

# Remote access — implementation notes (v1 WireGuard)

WireGuard v1 — implementation and deploy

| Field                  | Value           |
|------------------------|-----------------|
| Author                 | Anton Afanasyeu |
| Revision               | R1              |
| Creation date          | 2026-06-03      |
| Last modification date | 2026-06-03      |
| Co-authored            |                 |
| Severity               | medium          |
| State                  | in progress     |
| Document type          | technical       |

## Table of contents

- App (developer settings)
  - Android VPN consent (WireGuard)
- Production deploy (no new nginx HTTP paths)
  1. Database
  2. config.php — required keys
  3. BE packages and permissions
  4. Cron (session cleanup)
  5. Verify
  6. Browser
- WireGuard data plane (UDP, not nginx)
- VPN process (AIDL)
- WireGuard third-party (Android)
- RSSH alpha (reverse SSH)
- RA control HTTP (:ra\_control)
- Tests

Source: `REMOTE_ACCESS_IMPL.md` — section links work in PDF viewers that support internal anchors.





## App (developer settings)

- Dropdown: **Disabled** | **WireGuard** (lab) | **RSSH** (alpha — outbound reverse SSH, no VPN)
- **VPN routing (WireGuard only)**: route scope **Hub only** (172.200.2.1/32, split tunnel) or **All traffic** (0.0.0.0/0); app scope **All apps** or **This app only** (IncludedApplications / VpnService.addAllowedApplication). RSSH ignores these. Prefs sent on heartbeat as vpn\_route\_scope / vpn\_app\_scope; changing them reconnects an active WG session.
- Pref key: dev\_remote\_access\_mode (AppPreferences)
- **Disabled**: notifies BE (status:disable), tears down VPN, stops RemoteAccessService
- **WireGuard**: foreground service; jittered poll **1–7 min** → heartbeat.php with type: ra
- Heartbeat URL derived from crash upload URL (.../api/heartbeat.php)
- Session credentials persisted until expires\_at; expiry tears down tunnel locally
- Userspace WG via third-party/wireguard-android (:tunnel / GoBackend) in isolated process com.fox.androidcast:vpn; main app controls via AIDL (IVpnService / VpnServiceClient); TUN fallback only if GoBackend fails

## Android VPN consent (WireGuard)

WireGuard on Android **must** use VpnService (see DeveloperSettingsActivity → VpnService.prepare()). This is **not avoidable** on stock devices:

| UX                     | WireGuard (v1, lab)                                 | RSSH (alpha essential)               |
|------------------------|-----------------------------------------------------|--------------------------------------|
| One-time system dialog | Yes — “Allow VPN?”                                  | No VPN permission                    |
| Status-bar key icon    | Yes while tunnel up                                 | No                                   |
| Full-device routing    | TUN interface (split routes in config, still a VPN) | No L3 tunnel                         |
| Outbound from phone    | UDP to BE : 45340                                   | HTTPS poll + outbound SSH to bastion |

There is no supported “hidden VPN” on non-root Android: any TUN creator is classified as a VPN app. **RSSH** (foreground service + existing HTTPS poll + outbound SSH reverse tunnel) is the **alpha deliverable** agreed by project owners — device initiates, operator reaches a forwarded port on the bastion, without VPN consent or status-bar key. See Rev. 3 Q&A and ROADMAP.md § Remote access.

Uses the **existing crashes vhost** — no new nginx location blocks for HTTP.

| Piece         | Path                                                            |
|---------------|-----------------------------------------------------------------|
| Migration     | sql/migrations/007_remote_access.sql                            |
| Repository    | src/RemoteAccessRepository.php                                  |
| WG peers      | src/WireGuardPeerProvisioner.php (wg set wg0 peer ...)          |
| Device API    | public/api/heartbeat.php (type: ra)                             |
| Admin API     | public/api/remote_access.php                                    |
| UI            | ?view=remote_access (nav + hub card)                            |
| RBAC          | remote_access_view, remote_access_operate, remote_access_admin  |
| RBAC admin UI | ?view=rbac — global roles (root), company roles, privilege sets |
| Cron          | scripts/purge_remote_access.php (hourly recommended)            |

## RBAC rules (enforced in API)

| Permission            | Scope                                                  |
|-----------------------|--------------------------------------------------------|
| remote_access_view    | List devices/sessions/events in allowed companies      |
| remote_access_operate | Open session; close <b>own</b> sessions                |
| remote_access_admin   | Whitelist devices; close <b>any</b> session in company |
| Global admin / root   | All companies; root edits global roles via ?view=rbac  |

Privilege sets (remote\_access\_user, remote\_access\_admin) on company\_memberships.permissions\_json grant extra actions to member rows — see crash console README § RBAC.

## Flow

1. Device polls with {type:ra, status:enable, capabilities:[wireguard], tunnel\_mode:wireguard}.
2. BE upserts remote\_access\_devices; returns wait until device is **whitelisted** and operator **opens session**.
3. Next eligible poll returns action:connect + ephemeral WG credentials; BE runs wg set when provision\_peers=true.
4. Disable from app → status:disable → closes sessions, removes peers.

## URLs (production)

- Dashboard: https://apps.f0xx.org/app/androidcast\_project/crashes/?view=remote\_access
- Heartbeat: .../crashes/api/heartbeat.php
- Admin API: .../crashes/api/remote\_access.php

## Production deploy (no new nginx HTTP paths)

After syncing PHP/JS to the BE tree under

.../androidcast\_project/android\_cast/examples/crash\_reporter/backend/:

## 1. Database

Migration **007** (already applied on prod per deploy notes). Tables auto-create on SQLite dev; MariaDB should use the migration file once as root.

## 2. config.php — required keys

```
'remote_access' => [
    'wg_endpoint' => 'ra.apps.f0xx.org:45340', // public UDP hostname:port (FE DNAT → BE)
    'wg_server_public_key' => '...',          // wg show wg0 public-key on BE — REQUIRED
    'wg_interface' => 'wg0',
    'provision_peers' => true,                 // wg set on connect/close
    'require_wg_tools' => true,               // reject connect if wg genkey missing
    'session_ttl_s' => 3600,
    'min_poll_interval_s' => 45,              // poll rate limit per device
],
```

**Critical:** wg\_server\_public\_key must match the live wg0 interface. Empty value makes connect credentials unusable on real devices.

## 3. BE packages and permissions

```
apk add wireguard-tools wireguard-tools-wg
```

```
sudo -u www-data wg show wg0
```

## 4. Cron (session cleanup)

```
0 * * * * cd /var/www/.../backend && php81 scripts/purge_remote_access.php --hours=24
```

Use php81 when Alpine default php is 8.5+ without the session module; FPM is php-fpm81.

## 5. Verify

```
cd examples/crash_reporter/backend
chmod +x scripts/ra_*.sh scripts/verify_remote_access_prod.sh scripts/test_remote_access_api.sh
./scripts/verify_remote_access_prod.sh
BASE=https://apps.f0xx.org/app/androidcast_project/crashes ./scripts/test_remote_access_api.sh
BASE=https://apps.f0xx.org/app/androidcast_project/crashes ./scripts/ra_e2e_cli.sh
```

**Linux CLI device simulator** (same heartbeat payloads as the app): REMOTE\_ACCESS\_VALIDATION.md.

## 6. Browser

Hard-refresh after deploying public/assets/js/remote\_access.js.

## WireGuard data plane (UDP, not nginx)

HTTP control plane stays on FE→BE :80. **WireGuard traffic is UDP** — configure at the **FE firewall/DNAT**, not in nginx:

Internet UDP :45340 → FE DNAT → BE wg0 :45340

Hostname ra.apps.f0xx.org (or similar) should resolve to the FE public IP. Devices use wg\_endpoint from config/connect payload.

## VPN process (AIDL)

| Component              | Process                     | Role                                                         |
|------------------------|-----------------------------|--------------------------------------------------------------|
| VpnServiceClient       | main (com.foxx.androidcast) | Binds to IVpnService; async apply/tearDown + state callbacks |
| AndroidCastVpnService  | :vpn                        | AIDL stub; worker thread for tunnel ops                      |
| WireGuardVpnEngine     | :vpn                        | GoBackend + TUN fallback                                     |
| RemoteAccessVpnService | :vpn                        | VpnService TUN when GoBackend fails                          |
| GoBackend\$VpnService  | :vpn                        | WireGuard tunnel library VPN entry (manifest merge)          |

Developer settings still uses VpnService.prepare(Activity) for the one-time consent dialog (same app UID).

**Crash capture:** CrashReporter.install() registers uncaught handlers in :vpn (scenario vpn\_service, process com.foxx.androidcast:vpn). Reports persist to shared app storage and upload via :crashwatcher. Lab: scripts/simulate-vpn-crashes.sh (debug APK + MODE=device, or MODE=upload for BE ingest validation).

## WireGuard third-party (Android)

```
bash scripts/init-third-party-submodules.sh # or ./rebuild.sh (includes init + :tunnel build)
./gradlew :tunnel:assembleDebug :app:assembleDebug
```

Gradle fails fast if the submodule is missing. VPN runs in :vpn (AndroidCastVpnService + AIDL); WireGuardVpnEngine uses GoBackend with TUN fallback in the same process.

## RSSH alpha (reverse SSH)

| Layer | Component | Notes |
|-------|-----------|-------|
|-------|-----------|-------|

|            |                                                    |                                                           |
|------------|----------------------------------------------------|-----------------------------------------------------------|
| <b>App</b> | ReverseSshTunnelBridge                             | JSch outbound -R<br>127.0.0.1:<port>:127.0.0.1:8022       |
| <b>App</b> | RsshLocalSshServer                                 | Apache MINA SSHD on 127.0.0.1:8022                        |
| <b>BE</b>  | RsshSessionProvisioner /<br>RsshBastionProvisioner | DB creds; optional Linux user via<br>rssh_bastion_user.sh |
| <b>FE</b>  | nginx stream                                       | nginx/rssh-bastion-stream.conf.example                    |
| <b>Sim</b> | examples/rssh/linux-sim/                           | Laptop heartbeat + ssh -R                                 |

Developer settings → **RSSH** (no VPN dialog). Operator command shown in admin active sessions.

## RA control HTTP (:ra\_control)

Isolated process: template web UI + sandbox shell when RA session is up. Port/token in adb.json ra\_control block.

## Tests

```
./gradlew :app:testDebugUnitTest --tests 'com.foxx.androidcast.remoteaccess.*'
bash examples/crash_reporter/backend/scripts/test_rssh_unit.sh
bash examples/crash_reporter/backend/scripts/test_rssh_api.sh
bash examples/crash_reporter/backend/scripts/test_remote_access_api.sh
BASE=https://apps.foxx.org/app/androidcast_project/crashes bash examples/crash_reporter/backend/scripts/test_remote_access_api.sh
BASE=https://apps.foxx.org/app/androidcast_project/crashes bash examples/crash_reporter/backend/scripts/ra_e2e_cli.sh
```

CLI validation guide: REMOTE\_ACCESS\_VALIDATION.md.