompt wird mit dem Originaltext kombiniert und an den aktiven Provider gesendet.

Integration:

- Preset-Auswahl via PrePromptPicker
- Rezept-Auswahl für komplexe Anweisungen
- Fallback auf globalen Pre-Prompt wenn kein spezifischer ausgewählt
- Reminder für Dashboard



7. Architektur-Übersicht

7.1 Schichtenmodell

Views, ViewModels, Navigation	UI Layer (SwiftUI)
AIClient, MailRepository, DataStore	Service Layer
DAOs, SQLite, UserDefaults, Keychain	Data Layer
IMAP/SMTP, HTTP, URLSession	Network Layer

7.2 Lokalisierung

AIILO unterstützt vollständige Lokalisierung in Deutsch (primär) und Englisch. Alle UI-Strings sind in Localizable.strings ausgelagert und folgen dem Muster "app.feature.key".

7.3 Datensicherheit

- API-Keys und Passwörter im Keychain verschlüsselt
- Lokale Datenspeicherung ohne Cloud-Synchronisation
- Audio-Dateien verbleiben auf dem Gerät

— Ende der Dokumentation —