

## ON-DEVICE-KI IN DER PRAXIS

# Referenzprojekt

## On-Device-KI in KM Geld

KI-gestützte Fahrterfassung und lokaler Hilfe-Assistent - entwickelt, integriert, getestet und als Mobile-App-Release umgesetzt.

### Projektprofil

**PROJEKT**

KM Geld - Fahrtenbuch & Kilometergeld

**ZEITRAUM**

Juni bis Juli 2026

**VERANTWORTUNG**

Klemens Zlepntig - End-to-end

**KONTEXT**

Eigenprodukt der creative workline GmbH

**STATUS**

Veröffentlicht (1.2.1)

**PLATTFORMEN**

Android und iOS

### Ausgangslage

KM Geld sollte berufliche Fahrten schneller erfassbar machen, ohne Offline-first und No-Account aufzugeben. Freitext sollte zu strukturierten Vorschlägen werden, ohne dass KI eigenständig Daten übernimmt oder speichert.

### Realisierte KI-Funktionen

- Fahrterfassung: Freitext oder Diktat wird lokal in sechs prüfbare Felder strukturiert.
- Hilfe: Lokales Retrieval und On-Device-Generierung antworten auf Basis der App-Dokumentation.

**END-TO-END-VERANTWORTUNG**

Use-Case-Auswahl und Scope, UX und Review-Flow, KMP-Architektur, Android- und iOS-Integration, Prompting und strukturierte Ausgabe, lokales Retrieval, Datenschutzkonzept, Testautomatisierung und Release.

**ON-DEVICE****REVIEW-FIRST****ANDROID + IOS****OHNE BACKEND**

## KI-FUNKTION 1

# Freitext oder Diktat wird zum geprüften Fahrtenentwurf

Eine Fahrt wird in Alltagssprache beschrieben oder diktiert. Die lokale KI extrahiert Datum, Start, Ziel, Zweck, Fahrtart und Kilometer. Erst nach sichtbarer Prüfung gelangen ausgewählte Felder in das Formular; Speichern bleibt eine eigene Aktion.



**1. EINGABE**

Freitext oder lokale Spracheingabe. Das Mikrofon startet nur nach bewusstem Tippen.

**2. ERKENNEN**

Die Verarbeitung startet explizit. Beschreibung und Diktat bleiben editierbar und lokal.

**3. PRÜFEN**

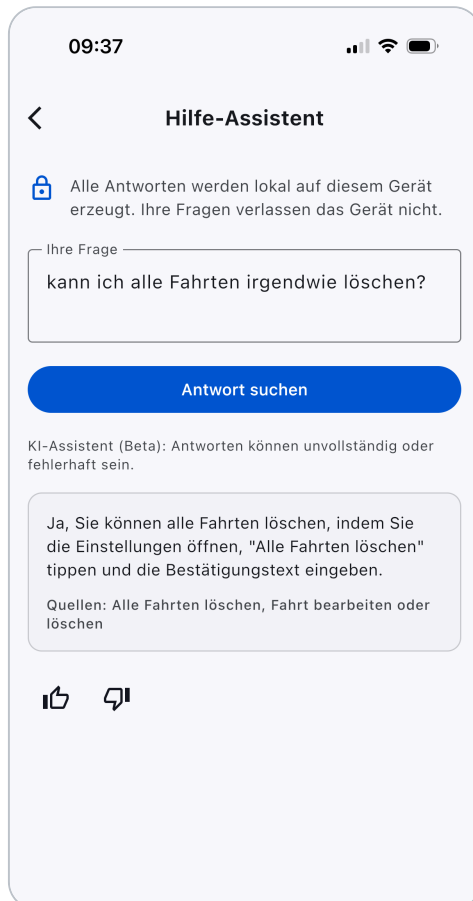
Sechs Felder sind einzeln prüfbar, abwählbar und erst danach übernehmbar.

## Technische Umsetzung

- Android: ML Kit GenAI Prompt API mit Gemini Nano über AICore.
- iOS: Apple Foundation Models mit typisierter Ausgabe.
- Shared: KMP-Use-Case, ViewModel, UI-State und Validierung.
- Fallback: Deterministischer lokaler Parser bei fehlendem Modell.
- Safety: Keine automatische Übernahme oder Speicherung.
- Nutzen: Weniger Formulareingabe bei voller Kontrolle.

## KI-FUNKTION 2

# Lokaler Hilfe-Assistent mit eigener Wissensbasis



## RAG vollständig auf dem Gerät

Der Assistent beantwortet Fragen anhand mitgelieferter Hilfeartikel. Lokale Suche und On-Device-Modell erzeugen eine kurze Antwort und zeigen die verwendeten Quellen direkt an.

1

**Frage**

Eingabe bleibt lokal

2

**Retrieval**

BM25 sucht Artikel

3

**Generierung**Nano oder  
Foundation Models

4

**Antwort**

Kurz und mit Quellen

## Datenschutz und sichere Degradation

- Kein Backend, Login, Tracking oder Analytics.
- Ohne unterstütztes Modell öffnet die lokale Artikelsuche.
- Lokaler Kontext, Quellen, Beta-Hinweis und Feedback begrenzen Fehlantworten.

## Qualitätssicherung

- Unit-Tests für Parser, Retrieval, Prompts und ViewModels.
- Golden Questions und Fallback-Tests für die Wissensbasis.
- Android-UI- und iOS-Simulator-Tests der kritischen Abläufe.

## Beratungskompetenz

- KI-Use-Cases aus realen Geschäftsprozessen ableiten.
- Modell, lokale Datenbasis, UX und Fachlogik integrieren.
- Datenschutz, Fallbacks, Tests und Release mitverantworten.

## KM GELD IN DEN APP STORES

Bei Google Play ansehen

[play.google.com](https://play.google.com) | Android

Im App Store ansehen

[apps.apple.com](https://apps.apple.com) | iPhone und iPadKlemens Zleptnig | creative workline GmbH | [office@creativeworkline.at](mailto:office@creativeworkline.at) | [creativeworkline.at](https://creativeworkline.at)

Dokumentversion 1.0 | Stand: 29.07.2026