

RDP-FIDO-GATE

Software Lifecycle, Maintenance and Support

Document ID	IM.RDP-FIDO.INT.04
Document version	1.0
Date of issue	2026-10-06
Applies to	RDP-FIDO-GATE 1.25.1
Language	English
Publisher	iMobile

Contents

- 1. General 3
- 2. Development and release 3
 - 2.1. Development tools 3
 - 2.2. Release build 3
 - 2.3. Pre-release verification 4
 - 2.4. Version numbering 4
 - 2.5. Delivery of updates 4
- 3. Defect correction 4
 - 3.1. Reporting a defect 4
 - 3.2. Handling 4
 - 3.3. Timescales 5
- 4. Product improvement 5
- 5. Technical support 5
- 6. Personnel 5
 - 6.1. Customer personnel 5
 - 6.2. Vendor personnel 5

About this document

This document describes how RDP-FIDO-GATE is developed, released, corrected and supported, and which skills are needed to operate it. It is part of the documentation delivered with the product.

Response and resolution times are not set by this document; they are agreed in the contract.

1. General

Item	Value
Product	RDP-FIDO-GATE
Version covered by this document	1.25.1
Release date	2026-10-06
Publisher	iMobile
Website	https://new-imobile.com

Keeps the Remote Desktop (RDP) port, and on Linux also SSH and other chosen TCP ports, closed in the computer's own firewall and opens them for a short time only to the address that presented an enrolled FIDO2 key or a TOTP one-time code. It consists of a gate service, clients for Windows, Linux and Android, its own screen-streaming protocol (FIDO Stream) and an optional relay server (hub) that the customer hosts.

The description follows the process groups of ISO/IEC/IEEE 12207 (Software life cycle processes): development, release, maintenance and operational support.

2. Development and release

2.1. Development tools

C++20. Windows: the Win32 API with no third-party runtime, Windows CNG and DPAPI cryptography, the http.sys HTTP server, WinHTTP, Media Foundation, Direct3D 11 and the Windows WebAuthn API; the C++ runtime is linked statically. Linux: GCC, OpenSSL 3 (static), GTK 3, SDL 2, libfido2, FFmpeg, OpenH264. Android: Kotlin, Jetpack Compose, the Android Keystore. The SSH console page is HTML and JavaScript (xterm.js).

2.2. Release build

Windows: built by the `build.ps1` script with the Visual Studio C++ compiler (x64), MSI packages with WiX Toolset 6; a release is produced by the publishing script, which runs the release checks first and writes SHA-256 checksums into the manifest. The Windows executables and packages carry no code-signing (Authenticode) signature; update authenticity rests on the signed manifest. Linux: one build for all distributions (GCC 10 or later, static OpenSSL 3 and libstdc++), deb and rpm packages and an archive with a `SHA256SUMS` file. Android: Gradle; the APK is signed with the vendor's developer key.

2.3. Pre-release verification

Built-in self-tests: `RdpFidoGate.exe selftest` and `rdpfido-gate selftest` (parsing of WebAuthn structures, signature verification and rejection of forgeries, one-time codes against the RFC 6238 test vectors), `stream selftest`, `cloud selftest`. A FIDO Stream transport test (unit checks and a lossy UDP loopback). Entry-lock tests (logic and a window scenario; the FIDO2 key is a software one). Before a Windows release the publishing script checks the source tree lineage, the change-log entry, the file versions and that the stream window shows the test picture. Linux: the `rdpfido-testkit` set (software key, load, port check, attack scenarios), a virtual USB key and an end-to-end scenario on installed packages. Android: unit tests and an emulator run. No certification testing has been performed.

2.4. Version numbering

A version number has three parts, major.minor.patch (for example 1.25.0). The Windows client and gate are released under one number. Patch releases may cover a single platform: 1.25.0 is the Windows release, 1.25.1 the Linux packages. The Android client has its own numbering (0.9.x are pre-beta builds); the hub carries the number of the version in which it last changed.

2.5. Delivery of updates

The Windows client asks the account server for an update manifest at start (when the setting is on) and on the user's command, sending the product identifier and its own version. The manifest is signed with the server's key (ECDSA P-256, the public key is built into the client); the client downloads the file, checks its SHA-256 against the manifest and replaces itself with the user's consent. The Windows gate is updated by installing a new MSI package or from the client: after a successful FIDO key check the client sends the gate a new executable, the gate checks its SHA-256, replaces itself keeping a backup and rolls back if the service does not start. The Linux packages, the Android app and the hub are updated by installing the new version over the old one. One year after a version's release date an update is mandatory.

3. Defect correction

3.1. Reporting a defect

Defect reports are accepted through:

- contact form: <https://new-imobile.com/contact>
- e-mail: admin@new-imobile.com

A report should state:

- the product name and version;
- the operating system version and a description of the environment;
- the steps that led to the defect;
- the text of the error message;
- the product's log files, if any (their location is given in the Installation and Administration Guide).

3.2. Handling

1. The report is registered and its receipt is confirmed to the reporter.
2. The defect is reproduced on the version named by the reporter and on the current version.
3. The report is classified as a product defect, a configuration or environment problem, or a change request.

4. A product defect is corrected in the source code; the correction is verified and included in the next release. If the defect prevents the product from being used for its purpose, the correction is released separately, without waiting for the next scheduled release.
5. The reporter is told the outcome and the version that contains the correction.

3.3. Timescales

Response and resolution times depend on the contract (licence agreement or support agreement) and are not given in this document.

4. Product improvement

Changes originate from user reports, pre-release verification, changes in the operating systems and third-party components the product depends on and, for functions tied to legal requirements, changes in those requirements.

Changes are delivered in subsequent releases. The version number changes according to the rules given under "Version numbering". The content of each release is communicated to users when it is published.

5. Technical support

Questions about installation, configuration and use are accepted through the same channels as defect reports. Support languages are English and Russian.

Technical support covers:

- advice on installing, configuring and using the product;
- receiving and handling defect reports;
- providing updates to the extent covered by the licence.

Support hours and any services beyond the above are defined in the contract.

6. Personnel

6.1. Customer personnel

User: working knowledge of Windows, Linux or Android and of using a FIDO2 security key or an authenticator app. Administrator: Windows or Linux administration (services, the firewall, Remote Desktop or xrdp, port forwarding) and command-line work with administrator rights; for the hub, administration of a Linux server and DNS.

6.2. Vendor personnel

Development, release and technical support are carried out by the vendor's own specialists, who are proficient in the tools listed under "Development tools" and know the product's internals.

The location of the development team, the development infrastructure and the support desk is stated in the contract.

Document history

Document version	Date	Product version	Change
1.0	2026-10-06	1.25.1	First issue