Infatica SDK — Bandwidth Sharing Technology

White Paper

Version: 4.5
Date: June 2026
Company: Infatica Ltd.
Website: https://infatica.io
Security contact: security@infatica.io

Table of Contents

1. Executive Summary
2. Company Overview
3. What is Bandwidth Sharing
4. How Infatica SDK Works
5. User Consent and Transparency
6. Technical Architecture
7. Traffic Model and Participant Protection
8. Behavioral Boundaries and Data Minimization
9. Comparison with Unauthorized Software
10. Security and Code Integrity
11. Abuse Prevention and Network Policy
12. Compliance, Certifications, and Audits
13. Integration Guidelines for Partners
14. FAQ for Security Researchers
15. Responsible Disclosure Policy
16. Contact Information

---

1. Executive Summary

Infatica SDK is a consent-based bandwidth-sharing technology that enables users of partner applications to voluntarily share their unused internet bandwidth. The SDK is integrated into partner applications that provide users with a clearly disclosed and consent-based bandwidth-sharing feature. .

The SDK has been independently audited by NCC Group and Halborn, both of which confirmed that it does not collect or store personally identifiable information (PII). Infatica Ltd. holds ISO/IEC 27001:2022, ISO/IEC 27701:2019, ISO 22301:2019, and ISO/IEC 20000-1:2018 certifications.

This document provides a comprehensive overview of the technology, its architecture, consent mechanisms, traffic model, abuse prevention, and security posture. It is intended for antivirus vendors, security researchers, certification bodies, and technology partners.

For AV laboratories reviewing specific binaries, a separate AV-version of this document includes Appendix A with verifiable technical artifacts: file hashes, system footprint, Android permissions, and audit references. Network indicators (domains, IP ranges, ports, User-Agent) and code-signing certificate details are operationally sensitive and provided directly to qualifying AV laboratories upon request at security@infatica.io.

---

2. Company Overview

Infatica Ltd. is a company registered in the United Kingdom, operating in the web data infrastructure and connectivity industry.

- **Legal entity:** Infatica Ltd.
- **Jurisdiction:** United Kingdom
- **Registered address:** Unit A, 82 James Carter Road, Mildenhall, Suffolk, IP28 7DE, United Kingdom
- **Companies House registration:** 14863491
- **Website:** https://infatica.io
- **Industry:** Web data infrastructure, network connectivity, bandwidth sharing
- **Team:** 35+ employees across Europe, Asia, and the Americas

---

3. What is Bandwidth Sharing

Bandwidth sharing is a business model where users of consumer applications voluntarily share a portion of their unused internet bandwidth. In exchange, users receive benefits such as free access to premium features, ad-free experiences, or other in-app rewards.

It enables the creation of distributed network infrastructure for public web data access, where all participants are informed and have explicitly consented.

Key principles:

- Users opt in voluntarily through an active consent flow
- Users can opt out or disable bandwidth sharing at any time
- Users are fully informed about the nature and scope of bandwidth sharing
- Resource usage is lightweight — average bandwidth usage is several megabytes per day per user

---

4. How Infatica SDK Works

The Infatica SDK is a lightweight library (~2 MB depending on platform) that partner applications integrate into their products.

Workflow:

1. User installs a partner application (e.g., a VPN app from Google Play or a website)
2. A consent screen is displayed — the user is informed about bandwidth sharing and must actively accept
3. User accepts the EULA — the SDK is activated only after explicit consent
4. The SDK operates in background — relays a small amount of third-party web traffic through the user's internet connection
5. User can disable bandwidth sharing at any time through application settings, with immediate effect

Supported platforms:

- **Windows:** 32-bit and 64-bit DLL (sdk_win32.dll, sdk_win64.dll)
- **macOS:** supported
- **Android:** AAR library, API 21+ (android-service.aar, unity-service.aar for Unity-based apps). Java and Kotlin.
- **iOS:** XCFramework. Swift and Objective-C.
- **Smart TV / Streaming:** Roku, WebOS, Tizen
- **Game engines:** Unity (certified), Unreal Engine (compatible)

Android SDK details:

- Package: `com.infatica.agent.service`
- Build format: AAR (Android Archive)
- Supports foreground and background operation, optimized to minimize battery impact
- No minimum device state required

Android security boundaries:

- The SDK operates entirely within standard Android application sandbox boundaries
- The SDK does not bypass Android platform security mechanisms or SELinux protections
- The SDK does not require root privileges or elevated system permissions
- The SDK does not use Accessibility Services, Device Administrator APIs, overlay abuse techniques, or undocumented Android APIs
- The SDK cannot execute outside the privilege scope granted to the host application by Android

---

5. User Consent and Transparency

User consent is the cornerstone of Infatica's approach. The SDK is never activated without explicit, informed consent.

Consent flow:

1. Partner application displays a prominent disclosure explaining bandwidth sharing
2. User actively opts in (no pre-checked boxes, no hidden terms)
3. SDK activates only after consent is recorded
4. User can opt out at any time through a dedicated settings toggle

The EULA describes the bandwidth-sharing functionality and is presented as part of the consent process. Example disclosure text (from Microsoft Store compliance documentation):

> "This app includes a feature that lets you share unused internet bandwidth to support app monetization. This may involve sharing your IP address with our monetization partner, Infatica. No other personal information or device identifiers are accessed or stored. Participation is optional, and you can opt out at any time."

Consent enforcement:

Consent enforcement is contractual and architectural. Partners are required to implement the consent flow before initializing the SDK. Infatica reviews partner implementations during onboarding and can suspend access for non-compliant partners by revoking their API key.

On Android, the SDK records the consent state, although the consent UI implementation itself is the partner's responsibility.

Technical consent verification at the SDK level (where the binary confirms user consent before initializing) is under consideration for future SDK versions.

Example implementation:

Riff VPN (https://riffvpn.com) demonstrates the complete consent flow with disclosure, opt-in, settings toggle, and opt-out.

---

6. Technical Architecture

SDK behavior:

- The SDK establishes outbound connections to Infatica's API infrastructure
- It functions as a distributed connectivity layer, routing authorized web requests through the user's internet connection
- Traffic routing: peer-to-business, public web data

- No remote command execution functionality is included in the SDK. The SDK does not dynamically download executable payloads, perform code injection, invoke shell execution, modify operating system security controls, or establish persistence mechanisms outside the host application lifecycle.
- The SDK has a lightweight, low CPU and memory footprint

Threat model and execution boundaries:

- The SDK establishes outbound-only communications initiated from the host device
- The SDK does not expose inbound listening services accessible from external networks
- The SDK does not accept arbitrary executable instructions from remote systems
- The SDK cannot independently install software or modify the host operating system
- The SDK operates solely within the runtime context of the host application

Network communication:

- SDK establish connection with API server and consistently communicate with it with heartbeat + get additional server list
- SDK establish connection with addition servers from cluster and waiting for requests
- SDK pass requests through user devices to destination resources, for example get html and images from amazon.com

Resource usage:

- SDK size: ~2 MB depending on platform
- Average bandwidth usage: several megabytes per day per user
- CPU and memory: lightweight, low footprint
- Android: supports foreground and background operation, optimized to minimize battery impact
- No minimum device state required

Idle behavior:

When no relay tasks are assigned, the SDK enters a standby mode — it is not actively processing traffic and consumes negligible resources while waiting for new tasks from the API server.

---

7. Traffic Model and Participant Protection

This section addresses how the traffic routing model works and what protections exist for users who participate in bandwidth sharing.

Traffic flow:

There are two distinct types of traffic on a device running the SDK:

1. **User's own traffic** — the user's personal browsing, apps, and services. The SDK does not interact with, inspect, log, or modify this traffic in any way. (Confirmed by NCC Group and Halborn security assessments.)
2. **Relay traffic** — authorized web requests that are routed through the user's internet connection by the SDK. This traffic is routed peer-to-business for public web data access.

The SDK does not inspect, log, store, or modify relay traffic content. The device acts as a transparent relay point. Independent assessment by Halborn (conducted July 2025, report issued August 2025) confirmed that the SDK code does not collect, process, or store end-user personal data beyond IP addresses and an anonymized UUID used solely for routing.

Participant protection measures:

Users who participate in bandwidth sharing are protected through layered measures spanning both the SDK and Infatica's infrastructure:

- **Server-side traffic filtration:** Abusive protocols and ports are blocked at the infrastructure level. The specific port and protocol list is operationally sensitive and provided to AV laboratories under NDA.
- **SDK-level protections (on participant device):** The SDK itself blocks relay traffic on a defined set of abuse-prone ports and prevents relay requests targeting internal/private network address ranges. This prevents the participant's device from being used as a pivot point into local networks (LAN, corporate intranets) and reduces the risk that the device is used to scan or attack non-public infrastructure.
- **Customer verification (KYC):** All enterprise customers accessing network capacity undergo identity verification.
- **IP blocks:** Sensitive websites are blocked from being accessed through the network.
- **Threat detection:** Relay traffic is monitored server-side (not on the participant's device) for threats, in partnership with BitDefender.
- **Abuse complaint handling:** Infatica responds to abuse reports at security@infatica.io within 24 hours. When a complaint is traced to relay traffic, the offending customer's access is revoked.

What participants should know:

When participating in bandwidth sharing, the user's IP address is visible to destination servers for relay traffic. This is inherent to distributed residential connectivity infrastructure. The IP address is used transiently server-side to support bandwidth sharing — it is not stored or logged on the device. Infatica mitigates associated risks through SDK-level port restrictions, infrastructure-level traffic filtration, server-side threat detection, customer verification, and abuse complaint handling.

---

8. Behavioral Boundaries and Data Minimization

What the SDK does not do:

The following boundaries have been verified through independent security assessments by NCC Group (conducted October 2025, report issued January 2026; dynamic black-box assessment, not a full source-code audit) and Halborn (conducted July 2025, report issued August 2025).

- Does not install additional software or modify system settings
- Does not mine cryptocurrency
- Does not display advertisements
- Does not operate without a valid partner API key (infrastructure-level enforcement)
- Does not include remote command execution functionality
- Does not interact with the user's own network traffic — only relay traffic passes through the SDK
- Fully removed upon uninstallation of the host application (verified — see Appendix A.3)

Additional security boundaries:

- Does not perform process injection or memory manipulation against other applications
- Does not attempt to evade security analysis, antivirus detection, or platform review mechanisms
- Does not disable, interfere with, or tamper with endpoint security software
- Does not establish covert background persistence mechanisms
- Does not self-update outside authorized application distribution mechanisms

Data minimization:

Confirmed independently by NCC Group (no sensitive or PII data found) and Halborn (no end-user personal data beyond IP and UUID). The SDK collects only:

- **IP addresses** — used transiently server-side for routing, not stored on device
- **Anonymized UUID** — used solely for routing, non-persistent

The SDK does not collect or access:

- Account data or credentials
- Cookies or session tokens
- Advertising IDs
- IMEI or device serial numbers
- Precise location
- Browsing history
- Installed applications
- Contacts, photos, or messages

No PII collection or logging occurs on the device. Only user IPs are logged on the server side for operational purposes.

Operational telemetry:

In addition to the routing data above, the SDK transmits minimal operational telemetry to Infatica's API servers, consisting of the anonymized UUID and aggregate traffic volume metrics (bytes relayed). No content data, device identifiers, browsing data, or PII is included in telemetry.

On-device vs server-side — what happens where:

Function	**On participant device**	**On Infatica servers**
Relay traffic routing	Yes — passes through device	Orchestrated server-side
User's own traffic	Not touched by SDK	N/A
Relay traffic inspection/logging	No	Operational metadata only (routing, aggregate volume); no content inspection or logging
PII collection	No	Only IP addresses logged for operations
Threat detection	Yes — SDK-level port restrictions and internal/private network address blocking	Yes — server-side monitoring (BitDefender)
Traffic filtration (blocked protocols)	Yes — abuse-prone protocols and ports blocked at SDK level	Yes — infrastructure-level filtration
Customer verification (KYC)	No	Yes — before customer access granted
Telemetry	Anonymized UUID and aggregate traffic volume metrics sent to API	Received and stored
Kill-switch / partner revocation	SDK stops when API key invalid	API key revocation managed here

---

9. Comparison with Unauthorized Software

The SDK is designed for transparent, consent-based commercial network participation and is not intended for covert deployment, unauthorized persistence, malware distribution, credential access, or circumvention of platform security controls.

The following comparison is factual, based on observable behavior and independent audit results:

Characteristic	**Unauthorized software (typical)**	**Infatica SDK**
User consent	No consent mechanism present	Active opt-in required by partner agreement before SDK use
Disclosure	No disclosure to user	Described in app UI, EULA, and store listing
Opt-out	No opt-out mechanism	Disable toggle in app settings, immediate effect
Distribution	Bundled with pirated software, malware droppers	Integrated by partner application developers after onboarding review
Consent enforcement	N/A	Contractual: partners required to implement consent; API key revoked for violations
Data access	May access personal data, credentials	No PII collected (confirmed by NCC Group and Halborn)
Remote execution	Often includes remote command capabilities	No remote command execution functionality
Resource usage	No limits	Lightweight — several MB/day bandwidth
Uninstallation	May persist, reinstall, or leave artifacts	Fully removed with host application (verified, see A.3)
Accountability	Anonymous operators	Registered UK company, ISO 27001 certified
Security audits	None	NCC Group, Halborn — no critical/high findings

Classification by major AV vendors:

Kaspersky classifies Infatica SDK as "Not-a-virus: Server-Proxy" — recognizing the proxy functionality while explicitly confirming the absence of malicious behavior. Microsoft Defender reviewed and removed its detection for multiple SDK builds following false positive submission (as of April 2026; detection status may change with future signature updates).

---

10. Security and Code Integrity

Independent security assessments:

NCC Group (conducted October 2025, report issued January 2026) NCC Group conducted a dynamic black-box security assessment of the Infatica SDK with no prior knowledge of the systems. This was an external dynamic assessment, not a full source-code audit. Results: no critical or high risk vulnerabilities found. The SDK was confirmed not to collect or store any sensitive or PII data. (Letter of Assurance dated January 5, 2026, reference O-225889 / E026983.)

Halborn (conducted July 2025, report issued August 2025) Halborn conducted a source code review and security assessment of the Android SDK. Results: 0 critical, 0 high, 2 medium, 1 low severity findings. 100% of reported findings were addressed. The code was confirmed not to collect, process, or store any end-user personal data beyond IP addresses and an anonymized UUID used solely for routing. Findings addressed:

- Cleartext traffic in test application — risk accepted (test app only, not production SDK)
- Release build without code obfuscation — solved (standard R8/ProGuard minification and obfuscation enabled for Android release builds; no packers or crypters are used)
- Sensitive information disclosure via logging — solved (debug logging of DNS addresses removed from production builds)

Software supply-chain controls:

- Release artifacts undergo automated validation prior to distribution
- Platform-specific binaries are verified during the CI/CD process
- Build artifacts are hash-tracked and version-controlled
- Dependency and release validation procedures are applied prior to publication
- Production releases are scanned for unauthorized executable artifacts before distribution

Code signing:

The Windows SDK is distributed in two forms:

1. **Signed by Infatica** — using an Extended Validation (EV) code signing certificate for partners who do not have their own certificate. Certificate details (subject, issuer,

thumbprint, validity) are extractable from any signed Infatica build and can be confirmed by AV laboratories upon request at security@infatica.io.

2. **Unsigned** — provided exclusively to partners who sign the DLL with their own EV certificate as part of the white-label strategy. This is a contractual arrangement — partners receive unsigned builds only after completing the onboarding process and signing the partnership agreement. Unsigned builds are never distributed to end users.

Android SDK:

Distributed as a standard AAR library. The partner's application signing key covers the SDK as part of the APK.

---

11. Abuse Prevention and Network Policy

Customer verification:

All enterprise customers accessing Infatica's network capacity undergo identity verification (KYC).

Safety measures:

Infatica implements the following safety measures at the infrastructure level and within the SDK:

- Infrastructure-level traffic filtration
- Identity verification (KYC) for all enterprise customers
- IP blocks for sensitive websites
- Server-side threat detection on relay traffic, in partnership with BitDefender
- SDK-level blocking of abuse-prone ports and protocols on participant devices
- SDK-level blocking of relay traffic destined for internal/private network address ranges, preventing the participant's device from being used as a pivot into local networks

Abuse response:

- Abuse reports: security@infatica.io
- Response SLA: 24 hours
- When abuse is confirmed, the offending customer's access is revoked immediately

Server-side control:

Infatica maintains the ability to remotely disable SDK functionality for any specific partner by revoking their API key. When an API key is revoked, the SDK stops operating — it does not retry or continue relaying traffic.

If a partner is found to violate consent requirements or distribute the SDK in unauthorized applications, their API key is revoked immediately.

---

12. Compliance, Certifications, and Audits

ISO certifications (all current):

Certification	Standard	Certificate No.	Issued	Expires	Certifying Body
Information Security Management	ISO/IEC 27001:2022	GB21905E	27 Nov 2025	26 Nov 2026	Integral Certification Ltd.
Privacy Information Management	ISO/IEC 27701:2019	MTS-45643	16 Jul 2025	15 Jul 2026	Sigmacert DOO
Business Continuity Management	ISO 22301:2019	MTS-45645	16 Jul 2025	15 Jul 2026	Sigmacert DOO
IT Service Management	ISO/IEC 20000-1:2018	MTS-45644	16 Jul 2025	15 Jul 2026	Sigmacert DOO

Independent security audits:

Auditor	Scope	Assessment period	Report issued	Critical	High	Medium	Low	PII Findings
NCC Group	SDK (dynamic black-box)	Oct 6–14, 2025	Jan 5, 2026	0	0	—	—	No PII collected or stored
Halborn	Android SDK (code review)	Jul 15–17, 2025	Aug 26, 2025	0	0	2 (resolved)	1 (resolved)	No PII beyond IP + anonymized UUID

Privacy compliance:

The SDK is designed to comply with GDPR, CCPA, and other major privacy regulations. The SDK does not process personal data as defined by GDPR — confirmed by independent audits. Partner applications are responsible for their own compliance regarding user data they collect independently of the SDK.

Infatica's privacy policy: https://infatica.io/privacy-policy

---

13. Integration Guidelines for Partners

Partners integrating the Infatica SDK must adhere to the following requirements:

1. **Consent** — implement a clear, visible consent flow with just-in-time opt-in dialog before activating the SDK
2. **EULA** — include bandwidth-sharing terms in the application's EULA
3. **Opt-out** — provide a dedicated settings section to enable/disable bandwidth sharing
4. **Disclosure** — describe bandwidth-sharing functionality in the app store listing, website, and privacy policy. Partners are additionally required to comply with applicable mobile platform policies, including Google Play disclosure, consent, background execution, and user data requirements.
5. **Updates** — keep the SDK updated to the latest version provided by Infatica
6. **Code signing** — partners with EV certificates are encouraged to sign the SDK with their own certificate. Partners without certificates will receive Infatica-signed builds.

Infatica reserves the right to revoke API access for any partner that fails to meet these requirements. Partner onboarding includes a review of the application and its consent implementation before API credentials are issued.

Integration time is typically a few hours to one day.

---

14. FAQ for Security Researchers

Q: Why does the SDK show network relay behavior? A: The SDK routes authorized public web requests through participant devices with explicit user consent. . Its purpose is to route authorized web requests through the user's internet connection with the user's consent. This is its intended and disclosed function.

Q: Why is the SDK detected by some antivirus products? A: Some AV products classify network relay functionality as PUA regardless of whether user consent is present. We work with AV vendors through their whitelisting programs to ensure accurate classification. Microsoft Defender has already reviewed and removed its detection.

Q: How does the SDK differ from a botnet? A: A botnet operates covertly without user knowledge. Infatica SDK requires explicit opt-in, provides an opt-out mechanism, is distributed through public application channels, and is operated by an ISO 27001 certified UK company with independent security audits.

Q: Does the SDK execute arbitrary or remotely supplied code? A: No. The SDK does not support arbitrary code execution, dynamic executable payload delivery, shell execution, or remote task execution beyond its documented network relay functionality.

Q: What prevents the SDK from being used without user consent? A: Infatica prohibits deployment of the SDK without informed user consent and reserves the contractual right to revoke API access for partners that violate consent, disclosure, or platform compliance requirements. Consent enforcement is contractual and architectural. The SDK requires a partner API key to function. Partners are required to implement user consent before initializing the SDK. On Android, the SDK records the consent state. Infatica reviews partner implementations during onboarding and revokes API access for non-compliant partners. Technical consent verification at the binary level is under consideration for future versions.

Q: Does the SDK collect any user data? A: The SDK collects only IP addresses (used transiently server-side for routing) and an anonymized UUID for routing. The SDK additionally transmits aggregate traffic volume metrics as operational telemetry. No other personal data, device identifiers, browsing history, or device content is collected. This has been confirmed by independent security assessments from NCC Group and Halborn.

Q: What traffic is allowed through the relay? A: Traffic routing is peer-to-business for public web data access. Sensitive websites are blocked at the infrastructure level. A defined set of abuse-prone ports and protocols is blocked at both the SDK level (on participant devices) and the infrastructure level. The SDK additionally blocks relay traffic to internal/private network address ranges to prevent its use as a pivot into local networks. The specific port and protocol list is dynamically managed and available to AV laboratories under NDA. Relay traffic is monitored server-side for threats in partnership with BitDefender.

Q: Does the SDK include remote command execution? A: No. The SDK does not include any remote command execution functionality.

Q: What happens if I find the SDK in a suspicious or unauthorized application? A: Report it to security@infatica.io immediately. Unauthorized use violates our terms. We will investigate and revoke the partner's API access if confirmed.

Q: How can I verify that an application is an authorized Infatica partner? A: Contact security@infatica.io with the application name, package ID, or binary hash. We will confirm authorization status.

---

15. Responsible Disclosure Policy

If you discover a security vulnerability in the Infatica SDK, we encourage responsible disclosure.

How to report: Send details to security@infatica.io. Please include:

- Description of the vulnerability
- Steps to reproduce
- Affected SDK version and platform
- Your contact information for follow-up

What to expect:

- Acknowledgment within 2 business days
- Assessment and initial response within 5 business days
- We will work with you on coordinated disclosure timing
- We will not take legal action against researchers who follow responsible disclosure practices

---

16. Contact Information

Security and abuse reports: security@infatica.io

False positive and AV whitelisting inquiries: security@infatica.io

General inquiries: https://infatica.io

Reference application with consent flow: https://riffvpn.com

© 2026 Infatica Ltd. All rights reserved.