Security Disclosure Report

fr.doctolib.www (Android)

Pre-consent SDK start, exported provider, Firebase configuration in the production build

PUBLISHED 2 NOVEMBER 2026 – ONE FIX VERIFIED, ONE FIX PARTLY VERIFIED, ONE FINDING DISPUTED

Package	fr.doctolib.www
Versions analysed	5.8.1 (versionCode 320096), 5.17.0 (versionCode 322453)
Operator	Doctolib SAS (France) and Doctolib GmbH (Germany)
Lead supervisory authority	CNIL (France), Art. 56 GDPR
Notified	4 August 2026
Publication	2 November 2026
Vendor responses	6 Aug, 18 Aug, 19 Aug, 10 Sep and 2 Nov 2026
SHA-256 (base APK)	`5.8.1 base.apk: 2d7ac65e5060d49ddbdaafcad5cc31467d89169495b0fd581484077b7c0669b5 · 5.17.0 base.apk: 7cccf0f951824ed4f92bfa3e7516f25a77f4c41ffadc40febd195053433f9dfb`

security@rfi-irfos.com • rfi-irfos.com

ZVR 1015608684 • GISA 39261441 • UID ATU83405245 • Steuernummer 68 696/8736 • Graz, Austria

Contents

1 Executive Summary 3

2 Subject and Operator Analysis 4

3 Architecture and Data Flow 4

4 Findings 5
4.1 H1: Exported content provider for clipboard files (ClipboardFileProvider) 5
4.2 H2: Screeb initialised at process start, before the application's consent layer . 6
4.3 H3: SDK inventory in the first report (corrected) 7
4.4 H4: Firebase configuration with API key in the production build 7

5 Impact Analysis 8

6 The Complete Chain 8

7 Health Data and Consumer Protection 9

8 Data Protection, Article by Article 9

9 Platform Accountability and the DSA 9

10 Regulatory and Legal Landscape 10

11 Indicators of Compromise and Reproducible Evidence 10

12 Disclosure Timeline and Outcome 11

13 Residual Risk and Recommendations 11

Executive Summary

Doctolib's Android app for booking appointments and managing health information starts a third-party session replay and in-app analytics SDK, Screeb, through an Android content provider. That provider runs before the app's first screen and before the app's own consent layer. We reported this on 4 August 2026, together with three further findings, to the company's data protection officer and to the supervisory authorities.

Doctolib confirmed the start-up order on 10 September 2026. On 2 November 2026 it reported that Screeb 4.4.1 has been live in the Play Store since mid-October and that the clipboard provider exposure has been removed. We verified the clipboard fix in the 5.17.0 build. We verified part of the Screeb change. The native provider still runs without a consent check, but in the native code nothing starts the SDK's web view unless a channel identifier is set. Whether the application's JavaScript layer waits for consent could not be checked, because the React Native bundle is compiled Hermes bytecode.

Doctolib disputes the Firebase finding. It states that the Firebase API key is a project identifier and that server-side controls are in place. We confirm that the key is present in the 5.17.0 release. We did not test the server-side controls.

Our first report named fourteen SDK families. Nine of those names are not in the app. We corrected the list, and this report uses only the families we verified in the binaries.

The thesis of this report is narrow. A health booking app should not start third-party analytics before the user has been asked. The vendor accepts that the start-up order existed in the version we analysed. The evidence in this report does not yet settle whether the current version complies with that rule.

ID	Severity	Title
H1	HIGH	Exported content provider for clipboard files (ClipboardFileProvider)
H2	CRITICAL	Screeb initialised at process start, before the application's consent layer
H3	LOW	SDK inventory in the first report (corrected)
H4	MEDIUM	Firebase configuration with API key in the production build

Subject and Operator Analysis

The operator is Doctolib SAS, with Doctolib GmbH as the German entity. The app provides appointment booking and health-related information for patients and practitioners in France and Germany. The app is distributed through Google Play. We have no information about the ownership structure beyond what the vendor publishes.

Attribute	Value
Package	fr.doctolib.www
Framework	React Native with Expo, native Android layer
Operator	Doctolib SAS (France), Doctolib GmbH (Germany)
Lead supervisory authority	CNIL (France), main establishment in France
Distribution	Google Play Store

Architecture and Data Flow

The app combines a React Native user interface with a native Android layer. Several third-party SDKs are registered in the native layer. Their start-up behaviour matters because Android initialises content providers before the first Activity. The table lists the components relevant to this report and their status in the two builds analysed.

Component	Role	5.8.1 (analysed in August)	5.17.0 (analysed 2 November)
ScreebInitProvider (app.screeb.sdk)	Session replay and in-app UX analytics	Registered, initOrder 2147483647, calls initSdk WithContextOnly	Still registered, same call, no consent check in the native path
Didomi	Consent management (CMP)	Present, consent layer	Present
Firebase App and Messaging	Push notifications	FirebaseInitProvider, initOrder 100	Present, Doctolib says the change reaches the Play Store on 16 Nov 2026
Datadog	Real-user monitoring	Present	Present, Doctolib says RUM starts with consent pending
Adjust	Attribution and advertising measurement	Present	Present
Sentry	Crash reporting	Present	Present, Doctolib says session replay is not enabled
ML Kit (barcode)	Barcode scanning	Present, MlKitInitProvider	Absent

Findings

H1: Exported content provider for clipboard files (ClipboardFileProvider)

Fixed, verified in the binary

In 5.8.1 the provider was exported without a permission guard. In 5.17.0 it is not exported.

The component is expo.modules.clipboard.ClipboardFileProvider. Doctolib states that the component comes from the expo-clipboard library, whose manifest declared exported=true in the shipped version, and that the upstream default has since changed.

Build	exported	Permission guard	Source
5.8.1	true	none	aapt2 xmltree on the manifest
5.17.0	false	not needed	aapt2 xmltree on the manifest

Impact: Other apps on the same device could address the provider while the 5.8.1 build was installed. We did not test that access in practice.

Fix: Keep exported=false in every release. Verify the manifest of each build before shipping.

H2: Screeb initialised at process start, before the application's consent layer

Start-up order confirmed by vendor, partly verified

The native path still runs without a consent check. Whether the JavaScript layer waits for consent is not verified.

The code path in the 5.8.1 build is shown below. The 5.17.0 build keeps the same provider and the same call.

```
ScreebInitProvider.onCreate()
  -> Screeb.INSTANCE.initSdkWithContextOnly(applicationContext)
       -> posts initSdkWithContextOnly$lambda$8 on the main thread
            -> if channelId != null: initSdk(...)   (deferred)
```

In the 5.8.1 build the SDK defines the telemetry endpoint https://r.screeb.app/rpc/1.3.0/l?source=sdk-android and a web view at https://t.screeb.app/webview.html?version=4. In the 5.17.0 build the web view URL carries version=6. We did not capture network traffic, so we cannot say which requests left the device at start.

Doctolib confirmed on 10 September 2026 that Screeb initialises at process start, ahead of any consent decision, and that it is remediating this. On 2 November 2026 it stated that Screeb 4.4.1 no longer loads its web view or sends telemetry at process start, and that its application-level initialisation depends on consent.

Our verification on 5.17.0

The provider is still registered with the maximum init order and still calls initSdkWithContextOnly. The native code sets no channel identifier, so the web view does not start from native code. The build contains the string 'Screeb android 4.4.0', while Doctolib names version 4.4.1. We could not read the JavaScript bundle to confirm where the application calls initSdk.

Impact: For users, the risk lies in the order: the SDK can record events before the user has been asked. Whether it did so in the 5.8.1 build is not measured in this report.

Fix: Gate the SDK start on consent in both the native provider and the JavaScript layer. Publish the Screeb version in release notes. Add a runtime test that checks outbound traffic before the first consent decision.

H3: SDK inventory in the first report (corrected)

Corrected

The first report named fourteen SDK families. Nine of them are absent from both builds.

Name in first report	5.8.1	5.17.0
AppsFlyer, CleverTap, OneSignal, Airship, Amplitude, Mixpanel, Segment, Branch, Heap	absent	absent
Firebase Messaging, Datadog, Adjust, Sentry, Screeb, Didomi	present	present
ML Kit	present	absent

Impact: The inventory error did not change the H2 finding. It did overstate the number of SDKs in one health binary.

Fix: Derive SDK inventories from the dependency tree and the merged manifest, not from class-name searches alone.

H4: Firebase configuration with API key in the production build

Disputed by the vendor

The key and project identifiers are present in both builds. The vendor states that server-side controls protect the project.

Resource	Value (partly redacted)	5.8.1	5.17.0
google_api_key	AIza… (value withheld)	present	present
google_app_id	1:203254526936:android:…	present	present
project_id	doctolib-dabf0	present	present

Doctolib's position on 2 November 2026: Firebase API keys identify the project and are not secrets, the email and anonymous sign-in providers are disabled, and data access is governed by security rules. No rotation is planned.

Our position: the keys are present in the production build, which the vendor confirms. We did not test the server-side configuration, and we cannot test it from outside the project. We therefore do not classify this as a vulnerability in this report.

Impact: Impact depends entirely on the server-side configuration, which is not verified here.

Fix: Publish the security rules and the list of enabled sign-in providers, or give an independent test window. Restrict the key to the app's package name and signing certificate.

Impact Analysis

The affected population is the users of the app. The vendor describes its user base as France and Germany. The Complete Chain below describes the path from installation to any data leaving the device. We did not measure what data the SDK sends, so this section describes possible impact, not confirmed impact.

- App usage events and screen interactions, which can reveal that a user is looking for a specialist or a health service.
- Device and advertising identifiers. The manifest declares AD_ID, and we did not verify its use.
- Crash data through Sentry, which the vendor says carries no session replay.
- No health record content was observed in the code paths analysed. That is not the same as proof that none is sent.

The Complete Chain

This is the spine of the report. Each later section refers to one of these steps.

- Step 1: The user installs the app from Google Play. In the 5.8.1 build, Android creates the app's content providers before any Activity starts.
- Step 2: ScreebInitProvider has the maximum init order Android accepts, so it runs before MainActivity and before the Didomi consent screen can appear.
- Step 3: ScreebInitProvider.onCreate calls the SDK init path without a consent check. Vendor: fixed in 4.4.1. Our check of 5.17.0: still present in the native path.
- Step 4: The SDK defines a web view and telemetry endpoint. Whether traffic left the device at start was not measured. Vendor: no web view and no telemetry at process start in 4.4.1. Our check of 5.17.0: no native web view start without a channel identifier. The JavaScript gate is unverified.
- Step 5: The consent layer appears only after MainActivity starts. Any event recorded before the user answers is already inside the SDK's pipeline.
- Step 6: The user has no means to withdraw an event that has already left the device. This is the point where the harm lies, and it is conditional on steps 3 and 4 having operated as described.

Health Data and Consumer Protection

A booking app for health services is a place where users reveal health context through their navigation. The question under Art. 9 GDPR arises only if health data reaches the third party. We did not establish that. Consumer protection is engaged by the order of events: a user who has not been asked cannot refuse processing that has already begun.

Data Protection, Article by Article

- Art. 5(1)(b) purpose limitation: not tested. It depends on what the SDK does with data after collection, which requires the vendor's contract and configuration.
- Art. 6(1)(a) and Art. 7(1) consent: engaged by H2 if analytics data is collected before consent. The vendor accepts the start-up order for 5.8.1.
- Art. 9(2)(a) explicit consent: engaged only if health data reaches the SDK. Not established.
- Art. 25 data protection by design: engaged by the exported provider (H1) and the start-up order (H2), in our assessment.
- Art. 32 security of processing: engaged by the Firebase configuration (H4), depending on the server-side controls, which we did not test.
- Art. 33 and 34 breach notification: the vendor states that it has not identified a breach. This report takes no position on notifiability.
- ePrivacy Directive Art. 5(3): storage of and access to information on the user's device before consent. Engaged by the SDK start in 5.8.1.

Platform Accountability and the DSA

Google Play distributes the app. We did not analyse Google's obligations under the Digital Services Act or any other rule, and this report makes no claim about them. The distribution channel matters only because the 5.17.0 build reached users through it.

Regulatory and Legal Landscape

- CNIL (France): lead supervisory authority for Doctolib SAS under Art. 56 GDPR.
- BfDI (Germany): copied in the correspondence until 19 Aug 2026. For a private company in Germany the competent authority is normally a state data protection authority, so this copy did not reach the competent body.
- DSB (Austria): copied until 19 Aug 2026 because RFI-IRFOS is based in Austria. No response from the DSB appears in the correspondence.
- EDPS: copied until 19 Aug 2026. It has no competence over a private company's processing, so the copy was a courtesy and not a referral.

Two public statements sit outside this report's findings. The vendor describes research programmes under the CNIL's MR-004 methodology, which we did not evaluate. Press reports about AI training on health records are unverified and are not part of any finding here.

Indicators of Compromise and Reproducible Evidence

Artefact	SHA-256 or value
5.8.1 base.apk (66.0 MB)	2d7ac65e5060d49ddbdaafcad5cc31467d89169495b0fd581484077b7c0669b5
5.17.0 base.apk (69.4 MB)	7cccf0f951824ed4f92bfa3e7516f25a77f4c41ffadc40febd195053433f9dfb
5.17.0 split_config.arm64_v8a.apk	f4b57f8b2ce297f143145ae83dd8204b4f21dce810199a20548e0fd1d6ae4494
5.17.0 split_config.de.apk	498386bf135656e61105944db1c6230feef309ba1d4cb6f27240622e8f7c0ce8
5.17.0 split_config.xxhdpi.apk	b17efc48accd86f26b10105af5d1afafe50e03a4f68a81424fc1098bfd5e394d

```
aapt2 dump xmltree --file AndroidManifest.xml base.apk
unzip -p base.apk classes*.dex | grep -a -o 'Screeb android [0-9.]*'
unzip -p base.apk classes*.dex | grep -a -o 'r.screeb.app/rpc/1.3.0/l'
jadx -d out base.apk   # JADX 1.5.0, see caveats in the footer
```

Endpoints: r.screeb.app/rpc/1.3.0/l?source=sdk-android (telemetry, 5.8.1 static), t.screeb.app/webview.html?version=4 (5.8.1) and version=6 (5.17.0). Package identifiers: fr.doctolib.www, app.screeb.sdk, expo.modules.clipboard.

Disclosure Timeline and Outcome

Date	Event
4 Aug 2026	First report sent to the DPO, CNIL, BfDI, DSB and EDPS
6 Aug 2026	Vendor flags the disclosure date as inconsistent with a 90-day window. We correct it to 2 Nov 2026 the same day
18 Aug 2026	Vendor: no privacy breach identified, review ongoing, no answer to the four questions. We restate the questions
19 Aug 2026	Our reply sets 25 Aug as the new deadline. Vendor: some SDKs load before consent, impact to be confirmed internally
3 Sep 2026	Follow-up with the four questions
10 Sep 2026	Vendor answers each question. Confirms H2, names corrections on H3, explains H1 and the research programme
17 Oct 2026	Our response: confirmations recorded, open points named, weekly binary re-pull promised
2 Nov 2026	Vendor: Screeb 4.4.1 and clipboard fix live, Firebase key disputed, Messaging change for 16 Nov, Sentry classification by 30 Nov
2 Nov 2026	Our verification on 5.17.0 and publication of this report

We did not run the weekly binary re-pull that we announced on 17 October 2026. This report is based on one check after the first report, on 2 November 2026.

Residual Risk and Recommendations

- Screeb consent gate in JavaScript: unverified. A runtime capture on a device with no consent given would settle it.
- Firebase server-side configuration: unverified. The vendor’s statements stand as the vendor’s statements.
- Permissions READ_PHONE_STATE and ACCESS_FINE_LOCATION: declared in the 5.8.1 manifest. The vendor did not address them in its responses. Usage is not verified.
- Dates to watch: Firebase Messaging change on 16 Nov 2026, Sentry classification and cookie policy and DPA update on 30 Nov 2026.
- Our own commitment: the weekly re-pull announced on 17 October 2026 did not happen. We say so here and in the correspondence.
- For users: revoke the camera, microphone and location permissions that are not needed, and review the app’s privacy settings after each update.

RFI-IRFOS security@rfi-irfos.com • rfi-irfos.com
ZVR 1015608684 • GISA 39261441 • UID ATU83405245 • Steuernummer 68 696/8736 • Elisabethinergasse 25/10, 8020 Graz, Austria Method: static analysis of two APK builds (JADX

1.5.0, aapt2, string extraction) and comparison of the Android manifests. No dynamic analysis or network capture was performed for this report. Vendor statements are quoted from correspondence and marked as such. The Firebase API key value is withheld in this report. REF DOCTOLIB-2026-R1.