

Android Cast — Email agent config & 2FA / mobile auth flow

Email forwarding + registration/2FA — alpha must-have

Field	Value
Author	Anton Afanasyeu
Revision	R1
Creation date	2026-06-08
Last modification date	2026-06-08
Co-authored	
Severity	medium
State	in progress
Document type	technical

Table of contents

- Document history
- Executive summary
- Current state (repo)
- §9 — Email addresses & agent config
 - §9.1 Prerequisites
 - §9.2 Register addresses (free — DNS, not “buy mailboxes”)
 - §9.3 Forward / duplicate to Gmail (info@ → foxxspambox0@gmail.com)
 - §9.4 Recommended prod layout for this project
 - §9.5 Outbound mail from BE (verification emails)
- §10 — Registration, 2FA, mobile auth
 - §10.1 Registration & login stages
 - §10.2 Open-source building blocks (recommended)
 - §10.3 Users DB migration (privacy-first)
 - §10.4 On-demand user maintenance (admin + self-service)
 - §10.5 Operator checklist (track progress)
 - §10.6 Mobile-friendly UX (not BE-only)
 - §10.7 Alpha implementation plan
- UI/UX alignment (existing consoles)
- Security & compliance notes
- Open questions
- Source linkage

Source: 20260607-2FA-email-mobile-auth-flow.md — section links work in PDF viewers that support internal anchors.

Document history

Rev	Date	Focus
1	2026-06-07	Initial FR capture: email forwarding + registration/2FA for alpha; infra + BE + mobile flows.

Alpha gate: agreed with project owners — user registration with verified email + second factor is **must-have for full alpha** (alongside RSSH remote access). LAN-only cast demo may still skip account flows.

Executive summary

Track	Goal
§9 Email	Public-facing addresses <code>info@</code> , <code>admin@</code> , <code>root@</code> on <code>apps.f0xx.org</code> (and optional aliases on <code>f0xx.org</code>) that forward to personal Gmail; no paid mailbox SKU required.
§10 Auth	Replace “Register (coming soon)” with email verification + flexible 2FA (TOTP QR, optional WebAuthn/passkey on Android), captcha on abuse-prone endpoints, privacy-preserving audit tables, admin tools for root/slug admins.
Mobile	Same flows usable on phone browsers and deep links / Custom Tabs from the Android app where appropriate.
Licenses	Prefer MIT / BSD / Apache-2.0 PHP libraries; avoid proprietary-only auth SaaS for core path.

Current state (repo)

Area	Today
BE login	Username + password_hash only — <code>Auth.php</code>
Registration	UI placeholder: “Register (coming soon)” — <code>login.php</code>
Users table	<code>username</code> , <code>password_hash</code> , <code>role</code> — no email, no 2FA — <code>schema.mariadb.sql</code>
Outbound email	Not configured — no SMTP in <code>config.example.php</code>
Mobile app	No account linking to crashes console yet
Roadmap	“User registration + email verification” — <code>examples/crash_reporter/ROADMAP.md</code>

§9 — Email addresses & agent config

Target addresses (any one canonical pair is fine; document both for flexibility):

Role	Preferred	Alias
Public / info	<code>info@apps.f0xx.org</code>	<code>info@f0xx.org</code>
Admin / ops	<code>admin@apps.f0xx.org</code>	<code>admin@f0xx.org</code> , <code>root@apps.f0xx.org</code>

Agent config (future): `BE config.php` → `mail.from`, `mail.reply_to`, `mail.envelope_from` pointing at these addresses once DNS and forwarding work.

§9.1 Prerequisites

You do **not** need to buy Google Workspace or Microsoft 365 mailboxes. You **do** need:

#	Prerequisite	Why
1	DNS control for <code>f0xx.org</code> and <code>apps.f0xx.org</code>	MX, SPF, DKIM, DMARC live at the DNS host (registrar or Cloudflare).
2	Decide canonical mail domain	Recommend <code>apps.f0xx.org</code> for product mail; keep <code>f0xx.org</code> as <code>forward/alias</code> only if desired.
3	Inbound path	MX records → free forwarder or self-hosted Postfix alias on FE/BE.
4	Outbound path (BE)	SMTP relay for verification/reset mail (see §9.5) — can be Gmail SMTP, SendGrid free tier, or local Postfix with SPF alignment.
5	TLS on web	Already have <code>https://apps.f0xx.org</code> — verification links use same vhost.

6	Gmail destination	e.g. foxxspambox0@gmail.com — receives forwarded + BCC copies.
---	-------------------	--

Deliverability minimum (avoid spam folder):

SPF TXT @apps.f0xx.org "v=spf1 include:<forwarder-or-smtp-provider> -all"
DKIM TXT (provider gives name/value)
DMARC TXT _dmarc.apps.f0xx.org "v=DMARC1; p=none; rua=mailto:admin@apps.f0xx.org"
Start with p=none; tighten after monitoring.

§9.2 Register addresses (free — DNS, not “buy mailboxes”)

“Register” here = DNS + forwarder rules, not purchasing seats.

Option A — Cloudflare Email Routing (recommended if DNS is on Cloudflare)

1. Cloudflare dashboard → Email → Email Routing → enable for apps.f0xx.org.
2. Add destination address foxxspambox0@gmail.com (verify Gmail link once).
3. Create Custom addresses:
 - info@apps.f0xx.org → forward to Gmail
 - admin@apps.f0xx.org → forward to Gmail
 - root@apps.f0xx.org → forward to Gmail (or same rule as admin)
4. Cloudflare adds MX automatically; copy SPF hint from dashboard.
5. Repeat for f0xx.org zone if you want info@f0xx.org (separate zone rules or redirect-only).

Cost: \$0 on Cloudflare free plan.

Option B — ImprovMX (DNS at any registrar)

1. Sign up at improvmx.com (free tier: limited aliases).
2. Add domain apps.f0xx.org → set MX to ImprovMX hosts (they show exact records).
3. Create aliases info, admin, root → foxxspambox0@gmail.com.
4. Add SPF include for ImprovMX.

Option C — Self-hosted alias (Gentoo FE or Alpine BE)

If you already run Postfix on FE:

```
/etc/postfix/virtual:
info@apps.f0xx.org    foxxspambox0@gmail.com
admin@apps.f0xx.org  foxxspambox0@gmail.com
root@apps.f0xx.org   foxxspambox0@gmail.com
```

Then postmap /etc/postfix/virtual && systemctl reload postfix. Requires MX pointing to FE public IP and port 25 reachable (often blocked on residential; OK on HVM with firewall rule).

Do first: pick A or B unless you explicitly want to operate SMTP on FE.

§9.3 Forward / duplicate to Gmail (info@ → foxxspambox0@gmail.com)

Method	Inbox copy	Send-as From	IMAP on apps domain
Forward only (Cloudflare / ImprovMX)	Yes — same subject/body in Gmail	Reply-from Gmail unless configured	N/A — read in Gmail
Gmail “Send mail as”	N/A	Can send as info@apps.f0xx.org if SMTP creds exist	Optional
BCC archive	Forward + hidden copy to second Gmail	—	Rarely needed

For FR §9.3 (“same topic and body redirected”): **plain forward** is enough. Gmail shows original To: header; you can filter with Gmail rule to: info@apps.f0xx.org.

Optional Gmail filter:

To: info@apps.f0xx.org → label: androidcast-info

No IMAP/POP3 server on apps.f0xx.org is required for this FR.

§9.4 Recommended prod layout for this project

Inbound (public):
info@apps.f0xx.org ■■forward■■■ foxxspambox0@gmail.com
admin@apps.f0xx.org ■■forward■■■ foxxspambox0@gmail.com
root@apps.f0xx.org ■■forward■■■ foxxspambox0@gmail.com (same inbox or +tag)

Outbound (BE PHP):
From: Android Cast Issues <noreply@apps.f0xx.org>
Reply-To: info@apps.f0xx.org
SMTP: transactional relay (see §9.5)

Document chosen provider in INFRA.md § changelog once live.

§9.5 Outbound mail from BE (verification emails)

Add to config.example.php (implementation phase):

```
'mail' => [
  'transport' => 'smtp', // smtp | sendmail
  'from' => 'Android Cast Issues <noreply@apps.f0xx.org>',
  'reply_to' => 'info@apps.f0xx.org',
  'smtp' => [
    'host' => 'smtp.example.com',
    'port' => 587,
    'encryption' => 'tls',
    'username' => '',
    'password' => '', // env / secrets file, not git
  ]
]
```

},
},

Free/low-cost SMTP options:

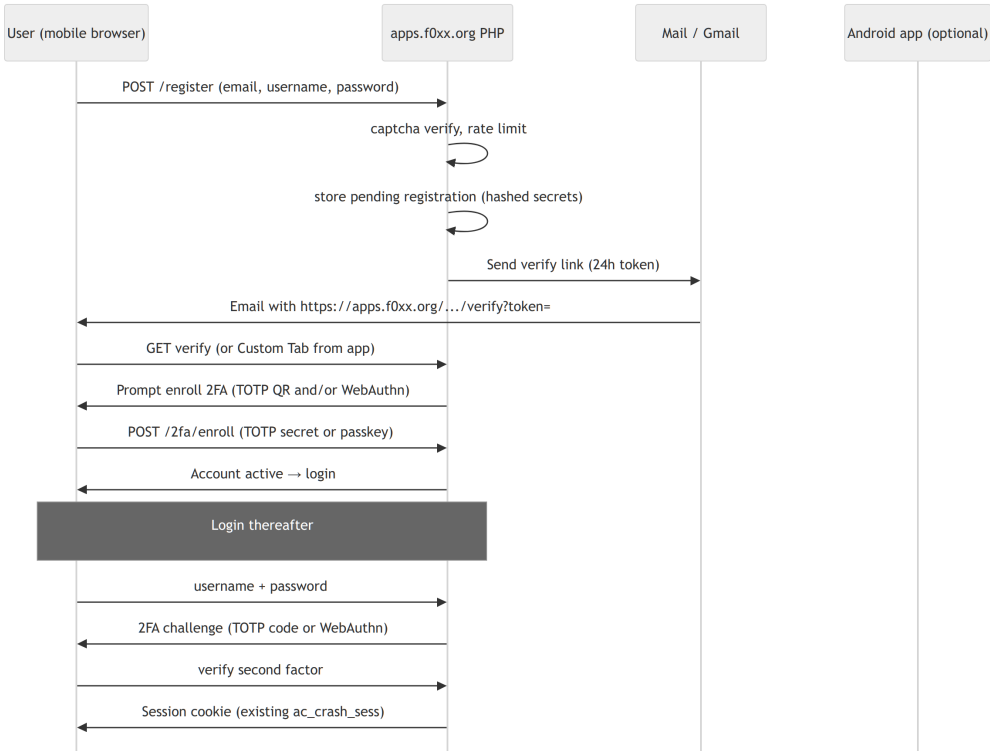
Provider	Notes
Gmail SMTP + App Password	OK for low volume; From may need "Send as" alignment
Brevo / SendGrid free tier	Better for noreply@apps.f0xx.org alignment
Local Postfix on BE	Full control; must publish SPF for BE egress IP

PHP library: **Symfony Mailer** (MIT) — already common in PHP ecosystems.

§10 — Registration, 2FA, mobile auth

§10.1 Registration & login stages

Proposed **alpha** flow (flexible — toggles in config):



Stage	Method	Mobile-friendly
10.1.2 Email confirmation	HTTPS link with signed token	Email app → Chrome Custom Tab; large tap target
10.1.3 TOTP / QR	Standard otpauth:// QR + manual key	Scan with Google Authenticator / Aegis; or type 6 digits
10.1.4 WebAuthn / passkey	Platform authenticator (fingerprint)	Android Chrome + future app WebView if needed
Captcha	ALTCHA (MIT, self-hosted) or rate limit only at alpha	Invisible/PoW — no Google reCAPTCHA dependency
Recovery	Backup codes + optional second recovery email	Printable one-time codes; store hashed

Roles: new self-registered users default to viewer + default company membership (existing RBAC). Elevation to admin/root remains admin-only.

§10.2 Open-source building blocks (recommended)

Function	Library	License	Notes
SMTP	symfony/mailer	MIT	Transport abstraction
TOTP	spomky-labs/otphp	MIT	RFC 6238; QR via endroid/qr-code
WebAuthn	web-auth/webauthn-lib	MIT	Passkeys; PHP 8.1+
Password hash	password_hash()	PHP core	Already used
Captcha	altcha-org/altcha	MIT	Self-hosted PoW widget
JWT / tokens	Signed random + HMAC in DB	—	Prefer opaque tokens stored hashed

Avoid as hard dependency for alpha: proprietary Auth0/Okta-only flows (OK as optional future SSO).

Android app (phase 2 of alpha auth):

Feature	Approach
Open verify link	CustomTabsIntent → https://apps.f0xx.org/.../verify
TOTP	Optional in-app entry (Web login sufficient for alpha)
Passkey	Chrome on device; link account in settings later
Deep link	androidcast://auth/verify?token= optional — not required if HTTPS works

§10.3 Users DB migration (privacy-first)

New migration 008_auth_email_2fa.sql (sketch — apply on MariaDB prod once implemented):

Extend users:

Column	Type	Notes
email_normalized	VARCHAR(254) NULL UNIQUE	Lowercase email
email_verified_at	TIMESTAMP NULL	NULL until link clicked
status	ENUM	pending, active, locked, disabled
recovery_email_normalized	VARCHAR(254) NULL	Optional second address

New tables (anonymous / hashed where possible):

```
-- auth_pending_registrations: delete after verify or TTL
-- auth_factors: user_id, type ENUM('totp', 'webauthn', 'backup_code'), secret_or_credential_encrypted, label, created_at
-- auth_tokens: purpose ENUM('verify_email', 'reset_password', 'login_magic'), token_hash, expires_at, used_at
-- auth_attempts: ip_hash, username_hash, outcome, created_at (no raw IP in prod if avoidable)
-- auth_audit: actor_user_id, action, meta_json, created_at
```

Privacy rules:

- Store **hashed** opaque tokens (SHA-256), never raw token in DB.
- Store TOTP secrets **encrypted at rest** (libsodium secretbox, key in config.php env).
- auth_attempts.ip_hash = HMAC(server_pepper, ip) — supports rate limit without retaining IP.
- WebAuthn credentials: store credential_id + public_key only (standard).

SQLite dev: mirror schema in Database::ensureSchema like existing RBAC migrations.

§10.4 On-demand user maintenance (admin + self-service)

Action	Who	UI location
Add / change recovery email	User	?view=account_security (new)
Re-enroll TOTP / add passkey	User	Same; requires recent password + 2FA
Revoke passkeys / backup codes	User	Same
Clear failed login/register attempts	root, company admin	?view=rbac → user row → “Clear auth lockouts”
Disable user / force password reset	root, platform_admin	RBAC admin
View auth audit tail	root	?view=auth_audit (read-only)

Session cookie path stays /app/androidcast_project (shared across crashes/build consoles).

§10.5 Operator checklist (track progress)

Use this table during implementation (copy to ticket #9 / #10):

#	Task	Owner	Done
E1	Enable Cloudflare Email Routing (or ImprovMX) for apps.f0xx.org	Ops	■
E2	Forward info@, admin@, root@ → Gmail	Ops	■
E3	SPF + DKIM + DMARC records live	Ops	■
E4	Test inbound: send mail to info@apps.f0xx.org, arrives Gmail	Ops	■
E5	Configure BE SMTP + send test from staging	Dev	■
A1	Migration 008_auth_email_2fa.sql + Database::ensureSchema	Dev	■
A2	Register + verify email views (EN/RU, app.css)	Dev	■
A3	TOTP enroll + login challenge	Dev	■
A4	WebAuthn enroll (optional alpha)	Dev	■
A5	ALTCHA on register/login	Dev	■
A6	Admin: clear lockouts + audit view	Dev	■

A7	Mobile soak: register on phone Chrome, verify, login	QA	■
A8	Document secrets in INFRA + config.example.php	Dev	■

§10.6 Mobile-friendly UX (not BE-only)

Requirement	Implementation
Viewport	Already width=device-width in layout.php
Touch targets	Buttons ≥ 44px; reuse .login-card, .btn from app.css
i18n	Extend en.json / ru.json
Email links	Single primary button “Confirm email” — no raw URL only
2FA QR	Responsive or SVG; “Can’t scan?” reveals manual secret
Errors	Same .alert pattern as login
Android app	Settings → “Project account” → opens Custom Tab to account/security (post-alpha polish OK)

§10.7 Alpha implementation plan

Phase	Scope	Estimate
P0 — Email infra	\$9 DNS + forward + outbound SMTP test	0.5–1 day ops
P1 — Register + verify	Pending table, mailer, verify link, captcha	2–3 days
P2 — TOTP login	Enroll QR + challenge on login	1–2 days
P3 — Admin tools	Recovery email, clear lockouts, audit	1 day
P4 — WebAuthn	Passkey enroll/login (Chrome/Android)	1–2 days optional for alpha
P5 — Mobile app link	Custom Tab entry points	0.5 day

Alpha sign-off (auth): new user can register on phone, verify email, enroll TOTP, log in with 2FA; admin can clear logout; inbound info@mail reaches Gmail.

Update ALPHA.md § G (auth) and ROADMAP.md alpha backend checklist when P1 starts.

UI/UX alignment (existing consoles)

Do **not** introduce a new design system. Reuse:

Asset	Path
Styles	public/assets/css/app.css — .login-page, .login-card, .alert, .muted, .locale-picker
Layout shell	views/layout.php — nav, hub card styling
i18n	public/assets/js/i18n.js + JSON catalogs
Analytics hook	AnalyticsHead::render() on auth pages (optional)

New views (planned):

- views/register.php
- views/verify_email.php
- views/account_security.php
- views/two_factor_challenge.php

Hub landing contact form (landing-pages.inc.php) should eventually use info@apps.fox.org as mailto: — align when email live.

Security & compliance notes

- Enforce **rate limits** on register, login, verify resend (e.g. 5/min/IP hash).
- Lock account after N failed 2FA attempts; unlock via email or admin.
- Do **not** log passwords or TOTP codes.
- Backup codes: generate 10 one-time codes, display once, store bcrypt hashes.
- GDPR-style minimization: optional auth_attempts retention job (purge > 90 days).

Open questions

#	Question	Default proposal
Q1	Require WebAuthn for alpha or TOTP-only?	TOTP required ; WebAuthn optional P4
Q2	Allow registration on prod immediately or invite-only?	Open register with captcha; viewer role
Q3	Single Gmail inbox for all aliases?	Yes for now (foxxspambox@gmail.com)
Q4	Separate noreply@ for outbound?	Yes — no forward needed; SPF only

Source linkage

Topic	Doc / code
Infra / DNS	INFRA.md
Crash console	CRASH_REPORTER.md, backend README
RBAC	backend README § RBAC
Alpha gates	ALPHA.md, ROADMAP.md
PDF build	scripts/build-all-docs-pdf.sh