DATENSCHUTZ-PRÜFUNG · QUELTEL GMBH

SeniorPDF – Programme, Module & Datenfluss

Projekt `pdf_reader` · Version 1.2.1+13 · Framework: Flutter (Dart) · Stand: 05.10.2026

Bestätigt: kein Tracking – geprüft auf Modul-, Code-, Berechtigungs- und Netzwerk-Ebene.

Alle 131 Module stammen aus dem offiziellen Dart-Register (pub.dev). Kein Firebase/Analytics/Crashlytics, keine Werbe- oder Tracking-SDKs, keine App-Tracking-Erlaubnis (iOS ATT), keine fremden Server im Code. Nach außen geht nur der **In-App-Kauf** (App Store / Google Play) – sonst nichts.

Ein unkritischer Aufräum-Hinweis (Kamera-/Standort-Textbausteine auf iOS) steht unten.

1 · Vertiefte Prüfung (100 %)

GEPRÜFT	ERGEBNIS
Firebase / Google-Analytics-Konfig (`google-services.json` , `GoogleService-Info.plist`)	**Nicht vorhanden → kein Firebase/Analytics/Crashlytics**
iOS App-Tracking (`NSUserTrackingUsageDescription`) & Werbe-Netzwerke (`SKAdNetwork`)	**Nicht vorhanden → kein IDFA-Tracking, keine Ad-Netzwerke**
iOS Transport-Sicherheit (`NSAllowsArbitraryLoads`)	**Nicht aufgeweicht**
macOS Sandbox & Netzwerk-Rechte	**Sandbox aktiv; nur `network.server` , kein `network.client` → App sendet von sich aus nichts nach außen**
Server/Endpunkte im App-Code (`lib/`)	**Nur eigene Seiten: seniorpdf.de/agb, /datenschutz, /impressum**
Netzwerk-Bibliotheken im App-Code (http, dio, HttpClient, Socket)	**Keine · `dart:io` nur für Datei-Zugriff & OS-Erkennung**

GEPRÜFT	ERGEBNIS
Android: selbst gesetzte Berechtigungen & native Fremd-SDKs	**Keine eigenen Berechtigungen, keine fremden SDKs im `build.gradle`**

2 · Die 13 aktiven Module der App

MODUL	ZWECK	HERKUNFT	NACH AUSSEN?
flutter + flutter_localizations	App-Grundgerüst & Mehrsprachigkeit	Flutter-Team (Google, USA)	**Nein**
pdfrx	PDF anzeigen/öffnen, auch passwortgeschützt (PDFium)	Open Source · PDFium = Google	**Nein · lokal**
printing	Drucken & PDF-Aufbereitung	Open Source (D. Pham-Van)	**Nur zum Drucker**
flutter_tts	Vorlesen (Text-to-Speech)	Open Source · System-Stimme	**Nein · OS**
file_picker	Datei-Auswahldialog	Open Source (M. Ruivo)	**Nein**
share_plus	Teilen über System-Menü	Flutter-Community	**Nein**
path_provider	Speicherorte finden	Flutter-Team (Google)	**Nein**
crypto	Prüfsummen/Verschlüsselung (Passwort-PDFs)	Dart-Team (Google)	**Nein**
url_launcher	Links & E-Mail öffnen	Flutter-Team (Google)	**Nur wenn Sie tippen**
package_info_plus	App-Version anzeigen	Flutter-Community	**Nein**
intl	Datums-/Zahlenformate	Dart-Team (Google)	**Nein**

MODUL	ZWECK	HERKUNFT	NACH AUSSEN?
cupertino_icons	Symbole/Icons	Flutter-Team (Google)	**Nein**
in_app_purchase	Vollversion kaufen	Flutter-Team (Google)	**Ja → Store (nur Kauf)**

Die weiteren 118 Einträge sind **Unter-Bausteine** dieser Module oder reine **Entwickler-Werkzeuge** (`flutter_test` , `flutter_lints` , `leak_tracker` …), die **nicht in die fertige App** gelangen – ebenfalls ohne Tracking.

3 · Empfehlung zum Aufräumen (kein Tracking)

Die iOS-Konfiguration enthält Nutzungstexte für **Kamera** (`NSCameraUsageDescription`) und **Standort** (`NSLocationWhenInUseUsageDescription`). Das **sendet keine Daten** – es erlaubt nur, dass die App theoretisch danach fragen dürfte. Für einen reinen PDF-Reader wirkt das überflüssig und kann im App-Store-Review Rückfragen auslösen.

Empfehlung: prüfen, ob Kamera/Standort überhaupt genutzt werden – falls nein, die Einträge aus `ios/Runner/Info.plist` (und ggf. `macos`) entfernen. Dann fragt die App gar nicht erst danach. (Foto-Zugriff ist für PDF-Import sinnvoll und kann bleiben.)

4 · Einordnung „ausländisch" & Absturzberichte

- Framework (Flutter), Stores und einige Basis-Pakete sind von **Google/Apple (USA)** – bei Apps praktisch unvermeidbar; Community-Pakete sind international und quelloffen. Keines sammelt oder trackt Daten.
- **Absturzberichte** laufen über die Store-Mechanik von Apple/Google (anonym, ohne personenbezogene Daten). Es ist **kein** Crash-SDK in der App – deckt sich mit der Darstellung auf seniorapps.de.
- **Einziger Außenkontakt:** der In-App-Kauf zum Store, ausschließlich zum Verkauf/zur Prüfung der Vollversion.

Grundlage: `pdf_reader` (pubspec.yaml/lock, 131 Pakete – 126× pub.dev, 5× Flutter/Dart-SDK, keine Fremdquellen) sowie AndroidManifest, iOS/macOS-Konfiguration und App-Quellcode (`lib/`).

Momentaufnahme der aktuellen Versionen; bei künftigen Updates erneut prüfen. Für eine gerichtsfeste Bestätigung zusätzlich eine einmalige Netzwerk-Aufzeichnung (z. B. Proxyman/Charles) beim Testlauf.