



Security & Privacy Disclosure, Final Report

Dundle (com.dundle.app)

A Consent Mode default of granted before any user choice, no certificate pinning on the checkout flow, a hardcoded Firebase key, two hardcoded Supabase projects with anon tokens, and two session replay capable SDKs in a gift-card marketplace, unresolved through 103 days and a 13-day-overdue embargo, with no reply of any kind across five messages to a published security contact

PUBLIC DISCLOSURE, 17 OCTOBER 2026 — EMBARGO REACHED 4 OCTOBER 2026, SIX FINDINGS REMAIN UNADDRESSED

Target	Korsit B.V., Zwembadweg 12, 5611 KS Eindhoven, Netherlands (KvK 69094438)
Package	com.dundle.app
Platform Audited	Android · v1.15.0 (versionCode 334), static analysis of the production APK and splits (Flutter, Dart AOT)
Initial Disclosure (R1)	2026-07-06 (REF com.dundle.app)
Disclosure / Embargo Date	2026-10-04
Report Published	2026-10-17
Overall Severity	HIGH
Findings	6 (5 High, 1 Medium)
Correspondence	5 messages sent by RFI-IRFOS over 59 days to compliance@korsit.com and security@dundle.com, both delivered without bounce, no reply of any kind
Authorities in copy	Austrian Datenschutzbehörde (DSB) - European Data Protection Supervisor (EDPS) - Dutch Autoriteit Persoonsgegevens

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

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

Contents

1	Executive Summary	4
2	Subject & Operator Context	5
3	Engagement Summary	5
4	Findings	5
4.1	H1: Google/Firebase Consent Mode defaults declared as granted in the manifest	5
4.2	H2: No certificate pinning and no network security config on the checkout flow	6
4.3	H3: Firebase API key and project identifiers hardcoded in the distributed binary	7
4.4	H4: Two Supabase project references with long-lived anon tokens hardcoded, one apparently non-production	7
4.5	H5: Session replay capable SDKs bundled in a checkout app: Datadog and Microsoft Clarity	8
4.6	M1: Staging backend hostname shipped inside the production binary	8
5	Impact Analysis	9
6	The Complete Chain: From Install to Unanswered Exposure	9
7	Data Protection — Article by Article	9
8	Regulatory & Legal Landscape	10
9	Indicators of Compromise	10
10	Disclosure Timeline & Outcome	10
11	What Dundle Got Right	11
12	Residual Risk & Recommendations	11

Executive Summary

Dundle (com.dundle.app, v1.15.0) is a gift-card and voucher marketplace operated by Korsit B.V. in Eindhoven. A static analysis of the production APK produced six findings, five HIGH and one MEDIUM, no CRITICAL. The manifest declares all four Google Consent Mode defaults as granted before any user choice (H1). The checkout flow has no certificate pinning and no network security config (H2). A Firebase API key with its project identifiers (H3) and two Supabase project references with long-lived anon tokens (H4) are hardcoded in the binary. Datadog Session Replay and Microsoft Clarity are bundled in an app that takes payments (H5). A staging hostname ships in the production build (M1).

Dundle publishes a named security contact, security@dundle.com, and the disclosure went there together with compliance@korsit.com on 6 July 2026. Both addresses accepted every message. Five messages over 59 days produced no reply of any kind, no human response, no automated acknowledgment and no bounce. The embargo expired on 4 October 2026 with all six findings unaddressed. None of the three questions has been answered.

The audit also found things done well: a real consent state abstraction exists in the code, Firebase Analytics collection is disabled by default, cleartext traffic is denied, authentication is passwordless OTP, sensitive local data sits in Keystore-backed storage, voucher codes show no plaintext local persistence, the permission set is proportionate, no Chinese or Russian SDK is present, and the security contact is public. The six findings stand on top of that baseline.

ID	Severity	Title
H1	HIGH	Google/Firebase Consent Mode defaults declared as granted in the manifest
H2	HIGH	No certificate pinning and no network security config on the checkout flow
H3	HIGH	Firebase API key and project identifiers hardcoded in the distributed binary
H4	HIGH	Two Supabase project references with long-lived anon tokens hardcoded, one apparently non-production
H5	HIGH	Session replay capable SDKs bundled in a checkout app: Datadog and Microsoft Clarity
M1	MEDIUM	Staging backend hostname shipped inside the production binary

Subject & Operator Context

Korsit B.V. of Eindhoven, the Netherlands, operates Dundle, a European marketplace for pre-paid gift cards, game credits and vouchers. The operator identity comes from the public privacy policy and about page of dundle.com. The main establishment in the Netherlands makes the Autoriteit Persoonsgegevens the lead supervisory authority. The app runs on Flutter with Dart AOT code, Firebase, Supabase, Commerce Layer for orders, Customer.io for messaging, Segment, Datadog, Microsoft Clarity and Sift for fraud scoring, with Paysafe-Card, PayPal and iDEAL named in the client.

Engagement Summary

RFI-IRFOS sent the initial disclosure on 6 July 2026 to compliance@korsit.com and security@dundle.com, with the Austrian DSB, the EDPS and the Autoriteit Persoonsgegevens in copy, and fixed a 90-day embargo of 4 October 2026. Follow-ups followed on 23 July, 9 August, 24 August and 3 September 2026. Each restated the open findings, and the last repeated the three questions word for word with a response deadline of 10 September 2026.

Every message was delivered and none bounced. Dundle is the only target in this audit series with a publicly named security contact that stayed silent through four consecutive messages. This report publishes 13 days after the embargo.

Findings

H1: Google/Firebase Consent Mode defaults declared as granted in the manifest

HIGH, CVSS v4.0 AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N, 7.1 — Google/Firebase Consent Mode defaults declared as granted in the manifest

```
firebase_analytics_collection_enabled      = false
google_analytics_adid_collection_enabled  = true
google_analytics_default_allow_analytics_storage = true
google_analytics_default_allow_ad_storage   = true
google_analytics_default_allow_ad_user_data = true
google_analytics_default_allow_ad_personalization_signals = true
```

```
package:dundle/infrastructure/user_tracking/tracking_consent_service.dart
AndroidTrackingConsentService, IosTrackingConsentService
TrackingConsentStatus.granted() / .denied() / .notDetermined() / .restricted()
FirebaseAnalyticsHostApi.setConsent
```

Impact: The manifest declares `google_analytics_default_allow_analytics_storage`, `allow_ad_storage`, `allow_ad_user_data` and `allow_ad_personalization_signals` all as true. This is the Consent Mode state applied to the first SDK calls at process start, before any runtime

code, including the app's own TrackingConsentService, can override it. Consent must precede processing under Art. 6(1) and Art. 7(1), and a default of granted is the opposite of privacy by default under Art. 25. The same manifest sets `firebase_analytics_collection_enabled` to false, so the configuration is inconsistent, which is a question for the controller. The Dart snapshot contains a real TrackingConsentService with granted, denied, notDetermined and restricted states that calls Firebase `setConsent`, and no named consent management platform vendor appears anywhere in the binary. Whether the service decides before the first network call that carries these defaults, and whether it governs Datadog, Clarity, Segment and Customer.io as well as Firebase, is for Dundle to document under Art. 5(2).

H2: No certificate pinning and no network security config on the checkout flow

HIGH, CVSS v4.0 AV:A/AC:L/AT:N/PR:N/UI:N/VC:L/VI:L/VA:N/SC:N/SI:N/SA:N, 7.0 — No certificate pinning and no network security config on the checkout flow

```
AndroidManifest.xml: no android:networkSecurityConfig on <application>
res/xml/: no network_security_config.xml (base APK and both splits)
libapp.so: no CertificatePinning, no pin-set constant
Endpoints without pinning: v2.dundle-api.com, phoenix.dundle.com, dundle.commercelayer
.io
```

Impact: The application declares no `android:networkSecurityConfig` and ships no `network_security_config.xml` in the base APK or either split. The binary contains no `CertificatePinning` class and no pinned SHA-256 constant. The certificate strings present (`registerBadCertificateCallback`, `SecurityContext`, `IOHttpClientAdapter`) are the generic `dart:io` primitives found in every Flutter app. The checkout, order and payment endpoints (`v2.dundle-api.com`, `phoenix.dundle.com`, `dundle.commercelayer.io`) rely solely on the platform trust store. For a gift-card marketplace where users complete real purchases, interception through a locally installed trusted CA on a managed device or corporate proxy is a recognized scenario that Art. 32(1)(a) and (b) expects proportionate measures against. A positive finding sits next to this one: `targetSdk 36` denies cleartext traffic by platform default and no `http://` URL to a Dundle-owned domain exists in the binary.

H3: Firebase API key and project identifiers hardcoded in the distributed binary

HIGH, CVSS v4.0 AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N, 7.0 — Firebase API key and project identifiers hardcoded in the distributed binary

```
google_api_key: [REDACTED IN PUBLIC REPORT -- disclosed privately to
  Dundle in R1]
google_crash_reporting_api_key: [REDACTED IN PUBLIC REPORT -- disclosed privately to
  Dundle in R1]
google_app_id: [REDACTED IN PUBLIC REPORT -- disclosed privately to
  Dundle in R1]
gcm_defaultSenderId: [REDACTED IN PUBLIC REPORT -- disclosed privately to
  Dundle in R1]
google_storage_bucket: [REDACTED IN PUBLIC REPORT -- disclosed privately to
  Dundle in R1]
```

Impact: res/values/strings.xml carries the Firebase API key, the app ID, the sender ID and the storage bucket of the dundle Firebase project verbatim, extractable with apktool in under a minute. Google documents Firebase API keys as identifiers and not as secrets. The exposure is therefore one of quota abuse and accountability: an extracted key lets a third party probe quota limits, and it places the burden on the controller under Art. 5(2) to show that Firebase Security Rules and the Cloud Storage bucket are correctly restricted.

H4: Two Supabase project references with long-lived anon tokens hardcoded, one apparently non-production

HIGH, CVSS v4.0 AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N, 7.0 — Two Supabase project references with long-lived anon tokens hardcoded, one apparently non-production

```
project reference 1: [REDACTED IN PUBLIC REPORT -- disclosed privately to Dundle in R1
  ]
project reference 2: [REDACTED IN PUBLIC REPORT -- disclosed privately to Dundle in R1
  ]
anon JWT 1: [REDACTED IN PUBLIC REPORT -- disclosed privately to Dundle in R1
  ]
anon JWT 2: [REDACTED IN PUBLIC REPORT -- disclosed privately to Dundle in R1
  ]
```

```
Supporting strings: get:supabaseUrl, get:supabaseKey,
'Failed to initialize Supabase', 'Failed casting the wishlist from supabase.'
```

Impact: The Dart snapshot embeds two distinct Supabase project references, each with its own long-lived anon role token. Supabase is an active dependency of the app (initialization and wishlist handling strings are present). Anon tokens are public by design, and their safety depends entirely on Row Level Security on every exposed table. A second, apparently non-production project reference in the production build either marks a staging and production

pair left reachable, or an old project from a migration that was never removed. Both endpoints are reachable by anyone who extracts these strings. No query was made against either project, the Row Level Security posture is for Dundle to demonstrate under Art. 5(2) and Art. 32(1)(b).

H5: Session replay capable SDKs bundled in a checkout app: Datadog and Microsoft Clarity

HIGH, CVSS v4.0 AV:L/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N, 7.0 — Session replay capable SDKs bundled in a checkout app: Datadog and Microsoft Clarity

```
Datadog config keys (dex): session_replay_sample_rate, session_replay_touch_privacy,
    session_replay_text_and_input_privacy, session_replay_image_privacy,
    session_replay_start_immediate_recording
Datadog classes: com/datadoghq/flutter/DatadogRumPlugin, DatadogLogsPlugin,
    DatadogSdkPlugin
Manifest provider: com.datadog.android.rum.DdRumContentProvider
Clarity: clarity_flutter (bundled license manifest NOTICES.Z)
```

Impact: The application bundles Datadog Real User Monitoring and Logs with Session Replay configuration constants (sample rate, touch privacy, text and input privacy, image privacy, start immediate recording), and separately bundles Microsoft Clarity through the clarity_flutter package. Both products are built to reconstruct what a user sees and taps. In an app whose core function includes entering account details and paying for vouchers, session replay capable tooling engages data minimisation under Art. 5(1)(c) and privacy by design under Art. 25 directly. The masking constants show that Datadog supports configurable masking. Whether masking is enabled and whether the checkout screen is excluded from replay is for Dundle to document under Art. 5(2).

M1: Staging backend hostname shipped inside the production binary

MEDIUM, CVSS v4.0 AV:L/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N, 5.9 — Staging backend hostname shipped inside the production binary

```
staging host: [REDACTED IN PUBLIC REPORT -- disclosed privately to Dundle in R1]
alongside: app.dundle.com, cdn.api.dundle.com, cdn.api.dundle.dev, cdn.dundle.com,
    phoenix.dundle.com, phoenix.dundle.dev, search.dundle.com, v2.dundle-api.com,
    dundle.commercelayer.io
```

Impact: An Azure App Service staging hostname is present as a string constant in the production Dart snapshot, next to *.dundle.dev development domains. Staging environments are typically less hardened than production, with weaker rate limits and more verbose errors. Their presence in a production artifact widens the attack surface without operational necessity. No request was sent to the staging host.

Impact Analysis

H1 and H5 concern what the app observes about its users, H2 concerns the channel they pay through, and H3, H4 and M1 concern the back ends behind the app. A consent default of granted together with replay capable SDKs means behavioral data can flow before a user decides, in an app where users enter payment and account details. Without pinning, the checkout depends on the device trust store alone. The two Supabase projects and the Firebase project are reachable by anyone who unpacks the APK, and their safety rests on Row Level Security and Security Rules that Dundle has not shown to be correct.

Five messages to a published security contact without a single reply is a finding about process. The channel exists on paper and no one answers it.

The Complete Chain: From Install to Unanswered Exposure

Step 1, Install: Download Dundle (com.dundle.app) from Google Play.

Step 2, Extract and decompile: Unpack base.apk and the arm64 split with apktool and read the string tables. The manifest meta-data, strings.xml and the Dart snapshot strings listed in the Indicators section are present in plaintext.

Step 3, First launch: The manifest applies Consent Mode defaults of granted before the app's own consent service runs.

Step 4, Checkout: Orders and payments travel over endpoints without pinning, while Datadog and Clarity are bundled in the same app.

Step 5, Back ends: Firebase and two Supabase projects are addressable with values taken from the binary, and a staging host sits next to production names.

Step 6, Disclosure: Five messages to the published security contact and the compliance mailbox over 59 days led to no reply.

Data Protection — Article by Article

Art. 5(1)(a) and Art. 25 GDPR, fairness and privacy by default: H1 sets all four Consent Mode defaults to granted.

Art. 5(1)(c) GDPR, data minimisation: H5 bundles two session replay capable SDKs in a payment app.

Art. 5(2) GDPR, accountability: H3 and H4 place the burden of demonstrating correct Security Rules and Row Level Security on the controller.

Art. 6(1) and Art. 7(1) GDPR, lawfulness and consent: H1 means consent must precede processing, not be assumed and corrected later.

Art. 32(1)(a) and (b) GDPR, security of processing: H2 for the missing pinning, H3 and H4 for hardcoded backend identifiers, M1 for the staging host.

Art. 33 and Art. 35 GDPR: the third question asks for a documented impact assessment covering fraud scoring and marketing personalisation.

Regulatory & Legal Landscape

The Autoriteit Persoonsgegevens is the lead supervisory authority for Korsit B.V. as the main establishment is in Eindhoven. RFI-IRFOS copied the Autoriteit Persoonsgegevens, the Austrian Datenschutzbehörde and the European Data Protection Supervisor on the disclosure and on follow-ups. The Austrian authority is copied because the matter was initiated from Austria and the app is distributed to Austrian users.

Indicators of Compromise

The following identifiers were extracted from the production APK (v1.15.0, versionCode 334, analyzed 2026-07-06) by static analysis only. No Dundle servers, accounts or infrastructure were probed, and no query was made against any Supabase or Firebase project.

```
base.apk SHA-256: bf6f88be4f3a14106a9f1e5ddf147f7262f679bf1089dc6dac5b239aa17b5ae4
split_config.arm64_v8a.apk SHA-256: 0
a69869b716fc04d860742324a5f83c1be2e8dd9210d50a3930f7419541efd4e
```

```
H1 google_analytics_default_allow_* = true (AndroidManifest.xml)
   TrackingConsentService (libapp.so)
H2 no android:networkSecurityConfig, no network_security_config.xml
H3 Firebase values in res/values/strings.xml (redacted)
H4 two Supabase project references and anon tokens in libapp.so (redacted)
H5 session_replay_* keys (dex), DdRumContentProvider, clarity_flutter
M1 staging hostname in libapp.so (redacted)
```

Disclosure Timeline & Outcome

2026-07-06 Initial disclosure (R1, REF com.dundle.app) sent to compliance@korsit.com and security@dundle.com, with the Austrian DSB, the EDPS and the Dutch Autoriteit Persoonsgegevens in copy. Embargo fixed at 2026-10-04. Delivered, no bounce.

2026-07-23 First follow-up to the same two addresses, 17 days after disclosure. No reply.

2026-08-09 Second follow-up, 34 days after disclosure, three questions restated. No reply.

2026-08-24 Third follow-up, 49 days after disclosure. No reply.

2026-09-03 Fourth follow-up, 59 days after disclosure, three questions restated word for word with a deadline of 2026-09-10. No reply, no deadline response.

2026-10-04 Embargo date reached. Correspondence record: five messages sent by RFI-IRFOS, zero bounces, zero replies of any kind, human or automated.

2026-10-17 This report published, 13 days after the embargo and 103 days after the original disclosure.

What Dundle Got Right

A custom consent state abstraction exists in the code and calls the native Firebase setConsent API. Firebase Analytics collection is disabled by default at the manifest level. Cleartext traffic is denied by platform default and no http:// URL to a Dundle domain exists in the binary. Payment methods are server driven. Authentication is a passwordless one-time code. Sensitive local data uses Keystore-backed secure storage, and no plaintext voucher code storage was found. The permission set is proportionate, with no location, camera, microphone, contacts or phone state. No Chinese or Russian SDK is present. A named security contact exists, which not every audited company offers.

These choices do not offset the findings. A consent abstraction does not undo a manifest default that applies first, and a clean permission set says nothing about pinning, hardcoded back ends or replay tooling.

Residual Risk & Recommendations

H1 — Set the four Consent Mode defaults to denied in the manifest and grant them only after an affirmative choice through the consent service, and show that the service governs Datadog, Clarity, Segment and Customer.io as well.

H2 — Add a network security config with certificate pinning for the checkout, order and payment endpoints.

H3 — Restrict the Firebase API key by application and API, and verify Security Rules and the storage bucket.

H4 — Remove the non-production Supabase project from the production build and verify Row Level Security on every exposed table of both projects.

H5 — Exclude checkout and account screens from replay, enable full masking, or remove session replay from the payment app.

M1 — Remove the staging hostname and development domains from the production build.

Process — Give security@dundle.com a named owner who answers the three questions of this disclosure.

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

ZVR 1015608684 • GISA 39261441 • UID ATU83405245 • Steuernummer 68 696/8736
• Elisabethnergasse 25/10, 8020 Graz, Austria All findings derive from static analysis of

com.dundle.app as published on Google Play, analyzed 2026-07-06, using publicly available tools (apktool, unzip, string extraction). No Dundle servers, accounts or production infrastructure were probed or accessed. Technical evidence is archived under chain of custody and will be provided to the Autoriteit Persoonsgegevens on request. Research conducted under GDPR Art. 89 (scientific research exemption), ISO/IEC 29147 (coordinated disclosure standard), and the Austrian Federal Constitutional Law Art. 17 (freedom of research). REF DUNDLE-2026-R1.