

Repository and microservice reorganization — design review (DR)

Repository & microservice split — frozen DR R1.1

Field	Value
Author	Anton Afanasyeu
Revision	R1
Creation date	2026-06-18
Last modification date	2026-06-22
Co-authored	Cursor Agent (project assistant)
Severity	high
State	frozen — superseded by SPEC R1
Document type	DR
Superseded by	docs/specs/20260618_repos_reorganizing.md

Table of contents

1. Executive summary
2. Problem statement
3. PO decisions (locked)
4. Current state
 - 4.1 Monolith map
 - 4.2 What is not a git repo
 - 4.3 Git submodules (monolith — only these)
5. Target architecture
6. Git repository catalog
 - 6.1 Structure rationale (R1)
 - 6.2 Repo creation waves (git server)
7. Public URL map
 - 7.1 Rules (R1 — PO locked)
 - 7.2 Target path catalog (one URI per microproject)
 - 7.3 Migration table (legacy → target)
8. Dependency graphs
 - 8.1 End-to-end — clients, edge, services
 - 8.2 Backend UI → microservice (requires)
 - 8.3 Platform and deploy layer
 - 8.4 Service dependency matrix
 - 8.5 Repo creation waves
9. Monolith conversion steps
 - Step 1 — Git org ac on f0xx.org server (no code move)
 - Step 2 — Extract zero-coupling repos
 - Step 3 — Platform libraries (foundation)
 - Step 4 — Microservice template and OpenAPI baseline
 - Step 5 — Edge proxy (routing shell)
 - Step 6 — Identity microservice (GATE)
 - Step 7 — RBAC and devices microservices
 - Step 8 — Issues and tickets (domain core)
 - Step 9 — Graphs and remote access
 - Step 10 — Hub, access UI, URL migration
 - Step 11 — Build and OTA pipeline
 - Step 12 — Optional workspace and monolith retirement
 - Step 13 — Notifications and cloud hardening (post-alpha)
 - Conversion step summary (quick reference)
10. Microservice vs monolith comparison
11. Infrastructure and cloud
12. ac-workspace and OTA versioning
13. Risks and mitigations
- Appendix A — Git disk paths and remotes
- Appendix B — Optional HTTPS read-only git access
 - 1) DNS
 - 2) Install packages (git server — example: Gentoo/Alpine)
 - 3) Issue free TLS certificates (Certbot)
 - 4) nginx — HTTPS smart HTTP, read-only
 - 5) Mark public repos for anonymous HTTPS read
 - 6) Optional — require token for private HTTPS read
 - 7) Token-wise permissions (outside AndroidCast)
 - 8) Certbot renewal (crontab)
 - 9) Client remote examples
 - 10) Verification checklist
14. Changelog

Source: 20260618_repos_reorganizing.md — section links work in PDF viewers that support internal anchors.

Document type: DR (Design Review) — **FROZEN**

Specification: docs/specs/20260618_repos_reorganizing.md (normative implementation doc)

Source draft: docs/drafts/20260618_repos_reorganizing.txt

PDF: 20260618_repos_reorganizing.pdf · Regenerate: bash scripts/build-all-docs-pdf.sh

Status: Frozen R1.1 (2026-06-22) — accepted; **do not edit** except typo fixes; implement per SPEC

Scope: Split git://f0xx.org/android_cast monolith into git org ac on the **f0xx.org git server** — full microservice end-state, VM-first cloud path

Related: INFRA.md · BUILD_DEPLOY.md · specs/20100612_1_scaling.md · orchestration/sim/cluster0/ARCHITECTURE.md

VCS clarification (R1): Canonical remotes are git://f0xx.org/ac/<repo> — repos are **created and owned on the PO's git server** (bare repos, hooks, ACLs). **Gitea** at .../git/ is a **read-only browse UI** over those mirrors, not the system of record and **not** where new repos are provisioned. Ops mirror scripts (examples/crash_reporter/backend/scripts/gitea/) sync **server** → **Gitea** for HTML UI only.

1. Executive summary

Question: How should the androidcast monolith (git://f0xx.org/android_cast) be split into independent repos and deployable microservices?

Short answer: Create git org ac on the **f0xx.org git server** with **platform libraries**, **microservice APIs**, **thin backend UIs**, **client repos**, and ac-deploy (submodules ac-scripts). End-state is **full microservice** architecture. Delivery is **strictly ordered** — [§9 Monolith conversion steps](#) — with ac-ms-identity **as the gate** before any domain service split.

DR decision (R1):

Item	Decision
Git org	ac on f0xx.org git server — git://f0xx.org/ac/<repo>
Gitea	Browse-only UI at .../git/; not repo provisioning; PO creates bare repos on git server
Architecture	Full microservice end-state; phased migration from monolith
URL prefix	/app/androidcast_project/ (full path, locked)
Path-per-microproject	One URI root per surface — /issues/, /tickets/, /graphs/, ... — no /crashes/ and no ?view= in target URLs
Scripts	ac-scripts separate; ac-deploy consumes it (submodule or PATH)
Workspace	Optional ac-workspace — selective clone only; not build-all-in-one
Identity gate	Step 6 (ac-ms-identity) before domain MS extraction
Repo catalog (§6)	Approved — PO minor naming deltas acceptable; structure kept as R1 table
Cloud	VM lift-and-shift first; K8s optional later
Alpha	Monolith may run in prod until Step 8+; no forced big-bang

2. Problem statement

The monolith bundles Android app, PHP backend (crashes, tickets, graphs, RBAC, remote access), builder, hub, orchestration, docs, and one submodule (url-shortener). This blocks:

- Independent CI/CD and rollback per service
- Clear ownership and OpenAPI contracts
- Mapping repos to VM roles (scaling SPEC §6.3)
- Product URLs such as issues vs tickets — **distinct path roots**, not ?view= under a shared /crashes/ console

Today's coupling: **one MariaDB**, **one session cookie** (ac_crash_sess, path /app/androidcast_project), **shared Auth.php** across consoles (BUILD_DEPLOY.md).

3. PO decisions (locked)

#	Topic	Decision
1	Git org	ac on f0xx.org git server (bare repos); Gitea = browse UI only
2	Architecture	Full microservice (phased delivery)
3	ac-scripts	Separate repo; ac-deploy utilizes ac-scripts
4	Public URLs	One path root per microproject under /app/androidcast_project/
5	ac-workspace	Optional manifest; not unified build; supports multi-repo OTA
6	Cloud	VM lift-and-shift; no K8s requirement at R1
7	Identity gate	Approved — identity MS before other MS splits
8	Legacy /crashes/	301 only during migration; removed from docs, hub, mobile defaults
9	?view= routing	Forbidden in target state — each console is its own nginx location + repo

10	Git submodules	Only third-party/* and backend/url-shortener — no other nested git trees in product repos
----	----------------	--

4. Current state

4.1 Monolith map

Monolith path	Future repo(s)	Notes
app/, ndk/, gradle/	ac-mobile-android	third-party submodules
desktop/session-studio/	ac-session-studio	Low coupling
docs/	ac-docs	
scripts/	ac-scripts	
orchestration/	ac-deploy	cluster0, docker
examples/crash_reporter/backend/	ac-ms-* + ac-be-*	32 PHP classes , 24 APIs
examples/build_console/	ac-ms-build, ac-be-builder	Same DB users
examples/app_hub/	ac-be-hub	Loads shared platform-web assets (not /crashes/assets/)
backend/url-shortener/	ac-ms-url-shortener	Existing submodule

4.2 What is not a git repo

- **Gitea** — read-only web UI over git mirrors on BE (:3000); **not** where repos are created
- **MariaDB**, **Janus** (:8089)

4.3 Git submodules (monolith — only these)

Path	Role after split
third-party/*	Native/codecs deps (submodules of ac-mobile-android)
backend/url-shortener/	URL shortener → ac-ms-url-shortener

No other directories in the monolith carry git submodules. Post-split, **ac-deploy** may submodule **ac-scripts** (repo-to-repo pin, not monolith legacy).

5. Target architecture

Platform (Composer / static — not runtime MS)
ac-platform-php, ac-platform-db, ac-platform-web, ac-scripts, ac-docs

Infrastructure
ac-deploy submodules ac-scripts; docker, cluster0, nginx
ac-platform-edge nginx BFF: routes + session cookie domain
ac-workspace optional clone manifest (no unified build)

Clients
ac-mobile-android, ac-mobile-ios (future), ac-session-studio

Microservices (OpenAPI)
ac-ms-identity → ac-ms-rbac → ac-ms-devices
ac-ms-issues, ac-ms-tickets, ac-ms-graphs, ac-ms-remote-access
ac-ms-url-shortener, ac-ms-build, ac-ms-ota, ac-ms-notifications (later)

Backend UI (thin → MS APIs)
ac-be-hub, ac-be-issues, ac-be-tickets, ac-be-graphs,
ac-be-remote-access, ac-be-access, ac-be-builder, ac-be-auth

Consolidations vs initial PO sketch:

Initial name	R0 decision
ac-be-console (monolith)	Dissolved into ac-be-issues ... ac-be-access
ac-ms-tracker	ac-ms-issues + ac-ms-tickets
ac-ms-vpn-rssh + ac-ms-vpn-wireguard	ac-ms-remote-access (single control plane)
ac-ms-analytics	ac-ms-graphs (+ BI later)
ac-ms-orchestration	ac-ms-build + ac-deploy
ac-ms-vcs	Not a repo — Gitea is infra/UI only; git server is source of truth

6. Git repository catalog

Canonical remote: git://f0xx.org/ac/<repo> — PO provisions **bare repos on the git server** (org namespace ac).

Browse UI (optional): https://apps.f0xx.org/app/androidcast_project/git/ac/<repo> — Gitea mirror for humans; push/fetch use git server.

Legacy: git://f0xx.org/android_cast → read-only mirror until retired.

6.1 Structure rationale (R1)

The catalog below is the **recommended end-state** for this project:

- ****Platform libs (P*)**** before microservices — avoids auth/DB duplication across 10+ PHP trees.
- ****Thin BE UI (B*) separate from MS APIs (S*) — matches PO's one public URI per microproject**** (§7).
- **ac-deploy + ac-scripts** as the only deploy orchestration entry — fits VM roles in scaling SPEC.
- **No ac-ms-vcs repo** — VCS is infrastructure, not application code.

PO's initial sketch used deeper folder names (ac-backend/*, ac-microservice/*). **R1 keeps the flatter ac-<name> list** — easier permissions, clone URLs, and OTA manifest entries. Minor renames are acceptable; **dependency order and URI split are locked**.

ID	Repo	Git URL	Role
W0	ac-workspace	git://f0xx.org/ac/ac-workspace	Optional submodule manifest
P0	ac-platform-php	git://f0xx.org/ac/ac-platform-php	Composer: auth client, PDO, HTTP
P1	ac-platform-db	git://f0xx.org/ac/ac-platform-db	SQL migrations
P2	ac-platform-web	git://f0xx.org/ac/ac-platform-web	Shared CSS/JS/theme
P3	ac-platform-edge	git://f0xx.org/ac/ac-platform-edge	nginx routes + proxy
P4	ac-scripts	git://f0xx.org/ac/ac-scripts	CI, OTA, native codecs, PDF
P5	ac-docs	git://f0xx.org/ac/ac-docs	Documentation
D0	ac-deploy	git://f0xx.org/ac/ac-deploy	docker, cluster0; uses P4
C0	ac-mobile-android	git://f0xx.org/ac/ac-mobile-android	Android + ndk + third-party
C1	ac-mobile-ios	git://f0xx.org/ac/ac-mobile-ios	Future iOS
C2	ac-session-studio	git://f0xx.org/ac/ac-session-studio	Desktop analyzer
S0	ac-ms-template	git://f0xx.org/ac/ac-ms-template	New MS cookiecutter
S1	ac-ms-identity	git://f0xx.org/ac/ac-ms-identity	Users, login, 2FA, sessions
S2	ac-ms-rbac	git://f0xx.org/ac/ac-ms-rbac	Companies, privileges
S3	ac-ms-devices	git://f0xx.org/ac/ac-ms-devices	Device registry
S4	ac-ms-issues	git://f0xx.org/ac/ac-ms-issues	Crash/report ingest
S5	ac-ms-tickets	git://f0xx.org/ac/ac-ms-tickets	Ticket workflow
S6	ac-ms-graphs	git://f0xx.org/ac/ac-ms-graphs	Graph ingest/query
S7	ac-ms-remote-access	git://f0xx.org/ac/ac-ms-remote-access	WG + RSSH API
S8	ac-ms-url-shortener	git://f0xx.org/ac/ac-ms-url-shortener	Short links API
S9	ac-ms-build	git://f0xx.org/ac/ac-ms-build	Build worker API
S10	ac-ms-ota	git://f0xx.org/ac/ac-ms-ota	Channel manifests
S11	ac-ms-notifications	git://f0xx.org/ac/ac-ms-notifications	SMTP (post-alpha)
B0	ac-be-hub	git://f0xx.org/ac/ac-be-hub	Landing
B1	ac-be-issues	git://f0xx.org/ac/ac-be-issues	Issues UI
B2	ac-be-tickets	git://f0xx.org/ac/ac-be-tickets	Tickets UI
B3	ac-be-graphs	git://f0xx.org/ac/ac-be-graphs	Graphs UI
B4	ac-be-remote-access	git://f0xx.org/ac/ac-be-remote-access	RA admin UI
B5	ac-be-access	git://f0xx.org/ac/ac-be-access	RBAC admin UI

B6	ac-be-builder	git://f0xx.org/ac/ac-be-builder	Builder UI
B7	ac-be-auth	git://f0xx.org/ac/ac-be-auth	Login/register shell
F1	ac-ms-sfu-signaling	git://f0xx.org/ac/ac-ms-sfu-signaling	Future F1
F2	ac-ms-media-transcode	git://f0xx.org/ac/ac-ms-media-transcode	Future VOD

Third-party codecs remain **submodules of C0** (upstream URLs); optional **git-server mirrors** under ac/ for backup (sync scripts, not Gitea-as-SoT).

6.2 Repo creation waves (git server)

PO creates bare repos on the **git server** in waves (Gitea mirror sync follows). Agent/docs do **not** “create repos in Gitea.”



7. Public URL map

Base prefix (locked): https://apps.f0xx.org/app/androidcast_project

7.1 Rules (R1 — PO locked)

- No /crashes/ in target URLs** — legacy path **301** → correct microproject root for ≥ one release, then drop from nginx/docs/mobile.
- No ?view= routing** — each product surface has its **own path root** and (eventually) its **own BE UI repo + nginx** location.
- Auth at project root** — /login, /logout, /register, /two-factor, /verify-email (not under /issues/).
- Shared cookie path** — /app/androidcast_project at **ac-platform-edge** until SSO tokens replace shared PHP session.

7.2 Target path catalog (one URI per microproject)

Microproject	Target URL root	Repo (UI / API)	Notes
Hub / landing	.../	ac-be-hub	Entry cards link to path roots only
Auth	.../login, .../logout, ...	ac-be-auth → ac-ms-identity	Project-root paths
Issues (reports)	.../issues/	ac-be-issues / ac-ms-issues	Was ?view=reports
Issue detail	.../issues/{id}	ac-be-issues	Was ?view=report&id=
Tickets	.../tickets/	ac-be-tickets / ac-ms-tickets	Was ?view=tickets
Ticket detail	.../tickets/{id}	ac-be-tickets	Was ?view=ticket&id=
Analytics / graphs	.../graphs/	ac-be-graphs / ac-ms-graphs	Already separate
RBAC / access	.../access/	ac-be-access / ac-ms-rbac	Was ?view=rbac
Remote access	.../remote-access/	ac-be-remote-access / ac-ms-remote-access	Was ?view=remote_access
Short links	.../short-links/	ac-be-hub or thin UI / ac-ms-url-shortener	Was ?view=short_links
Live sessions	.../live-sessions/	ac-be-issues (or ac-be-live) / live_cast API	Was ?view=live_sessions
Live join / education	.../live/join, .../live/education	shared live pages	Under issues until split
Builder	.../build/	ac-be-builder / ac-ms-build	Unchanged
Git browse	.../git/	Gitea UI (mirror)	Not a microservice repo
OTA	/v0/ota/	ac-ms-ota	Top-level on apps host

API examples: .../issues/api/upload (not .../crashes/api/upload.php).

7.3 Migration table (legacy → target)

Legacy (forbidden in target)	Target URL
.../crashes/?view=reports	.../issues/
.../crashes/?view=report&id=N	.../issues/N
.../crashes/?view=tickets	.../tickets/
.../crashes/?view=ticket&id=N	.../tickets/N
.../crashes/?view=home	.../issues/ or .../ (hub)
.../crashes/?view=rbac	.../access/
.../crashes/?view=remote_access	.../remote-access/
.../crashes/?view=short_links	.../short-links/
.../crashes/?view=live_sessions	.../live-sessions/
.../crashes/?view=graphs	.../graphs/

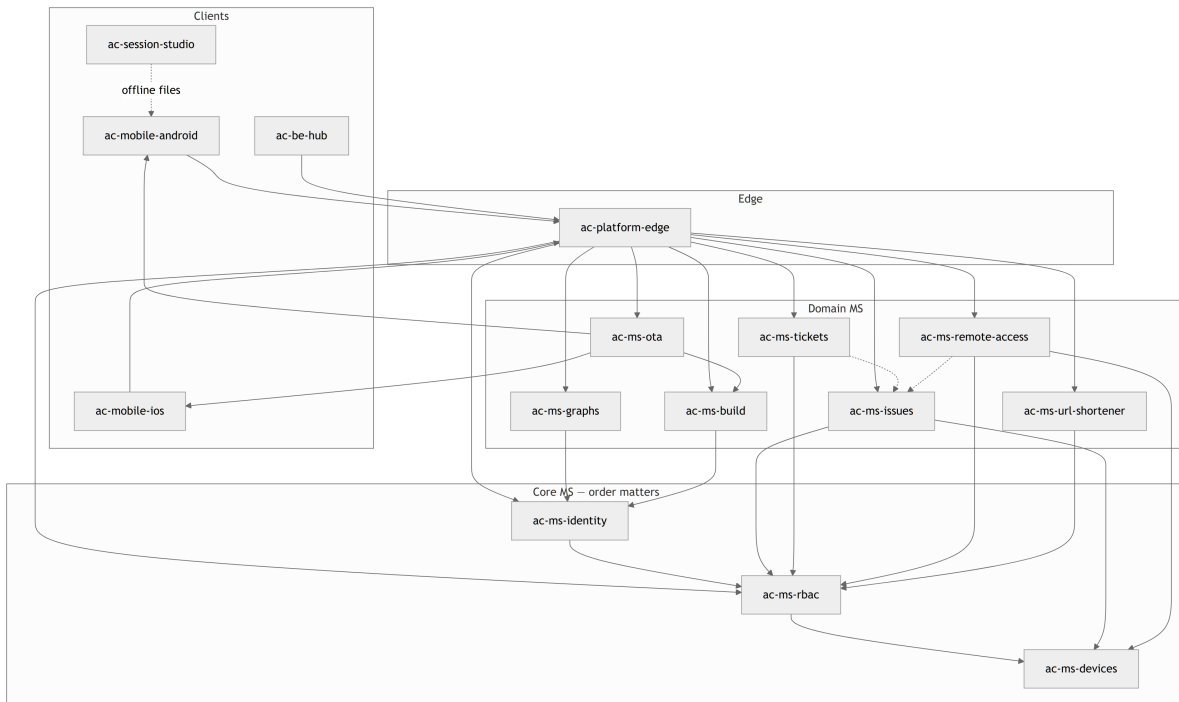
.../crashes/login	.../login
.../crashes/api/upload.php	.../issues/api/upload
.../crashes/ (any)	301 → matching row above

Monolith shim (Step 5 only): nginx may **internally** rewrite /issues/ → legacy PHP ?view=reports until ac-ms-issues exists — **not** exposed in links, docs, or mobile defaults.

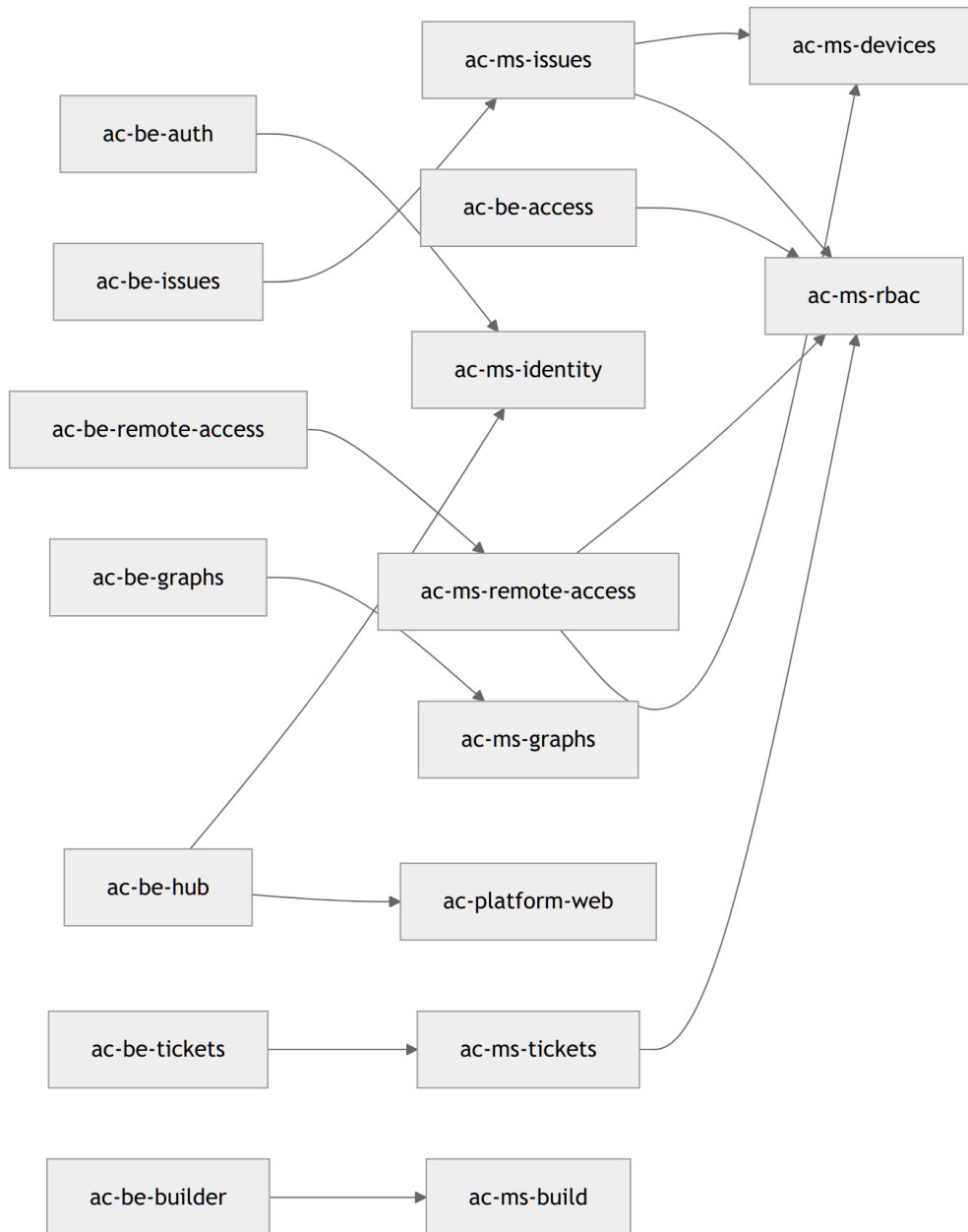
Code updates: BackendEndpoints.java, hub index.php, all console nav, OpenAPI base paths, deploy nginx fragments.

8. Dependency graphs

8.1 End-to-end — clients, edge, services

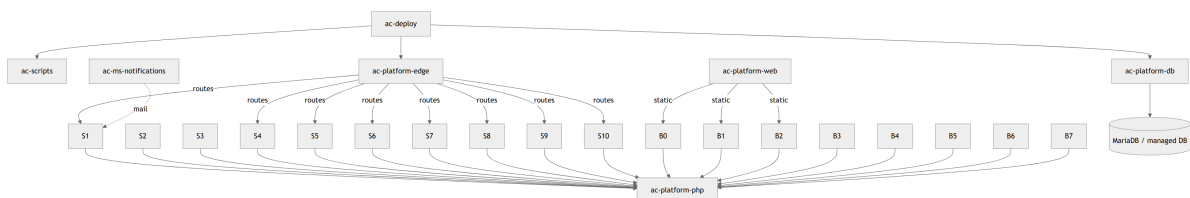


8.2 Backend UI → microservice (requires)



Every **B*** and **S*** also requires **P0** (ac-platform-php) and **P1** (ac-platform-db) via Composer — omitted from diagram for clarity.

8.3 Platform and deploy layer



8.4 Service dependency matrix

Service	Requires (hard)	Requires (soft)	Required by
ac-platform-db	—	—	All MS, migration jobs
ac-platform-php	ac-platform-db	—	All MS, all BE UI
ac-platform-web	—	—	ac-be-hub, all BE UI
ac-platform-edge	ac-deploy nginx templates	all MS upstream addrs	All public HTTP
ac-ms-identity	platform-db, platform-php	ac-ms-notifications	All MS, all BE UI, ac-ms-build
ac-ms-rbac	ac-ms-identity	—	devices, issues, tickets, RA, url-shortener
ac-ms-devices	ac-ms-rbac	ac-ms-identity	ac-ms-issues, ac-ms-remote-access

ac-ms-issues	ac-ms-rbac, ac-ms-devices	—	ac-be-issues, mobile upload
ac-ms-tickets	ac-ms-rbac	ac-ms-issues	ac-be-tickets
ac-ms-graphs	ac-ms-identity	—	ac-be-graphs, mobile
ac-ms-remote-access	ac-ms-rbac, ac-ms-devices	ac-ms-issues	ac-be-remote-access, mobile
ac-ms-url-shortener	ac-ms-rbac	—	hub/admin UI
ac-ms-build	ac-ms-identity	git server mirrors	ac-be-builder, ac-ms-ota
ac-ms-ota	ac-ms-build	C0, C1 SHAs	mobile OTA clients
ac-mobile-android	ac-ms-issues, ac-ms-graphs, ac-ms-remote-access APIs	ac-ms-ota	—

8.5 Repo creation waves

See §6.2 for the ordered wave diagram.

9. Monolith conversion steps

Ordered steps for converting the monolith. **Do not skip gates.** Each step lists repos created/moved, monolith paths affected, and verification.

Step 1 — Git org `ac` on `f0xx.org` server (no code move)

Goal: Namespace `ac` exists on the **git server**; legacy monolith mirror preserved; Gitea browse UI synced.

Action	Detail
Create org / paths	PO: bare repos under <code>git://f0xx.org/ac/<repo></code> on git server
Mirror	<code>android_cast</code> → read-only; plan <code>ac-workspace</code> or retire later
Gitea (optional)	Sync mirrors for <code>.../git/</code> browse UI only — not repo creation
Update docs	Pointer in <code>INFRA.md</code> , <code>AGENTS.md</code>

Verify: `git ls-remote git://f0xx.org/ac/ac-docs` (after Step 2) ; legacy push still works.

Requires: nothing

Enables: all following steps

Step 2 — Extract zero-coupling repos

Goal: Repos with no runtime dependency on PHP monolith.

Repo	From monolith	CI
ac-docs	<code>docs/</code>	PDF build
ac-scripts	<code>scripts/</code>	lint/smoke
ac-session-studio	<code>desktop/session-studio/</code>	unit tests
ac-ms-url-shortener	<code>backend/url-shortener/</code>	PHPUnit (change remote URL only if already split)
ac-deploy	<code>orchestration/</code> + <code>nginx seeds</code>	docker compose smoke

ac-deploy submodules **ac-scripts** at pinned SHA.

Verify: Each repo builds independently; monolith submodule paths updated OR mirror dual-push during transition.

Requires: Step 1

Enables: Steps 3–4

Step 3 — Platform libraries (foundation)

Goal: Shared code leaves monolith as Composer packages — **no MS split yet.**

Repo	Extract from
ac-platform-db	<code>sql/</code> , migrations from <code>crash_reporter</code> + <code>url-shortener</code>
ac-platform-php	<code>Auth*</code> , <code>Database</code> , <code>Rbac</code> helpers, bootstrap patterns, <code>shared_session.php</code>
ac-platform-web	<code>crashes/assets/</code> , shared theme referenced by hub

Monolith **requires** packages via Composer; behavior unchanged.

Verify: Monolith tests green; single login still works across `hub/crashes/build`.

Requires: Step 2

Enables: Step 4 (identity)

Step 4 — Microservice template and OpenAPI baseline

Goal: Cookiecutter + contract layout for all MS.

Repo	Action
ac-ms-template	Copy <code>url-shortener</code> + <code>platform-php</code> wiring
ac-docs	Add <code>openapi/</code> stubs per service

Verify: Scaffold new empty MS from template; deploy locally via `ac-deploy`.

Requires: Step 3

Enables: Step 5

Step 5 — Edge proxy (routing shell)

Goal: **ac-platform-edge** nginx config routes paths; still proxies to **monolith** PHP until MS exist.

Action	Detail
Deploy edge config	Map /issues/ → monolith ?view=reports internally (temporary)
Cookie	Single path /app/androidcast_project

Verify: `curl -I` new paths return 200 via rewrite; old URLs still work.

Requires: Step 2 (ac-deploy)

Enables: Step 10 URL cutover

Step 6 — Identity microservice (GATE)

Goal: **ac-ms-identity** owns users, login, register, 2FA, session issuance.

Repo	From monolith
ac-ms-identity	Auth*, UserRepository, AuthTotp*, AuthRegistration*, ...
ac-be-auth	login/register views (thin)

Monolith/builder **call identity** for auth (HTTP or shared session bridge during transition).

Verify: Login at /app/androidcast_project/ + builder; sessions valid; `./scripts/test_rbac_api.sh` still passes against bridged auth.

Requires: Steps 3, 4

Enables: Steps 7, 9, 11 — **nothing else splits before this**

Step 7 — RBAC and devices microservices

Goal: Company scope and device registry are independent APIs.

Repo	From monolith
ac-ms-rbac	Rbac*, RbacAdmin*, companies API
ac-ms-devices	DeviceRepository

Verify: Upload still registers device; RBAC panel works via APIs.

Requires: Step 6

Enables: Steps 8, 9

Step 8 — Issues and tickets (domain core)

Goal: Split PO's primary URL surfaces.

Repo	From monolith
ac-ms-issues	upload, Report*, Tag*, reports API
ac-ms-tickets	Ticket*, attachments
ac-be-issues	reports views + JS
ac-be-tickets	tickets views + JS

Edge routes /issues/ → S4/B1, /tickets/ → S5/B2.

Verify: Mobile upload to new URL; web issue/ticket lists; E2E ticket workflow.

Requires: Step 7

Enables: Step 9 (RA links to issues)

Step 9 — Graphs and remote access

Repo	From monolith
ac-ms-graphs	GraphRepository, graph_* API
ac-ms-remote-access	RemoteAccess*, WireGuard*, Rssh*
ac-be-graphs	graphs UI
ac-be-remote-access	remote_access view

Verify: BackendEndpoints graph URL; RA E2E (`ra_e2e_cli.sh`); mobile RA API.

Requires: Steps 7, 8 (RA → devices, issue links)

Enables: Step 11

Step 10 — Hub, access UI, URL migration

Repo	Action
ac-be-hub	Extract app_hub; use ac-platform-web assets
ac-be-access	RBAC UI → calls ac-ms-rbac
ac-platform-edge	Enable 301 from /crashes/ and ?view=* to S7 path roots

Verify: Hub cards link to /issues/, /tickets/; no broken CSS.

Requires: Steps 5, 8, 9

Enables: Step 11

Step 11 — Build and OTA pipeline

Repo	From monolith
ac-ms-build	BuildRunner, build APIs
ac-be-builder	build_console UI
ac-ms-ota	OTA manifest assembly
ac-mobile-android	app/, ndk/, gradle (extract from monolith)

Verify: Builder triggers APK; OTA channel publishes; manifest lists android (+ ios when C1 exists).

Requires: Step 6 (identity for builder users)

Enables: Step 12

Step 12 — Optional workspace and monolith retirement

Action	Detail
ac-workspace	Publish optional .gitmodules manifest
Retire	Stop commits to android_cast monolith; archive mirror
Gitea mirror sync	Update mirror scripts for org ac (server → Gitea UI; gitea README)

Verify: Fresh clone via ac-workspace or individual repos; prod deploy from ac-deploy only.

Requires: Steps 1–11 complete for services in scope

Enables: cloud VM scaling, future F1/F2 MS

Step 13 — Notifications and cloud hardening (post-alpha)

Repo	Action
ac-ms-notifications	Extract AuthMailer; wire 2.x mail when unblocked
ac-deploy	Managed DB DSN templates; cluster0 → cloud VM roles

Requires: Step 6; mail infra (developer)

Enables: full 2FA mail E2E

Conversion step summary (quick reference)

Step	Name	Gate
1	Git org ac on server	—
2	docs, scripts, deploy, session-studio, url-shortener	—
3	platform-php, platform-db, platform-web	—
4	ms-template, OpenAPI	—
5	platform-edge (shell)	—
6	ms-identity , be-auth	GATE
7	ms-rbac, ms-devices	after 6
8	ms-issues, ms-tickets, be-issues, be-tickets	after 7
9	ms-graphs, ms-remote-access, be UI	after 7–8
10	be-hub, be-access, URL 301	after 8–9
11	ms-build, ms-ota, mobile-android	after 6
12	ac-workspace, retire monolith	after 1–11
13	notifications, cloud	post-alpha

10. Microservice vs monolith comparison

Criterion	Full microservice (PO choice)	Modular monolith
Independent deploy	Yes — per MS/UI repo	Single PHP deploy
Failure isolation	Stronger	Shared FPM pool
Auth complexity	Needs identity MS + edge early	Works today
URL clarity	/issues/, /tickets/, ... — one URI per microproject	?view= under /crashes/
Ops load	More nginx, repos, logs	One sync path (INFRA.md)
Alpha risk	High if before Step 6	Lower
Cloud mapping	1:1 to VM roles	Lift entire tree
Rollback	Revert one repo SHA	Revert one commit

PO choice is approved with **Step 6 gate** enforced in [§9](#). **R1:** URL model is **path-per-microproject** (§7), not query-parameter routing.

11. Infrastructure and cloud

Stage	Pattern
Prod today	FE TLS → BE :80 nginx + PHP-FPM + MariaDB
cluster0 lab	cast01–03; NFS config only; GTID DB (ARCHITECTURE.md)
Cloud phase A	4–6 VMs: be-web, be-data, be-ops, be-build (scaling §6.3)
Cloud phase B	Managed DB; DSN in secrets (not in git)
Cloud phase C	K8s optional — not required at R0

Inter-service calls: HTTP on private VLAN / localhost initially; OpenAPI in ac-docs. No service mesh at R0.

ac-deploy ships Ansible/docker-compose per VM role; submodules **ac-scripts**.

12. ac-workspace and OTA versioning

ac-workspace — optional; lists repo SHAs for a release train. **Not** a unified Gradle/PHP build.

ac-ms-ota manifest (same product version, per-platform SHAs):

```
{
  "channel": "staging",
  "version": "00.02.00.00",
  "artifacts": {
    "android": { "repo": "ac-mobile-android", "git_sha": "abc123" },
    "ios": { "repo": "ac-mobile-ios", "git_sha": "def456" }
  }
}
```

13. Risks and mitigations

Risk	Mitigation
Auth split breaks all consoles	Step 6 gate; session bridge during transition
Hub broken CSS	ac-platform-web before ac-be-hub (Step 3 before 10)
Mobile upload URL change	Parallel endpoints + BackendEndpoints normalization
DB schema drift	ac-platform-db single migration stream
Too many repos to bump	Composer semver for P0; not @common submodule diamond
Alpha blocked	Monolith stays prod until Step 8+ complete per service
PO BE learning curve	ac-ms-template, ac-deploy, OpenAPI, localhost HTTP

Appendix A — Git disk paths and remotes

Convention: disk paths are **relative to the git server root** (example: /var/git/). PO may choose another root; keep the ac/<repo> layout.

Canonical clone/fetch (write + read): git://f0xx.org/ac/<repo>

HTTPS read-only (optional): see [Appendix B](#) — separate vhost; **not** https://f0xx.org/... app URLs.

#	Disk path (relative)	Git repo URL
1	ac/ac-workspace	git://f0xx.org/ac/ac-workspace
2	ac/ac-platform-php	git://f0xx.org/ac/ac-platform-php
3	ac/ac-platform-db	git://f0xx.org/ac/ac-platform-db
4	ac/ac-platform-web	git://f0xx.org/ac/ac-platform-web
5	ac/ac-platform-edge	git://f0xx.org/ac/ac-platform-edge
6	ac/ac-scripts	git://f0xx.org/ac/ac-scripts
7	ac/ac-docs	git://f0xx.org/ac/ac-docs
8	ac/ac-deploy	git://f0xx.org/ac/ac-deploy
9	ac/ac-mobile-android	git://f0xx.org/ac/ac-mobile-android
10	ac/ac-mobile-ios	git://f0xx.org/ac/ac-mobile-ios
11	ac/ac-session-studio	git://f0xx.org/ac/ac-session-studio
12	ac/ac-ms-template	git://f0xx.org/ac/ac-ms-template
13	ac/ac-ms-identity	git://f0xx.org/ac/ac-ms-identity
14	ac/ac-ms-rbac	git://f0xx.org/ac/ac-ms-rbac
15	ac/ac-ms-devices	git://f0xx.org/ac/ac-ms-devices
16	ac/ac-ms-issues	git://f0xx.org/ac/ac-ms-issues
17	ac/ac-ms-tickets	git://f0xx.org/ac/ac-ms-tickets
18	ac/ac-ms-graphs	git://f0xx.org/ac/ac-ms-graphs
19	ac/ac-ms-remote-access	git://f0xx.org/ac/ac-ms-remote-access

20	ac/ac-ms-url-shortener	git://f0xx.org/ac/ac-ms-url-shortener
21	ac/ac-ms-build	git://f0xx.org/ac/ac-ms-build
22	ac/ac-ms-ota	git://f0xx.org/ac/ac-ms-ota
23	ac/ac-ms-notifications	git://f0xx.org/ac/ac-ms-notifications
24	ac/ac-be-hub	git://f0xx.org/ac/ac-be-hub
25	ac/ac-be-issues	git://f0xx.org/ac/ac-be-issues
26	ac/ac-be-tickets	git://f0xx.org/ac/ac-be-tickets
27	ac/ac-be-graphs	git://f0xx.org/ac/ac-be-graphs
28	ac/ac-be-remote-access	git://f0xx.org/ac/ac-be-remote-access
29	ac/ac-be-access	git://f0xx.org/ac/ac-be-access
30	ac/ac-be-builder	git://f0xx.org/ac/ac-be-builder
31	ac/ac-be-auth	git://f0xx.org/ac/ac-be-auth
32	ac/ac-ms-sfu-signaling	git://f0xx.org/ac/ac-ms-sfu-signaling
33	ac/ac-ms-media-transcode	git://f0xx.org/ac/ac-ms-media-transcode
34	android_cast	git://f0xx.org/android_cast

Bare repo on server (PO): `git init --bare /var/git/ac/ac-workspace` (repeat per row; path = {GIT_ROOT}/ + disk path).

Appendix B — Optional HTTPS read-only git access

Scope: Standalone **git infrastructure** on the PO git host. **Not** part of the AndroidCast monorepo, BE PHP tree, Gitea product UI, or apps.f0xx.org consoles.

Goal: Keep `git://f0xx.org/...` for normal push/fetch; add `https://` for **read-only** clones (anonymous and/or token), with TLS certs and ACL files **outside** the AndroidCast project.

Suggested hostname: `git-r.f0xx.org` (read-only smart HTTP). Do **not** reuse `https://f0xx.org/` landing or `https://apps.f0xx.org/` app paths for git smart HTTP.

Suggested paths on git server:

Path	Purpose
/var/git/	Bare repos (Appendix A disk layout)
/etc/git-ro/	nginx snippets, token maps, install scripts — not in android_cast repo
/etc/git-ro/tokens/	Per-token permission files (see step 8)
/etc/letsencrypt/live/git-r.f0xx.org/	Certbot certificates

1) DNS

Add an **A** (or **AAAA**) record for the read-only git vhost pointing at the **git server** public or edge IP (same host that serves `git://` today, or FE DNAT to it):

```
git-r.f0xx.org. IN A <GIT_SERVER_PUBLIC_IP>
```

Verify:

```
dig +short git-r.f0xx.org
```

2) Install packages (git server — example: Gentoo/Alpine)

On the machine that holds `/var/git/`:

```
emerge -av nginx git fcgiwrap certbot
```

```
apk add nginx git git-daemon fcgiwrap certbot certbot-nginx openssl
```

Enable fcgiwrap (smart HTTP via git http-backend):

```
systemctl enable --now fcgiwrap
```

3) Issue free TLS certificates (Certbot)

Run on the git server (or on FE if it terminates TLS and proxies to git — adjust plugin):

```
certbot certonly --standalone -d git-r.f0xx.org \
--agree-tos -m admin@f0xx.org --non-interactive
```

If nginx is already bound to :80/443, use the nginx plugin instead:

```
certbot certonly --nginx -d git-r.f0xx.org \
--agree-tos -m admin@f0xx.org --non-interactive
```

Certificates land under `/etc/letsencrypt/live/git-r.f0xx.org/fullchain.pem` and `privkey.pem`.

4) nginx — HTTPS smart HTTP, read-only

Create `/etc/git-ro/nginx-git-ro.conf` (outside AndroidCast):

```
server {
    listen 443 ssl;
```

```

server_name git-r.f0xx.org;

ssl_certificate      /etc/letsencrypt/live/git-r.f0xx.org/fullchain.pem;
ssl_certificate_key  /etc/letsencrypt/live/git-r.f0xx.org/privkey.pem;

root /var/git;

location ~ ^/ac/(.+?)\.git?/(HEAD|info/refs|objects/info/.+|objects/([0-9a-f]{2}/[0-9a-f]{38}|pack/pack-.*))$ {
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME /usr/libexec/git-core/git-http-backend;
    fastcgi_param GIT_HTTP_EXPORT_ALL "";
    fastcgi_param GIT_PROJECT_ROOT /var/git;
    fastcgi_param PATH_INFO /$1.git/$2;
    fastcgi_pass unix:/run/fcgiwrap.socket;
}

location ~ ^/ac/(.+?)\.git$ {
    return 301 /ac/$1.git/info/refs?service=git-upload-pack;
}
}

```

Include it from the main nginx config:

```

ln -s /etc/git-ro/nginx-git-ro.conf /etc/nginx/conf.d/git-ro.conf
nginx -t && rc-service nginx reload # Gentoo

```

Read-only guarantee: do **not** enable receive-pack; only git-upload-pack (clone/fetch) is exposed. Push stays on git://f0xx.org (SSH or authenticated path), not this vhost.

Test anonymous clone:

```
git clone https://git-r.f0xx.org/ac/ac-docs.git /tmp/ac-docs-ro-test
```

5) Mark public repos for anonymous HTTPS read

For each bare repo that may be cloned without a token:

```

GIT_ROOT=/var/git
repo=ac/ac-docs
touch "${GIT_ROOT}/${repo}.git/git-daemon-export-ok"
git -C "${GIT_ROOT}/${repo}.git" config http.receivepack false
git -C "${GIT_ROOT}/${repo}.git" config http.uploadpack true

```

Repeat for every Appendix A row you want world-readable over HTTPS.

6) Optional — require token for private HTTPS read

Add HTTP basic auth for selected locations. Store credentials **outside** AndroidCast:

```

install -d -m 0750 /etc/git-ro/htpasswd
htpasswd -c /etc/git-ro/htpasswd/ro-ci-user

In /etc/git-ro/nginx-git-ro.conf, inside the location block for smart HTTP:
    auth_basic "git read-only";
    auth_basic_user_file /etc/git-ro/htpasswd/ro-users;

```

Clone with token (password = token):

```
git clone https://ro-ci-user:<TOKEN>@git-r.f0xx.org/ac/ac-ms-identity.git
```

7) Token-wise permissions (outside AndroidCast)

Keep a simple ACL file per token under /etc/git-ro/tokens/ — **not** referenced by android_cast code:

```

install -d -m 0700 /etc/git-ro/tokens
cat > /etc/git-ro/tokens/ci-read-issues.env <<'EOF'
ac/ac-ms-issues
ac/ac-be-issues
ac/ac-platform-php
EOF
chmod 0600 /etc/git-ro/tokens/*.env

```

Enforcement options (pick one on the git server):

- **Simple:** separate htpasswd users per token + nginx location blocks per repo prefix.
- **Scalable:** small git-ro-auth script (not in AndroidCast repo) consulted by nginx auth_request.

AndroidCast **never** reads /etc/git-ro/; CI machines hold tokens in their own secret stores.

8) Certbot renewal (crontab)

Edit root crontab on the git server:

```
crontab -e
```

Add:

```
17 3 * * * certbot renew --quiet --deploy-hook "nginx -t && rc-service nginx reload"
```

Or install a one-shot renew script /etc/git-ro/certbot-renew.sh:

```

#!/bin/sh
set -eu
certbot renew --quiet
nginx -t
rc-service nginx reload
chmod 0755 /etc/git-ro/certbot-renew.sh
crontab -e
17 3 * * * /etc/git-ro/certbot-renew.sh >> /var/log/git-ro-certbot.log 2>&1

```

9) Client remote examples

Use case	Remote URL
Developer push/fetch (canonical)	git://f0xx.org/ac/ac-mobile-android
Anonymous read (HTTPS)	https://git-r.f0xx.org/ac/ac-docs.git
Token read (HTTPS)	https://<user>:<token>@git-r.f0xx.org/ac/ac-ms-issues.git
Browse HTML (optional)	https://apps.f0xx.org/app/androidcast_project/git/ (Gitea mirror only)

Do **not** document https://f0xx.org/... as a git smart HTTP endpoint; landing stays human/PDF only.

10) Verification checklist

```
curl -sSI https://git-r.f0xx.org/ac/ac-docs.git/info/refs?service=git-upload-pack | head
```

```
git ls-remote git://f0xx.org/ac/ac-docs
```

```
git clone https://git-r.f0xx.org/ac/ac-docs.git /tmp/ro-clone-test
cd /tmp/ro-clone-test && git push origin HEAD # must fail on this vhost
```

14. Changelog

Rev	Date	Change
R0	2026-06-18	Initial DR from draft; PO locked org ac, full MS, URL prefix, identity gate; §9 conversion steps; dependency graphs
R1	2026-06-20	VCS: git server = SoT; Gitea = browse UI only — not repo provisioning. §6: structure rationale; flat ac-* catalog approved. §7: no /crashes/ or ?view= in target; full path-per-microproject catalog. PDF: removed spurious \newpage blank sheets. PO review before SPEC.
R1.1	2026-06-22	Appendix A: disk path ↔ git:// URL index table (§6 catalog + legacy). Appendix B: optional HTTPS read-only git (git-r.f0xx.org), certbot, nginx smart HTTP, token ACL outside AndroidCast.
R1.1-frozen	2026-06-22	Frozen — superseded by SPEC R1