

Platform scaling and multi-environment architecture — specification

Platform scaling — SPEC (rev.2: bandwidth + GEO)

Table of contents

1. Purpose
2. Vocabulary
3. Current state (baseline)
4. Goals and non-goals
 - Goals
 - Non-goals (this revision)
5. Requirements
 - 5.1 Must (platform)
 - 5.2 Should
 - 5.3 Future capability slots (normative intent)
6. Target architecture
 - 6.1 Environment tiers
 - 6.2 Service domains
 - 6.3 Reference cluster layout (prod)
 - 6.4 Traffic and control planes
7. Service inventory and scaling profile
8. Deployment and promotion pipeline
9. Zero-downtime and load balancing
10. Data and storage
11. Simulation lab (Alpine HVMS)
12. Capacity planning framework
 - 12.1 RSSH / remote access connections
 - 12.2 Cast and media paths (future)
 - 12.3 Network bandwidth and egress
13. Cloud migration (GCP / Azure / other)
14. Global availability and GEO policy
 - 14.1 Reach model
 - 14.2 Segment isolation
 - 14.3 Enforcement layers
15. RBAC and operator model evolution
16. Commercial and paid services (platform hooks)
17. Observability and SLO
18. Safety constraints (current topology)
19. Phased delivery map
20. Out of scope (this SPEC revision)
21. Related docs
- Changelog

Source: 20100612_1_scaling.md — section links work in PDF viewers that support internal anchors.

Document type: SPEC
Source draft: docs/drafts/20100612_1_scaling.txt
Design review: docs/DRs/20100612_1_scaling.md
PDF: 20100612_1_scaling.pdf · Regenerate: bash scripts/build-all-docs-pdf.sh (Mermaid → PNG diagrams; requires npx)
Date: 2026-06-12
Status: Planning / brainstorming (pre-implementation)
Severity: Strategic (post-alpha evolution)
Scope: FE/BE topology, service decomposition, dev/staging/prod clusters, cloud migration path

1. Purpose

Define how Android Cast evolves from a **single Alpine BE HVM** hosting all backend services into a **multi-VM, multi-environment platform** that supports:

- Stable **dev** → **staging** → **prod** promotion
 - **Horizontal scaling** and **zero-downtime** deploys where services need it
 - Optional **cloud** footprint (GCP, Azure, or hybrid) without breaking the existing Gentoo FE → Alpine BE production path
 - **Network bandwidth** budgets and egress planning alongside CPU/RAM/connection limits
 - **Global availability** with **GEO segment isolation** (worldwide reach by default; denylisted regions per content/policy — e.g. no publish in Zambia)
 - Room for **future RT/AV**, **VPN**, **payments**, and **third-party streaming** integrations described in the draft
- This SPEC is **normative for planning**; individual services retain their own SPEC/DR pairs (URL shortener, remote access, SFU, etc.).

2. Vocabulary

Term	Meaning
FE	Forward-End — Gentoo bare-metal / XEN host; TLS, public nginx, UDP DNAT
BE	Back-End — Alpine HVM(s); app services today on <code>artc0.intra.raptor.org</code>
Cluster	Named set of VMs + shared config for one environment tier (dev, staging, prod)
Control plane	HTTPS APIs, consoles, heartbeats, RBAC, orchestration metadata
Media plane	UDP/TCP cast, SFU/MCU relay, ffmpeg transcode pipelines (future)
Sysop	System operator (PO today) — owns infra simulation and promotion decisions
PROD	Production cluster serving end users
RSSH	Reverse-SSH remote access track (REMOTE_ACCESS_IMPL)
GEO segment	Country/region group (ISO 3166-1 alpha-2) used in allow/deny policy — e.g. ZM (Zambia)
geo_policy	Per-resource or catalog rule: where content/API may be served , published , or streamed
Egress budget	Planned GB/month or Mbps peak per service/link; triggers scale or CDN move

3. Current state (baseline)

Today one **BE Alpine VM** co-locates (see 20260608_BE_SERVICES_and_infra.md):

Domain	Services	Stack
Web / console	Hub, crashes, tickets, graphs, builder UI	nginx + PHP-FPM
APIs	Upload, heartbeat, remote access, graphs ingest, short links admin	PHP
Git	Gitea mirrors (AndroidCast/*)	Gitea + MariaDB/SQLite
Database	androidcast_crashes, url_shortener, Gitea DB	MariaDB
OTA	Channel JSON + artifacts	nginx static / PHP
VPN	WireGuard lab (wg0), RSSH bastion	kernel + sshd
URL shortener	<code>s.f0xx.org</code>	nginx + PHP (submodule)
Build	Docker builder orchestration	Docker on BE

Internet → router (134.17.26.161) → FE (10.7.0.10) → BE (10.7.16.128) :80

Constraint: This topology is **production truth** until a phased migration explicitly replaces it (INFRA.md §1, bottomline_reminder.txt).

4. Goals and non-goals

Goals

ID	Goal
G1	Split services by scaling profile (CPU/RAM vs storage vs RT media)
G2	Three environment tiers : dev (unstable), staging (beta), prod (stable)
G3	Seamless deploy/update with rollback per service
G4	Cost-aware sizing — no oversized DB VMs; burst CPU for build/ffmpeg
G5	Simulation lab on Alpine HVMs before cloud spend
G6	Provider L4/L7 load balancers ready for stateless tiers
G7	Preserve dual-remote git flow and existing public URLs during transition
G8	Bandwidth visibility — per-link and per-service throughput metrics before saturation
G9	Global reach for public services; segment isolation when PO denies regions for specific content

Non-goals (this revision)

ID	Non-goal
NG1	Alpha release blocker — scaling work is post-alpha unless PO reprioritizes
NG2	Immediate full cloud cutover
NG3	Replacing Gentoo FE in phase 1
NG4	Implementing payments, SFU, or DNS resale in this SPEC — only platform hooks
NG5	Full multi-region active-active prod on day one — phased via CDN → hybrid → regional PoPs
NG6	Legal counsel for country block lists — platform provides policy hooks ; PO owns compliance text

5. Requirements

5.1 Must (platform)

ID	Requirement
P1	Every deployable service has an owner , health endpoint , and config outside git documented
P2	Prod changes require promotion from staging (except hotfix path per GIT_FLOW.md)
P3	Stateless HTTP tiers scale behind a load balancer; stateful tiers (DB, object store) have backup/restore runbooks
P4	FE remains TLS termination and edge routing; BE/cluster addresses are internal until cloud edge replaces FE
P5	WireGuard UDP and RSSH TCP paths remain explicit in firewall/DNAT docs when split across VMs
P6	Simulation cluster MUST be reproducible from repo scripts (Alpine + package sets) — see §11
P7	No single VM hosts both primary MariaDB and unbounded ffmpeg transcode workers
P8	Every bandwidth-heavy service (OTA, object store, SFU, ffmpeg, CDN origin) has documented Mbps / GB-month budgets and alert thresholds
P9	geo_policy model: default global serve ; explicit deny segments (countries/regions) per content class or tenant — enforceable at edge and BE

5.2 Should

ID	Requirement
S1	Use managed MariaDB / object storage in cloud phase when cost ≤ ops time
S2	Blue/green or rolling deploy for PHP-FPM pools and Gitea
S3	Central secrets store (SOPS, Vault, or cloud SM) — no secrets in git
S4	Per-environment DNS: dev.apps..., staging.apps..., prod unchanged
S5	Connection and CPU dashboards before scaling RSSH and SFU

S6	laC fragments in repo (dep loy/, nginx snippets) for each new VM role
S7	CDN or edge cache for OTA/APK and static consoles before scaling origin VM count for bandwidth
S8	GeoDNS or latency routing when a second region/PoP is added; single canonical URL preserved

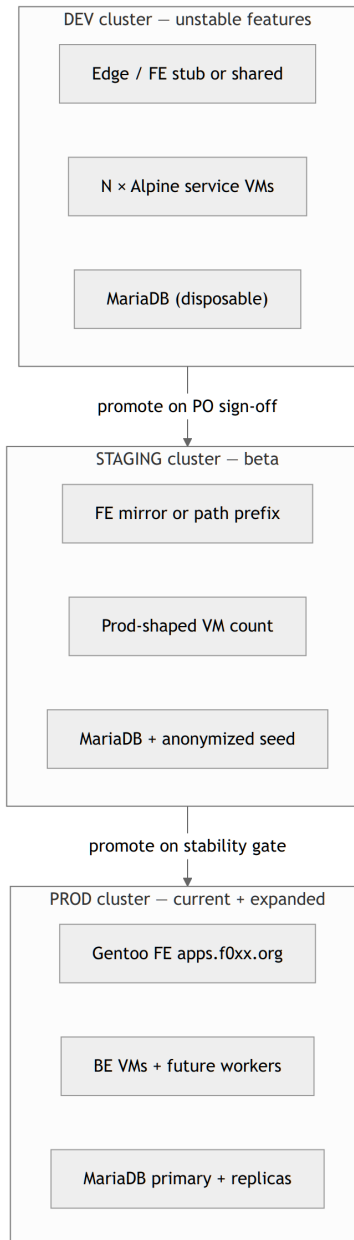
5.3 Future capability slots (normative intent)

Draft “future key roles” map to **service domains** (implementation deferred):

Slot	Description	Depends on
F1 Multipoint cast	MCU/SFU relay outside LAN; TURN/STUN	SFU SPEC, signaling
F2 Stream transcode	ffmpeg any→any; RT and non-RT	Media worker pool, object store
F3 Cast storage	Recordings on BE; paywalled retrieval	F2, billing
F4 Commercial VPN	Anonymous RT VPN; capacity planning	WG pool, payments
F5 Payments	Low fixed cost at scale; usage-based optional	F3, F4, add-ons
F6 Mobile add-ons	Masks, background replace	On-device or cloud GPU
F7 In-app transfers	P2P or ledger between users	F5, compliance
F8 DNS / subdomains	User subdomains + site builder	FE DNS API, tenant isolation
F9 3rd-party streaming	Ingest HLS/m3u8 or RTMP; re-emit as cast	F2, cast handlers
F10 GEO enforcement	Region deny/allow on publish, stream, download	F3, F5, edge CDN/WAF

6. Target architecture

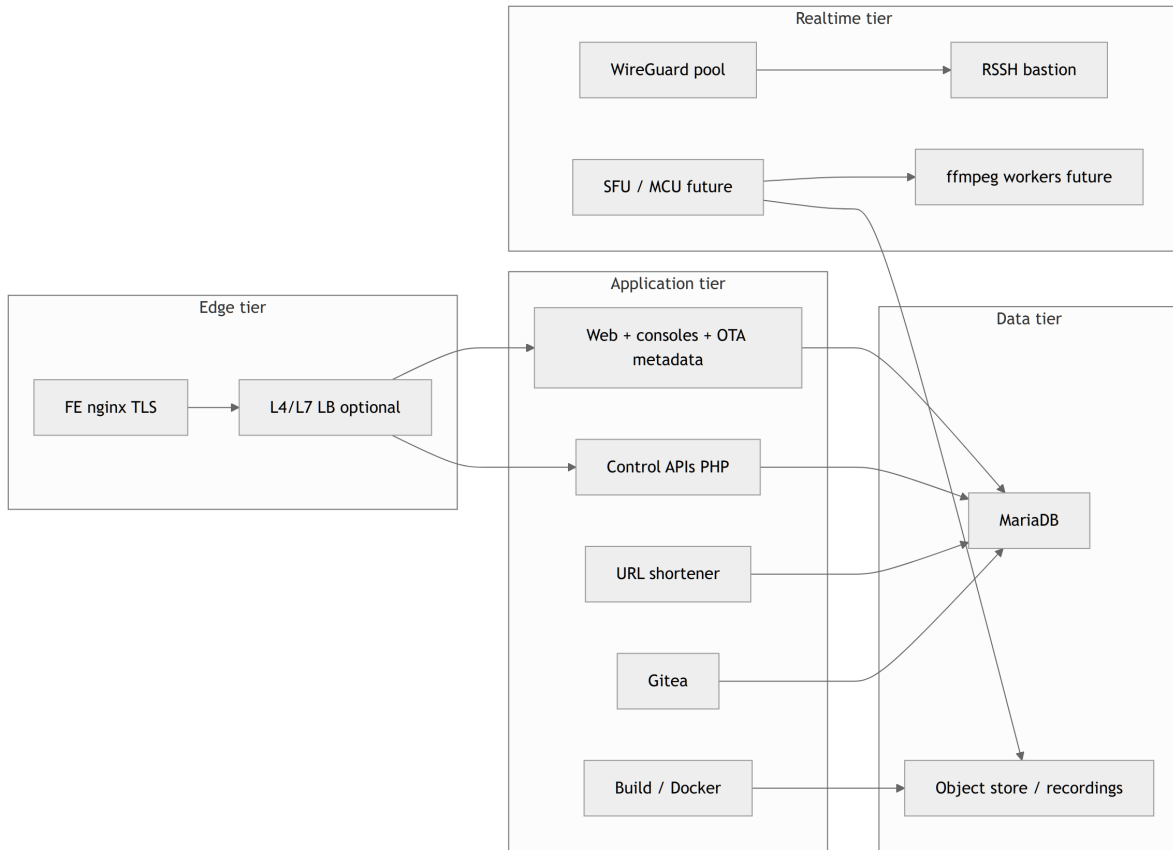
6.1 Environment tiers



Tier	Audience	Data	Deploy freedom
dev	Developers, agents	Synthetic / disposable	Daily; may break
staging	PO, QA, friendly beta	Anonymized prod subset	Weekly; feature flags
prod	End users	Real	Hotfix + scheduled; zero-downtime target

6.2 Service domains

Logical grouping for VM assignment (see DR §Architecture options):



6.3 Reference cluster layout (prod)

Phase A (minimal split) — cost-conscious; matches draft “tighter” intent:

VM role	vCPU	RAM	Disk	Services
be-web-1	2	4 GiB	40 GiB	nginx, PHP-FPM (hub, crashes, graphs, builder UI, OTA JSON)
be-data-1	2	8 GiB	200 GiB+	MariaDB primary; nightly backup to OBJ
be-ops-1	2	4 GiB	80 GiB	Gitea, url-shortener, small object cache
be-build-1	4	16 GiB	100 GiB	Docker builder; ephemeral
be-vpn-1	2	2 GiB	20 GiB	WireGuard + RSSH bastion
fe-1	(existing)	—	—	Gentoo FE — unchanged

Phase B (scale-out) — when metrics justify:

Add	When
be-web-2+	PHP-FPM CPU > 70% sustained or p95 latency SLO miss
be-vpn-2+	RSSH session count > §12.1 per node
be-media-N	SFU/ffmpeg SPEC approved
CDN edge	OTA/release origin egress > §12.3 budget
Regional PoP	p95 RTT > 300 ms for target markets + media revenue (F1/F3)
Managed SQL	DBA ops > 4 h/month or replica lag requirements

6.4 Traffic and control planes

Plane	Protocols	Termination	Notes
Control	HTTPS :443	FE → BE :80	Heartbeats, upload, consoles
Git	HTTPS / SSH	FE path or dedicated	Gitea behind same FE prefix
OTA	HTTPS static	FE → BE or CDN	Large artifacts → object store
VPN lab	UDP :45340	Router DNAT → FE → BE	Unchanged until WG pool
RSSH	TCP :443	ra.apps.f0xx.org	Bastion may move to dedicated VM
Cast LAN	UDP/TCP :41234–41235	Device-local	No BE in alpha
Cast WAN	UDP/TURN/WebRTC	SFU tier (future)	See 20260608_BE_SERVICES_and_infra.md §SFU

7. Service inventory and scaling profile

Service	Stateful?	Scale axis	Co-location rule
Hub / crashes / tickets	No (PHP)	Horizontal	With API pool
Graphs ingest + UI	No	Horizontal	Same pool
Builder UI	No	Horizontal	Same pool; triggers build VM
Docker build	Ephemeral	Queue workers	Dedicated CPU VM
MariaDB	Yes	Vertical → replica	Dedicated ; no ffmpeg
Gitea	Yes (git)	Vertical; mirror RO	Low traffic; ops VM OK
URL shortener	No	Horizontal	Small; can share ops VM
OTA artifacts	Yes (files)	CDN / object store	Split from PHP
WireGuard	Yes (peers)	Pool by region	UDP-friendly VM
RSSH bastion	Session	Horizontal + sticky	See §12.1
SFU / ffmpeg	Session	Horizontal	Dedicated media subnet

8. Deployment and promotion pipeline

```
feature/* or next ■■■ dev cluster (auto or manual)
    ■
    ▼ QA + PO
  staging cluster
    ■
    ▼ soak + sign-off ([ALPHA.md](../ALPHA.md) pattern extended)
  prod cluster
```

Step	Actor	Gate
Deploy to dev	DEV / CI	Unit tests green
Promote to staging	PO or release delegate	Integration + e2e smoke
Promote to prod	PO	Staging soak; GIT_FLOW.md tag on master for app releases

Config rule: config.php and secrets are **environment-specific**; post-sync scripts (e.g. ensure_url_shortener_prod_config.php) must be idempotent per environment.

9. Zero-downtime and load balancing

Tier	Strategy
PHP-FPM	N≥2 VMs behind FE upstream block or cloud LB; max_fpm_children tuned per VM
MariaDB	Primary + read replica; app read/write split optional later
Gitea	Single primary; maintenance window for upgrades unless Gitea HA adopted
WireGuard	New peers on least-loaded node; existing peers sticky until session end
Static OTA	Blue bucket + CDN swap or versioned path (v0/ota/...)

FE nginx pattern (multi-BE):

```
upstream androidcast_be {
  least_conn;
  server 10.7.16.128:80 max_fails=2 fail_timeout=10s;
  server 10.7.16.129:80 max_fails=2 fail_timeout=10s;
}
```

Cloud providers: prefer **managed LB** (GCP ILB, Azure Front Door/Application Gateway) when FE moves partial traffic to cloud.

10. Data and storage

Data class	Store	Backup
Relational (crashes, tickets, RA, short links)	MariaDB	Daily logical dump + binlog optional
Git mirrors	Gitea disk	gitea_collect_backup.sh; off-VM copy
OTA / APK / build artifacts	Object store or large volume	Version immutability
Cast recordings (future)	Object store + lifecycle policy	Glacier-class after N days
Audit / graphs time series	MariaDB or TSDB later	DR \$open

Rule: Object store VMs optimize for **disk IOPS and egress**, not CPU.

11. Simulation lab (Alpine HVMS)

Sysop MAY simulate a cluster on FE hypervisor:

Property	Value
Base OS	Alpine 3.x (match prod BE)
Pre-install	nginx, php81, php81-fpm, mariadb, git, docker, wireguard-tools
Network	Private 10.7.0.0/8 — no WAN topology change
Count	3–7 HVMS typical for dev cluster

FE XEN
■■■■ hvm-be-web-dev
■■■■ hvm-be-data-dev
■■■■ hvm-be-ops-dev
■■■■ hvm-be-build-dev

Deliverable (future script): orchestration/sim/cluster-up.sh — out of scope until PO prioritizes; SPEC reserves name.

12. Capacity planning framework

12.1 RSSH / remote access connections

Planning numbers for **single bastion VM** (2 vCPU, 2 GiB RAM, Alpine):

Metric	Conservative	Target design
Concurrent SSH sessions	50	100 with MaxSessions tuning
Concurrent heartbeat devices (poll 1–7 min)	2 000	5 000 (mostly idle HTTP)
Peak new sessions / min	10	30 with queue
WireGuard peers (lab track)	200	500 split across 2 WG nodes

Bottleneck order: MariaDB connection pool → sshd MaxStartups → PHP-FPM for heartbeat API → CPU on bastion.
When **staging** exceeds conservative column for 24 h, add **be-vpn-2** before cloud migration.

12.2 Cast and media paths (future)

Mode	Latency	Scale unit	ffmpeg role
LAN P2P	Lowest	N/A (alpha)	None
SFU relay	Low	1 SFU ≈ 50–200 viewers (codec dependent)	Optional simulcast
HLS/m3u8 bridge	Medium–high	1 worker ≈ 10–30 transcodes	Heavy CPU

Detailed SFU sizing deferred to owner SFU SPEC (OPEN_TASKS_GRAPH.md).

12.3 Network bandwidth and egress

Scale planning **MUST** include **throughput**, not only CPU/RAM/connections. Hidden bottleneck today: **single FE WAN uplink → FE→BE backhaul → single BE NIC**.
Link inventory (baseline)

Link	Typical role	Metric
Router WAN (134.17.26.161)	Public ingress/egress	Mbps sustained + peak; GB/month
FE ↔ BE (10.7.0.10 → 10.7.16.128)	Proxied HTTP(S), internal sync	Mbps; must exceed sum of BE-facing services
BE origin	PHP, MariaDB, Gitea, OTA origin	NIC util %; egress GB/month
UDP DNAT	WireGuard lab	Mbps per active peer × peers
Future CDN origin pull	OTA, recordings, HLS	Origin Mbps; CDN egress billed separately

Per-service bandwidth profile (planning)

Service	Direction	Order of magnitude (prod planning)	Scale trigger
Control APIs (upload, heartbeat)	In ■ out	1–20 Mbps peak; low GB/month	p95 latency, not Mbps
Consoles / hub	Out	< 5 Mbps sustained	CDN cache static assets
OTA APK / .otapkg	Out	Burst 50–500 Mbps during release	CDN required before wide rollout
Gitea clone/fetch	Out	10–100 Mbps burst	Mirror geo cache
URL shortener redirect	Out	Low	—
Cast recordings (F3)	Out	100 Mbps–1 Gbps per hot object	Object store + CDN; geo replicas
SFU relay (F1)	In+out symmetric	5–50 Mbps × streams	Dedicated media NIC / region
ffmpeg transcode (F2)	In+out	10–100 Mbps per job	Worker pool + queue depth
Commercial VPN (F4)	UDP bidirectional	1–5 Mbps × active tunnels	WG pool per region

Budget template (PO fills per environment)

Metric	Dev	Staging	Prod (initial)	Alert
WAN egress GB/month	—	—	_TBD_	80% of ISP/cloud quota
Peak WAN Mbps	—	—	_TBD_	70% of measured cap
FE→BE peak Mbps	—	—	_TBD_	sustained > 60% 15 min

OTA release peak	—	soak test	CDN absorbs > 90%	origin > 100 Mbps
Media egress GB/month	—	—	_TBD_ (F3+)	cost budget

DR rule: Do not add **be-web-2** for bandwidth until CDN/OTA offload is evaluated — horizontal PHP nodes do not fix a saturated WAN link.

13. Cloud migration (GCP / Azure / other)

Hybrid recommended for first cloud phase:

Stay on-prem (FE + latency-sensitive)	Cloud candidate
Gentoo FE TLS (or parallel cloud LB)	Object store (GCS/Azure Blob)
WireGuard UDP if DNAT chain kept	Managed MariaDB
LAN cast	ffmpeg worker MIG / VMSS
	SFU pool (GPU optional)

Migration sequence (normative):

1. Object store for OTA + build artifacts
2. Read-only analytics replica or backup target
3. Staging cluster entirely in cloud (dress rehearsal)
4. Prod PHP tier behind hybrid LB
5. Media plane last (highest risk)

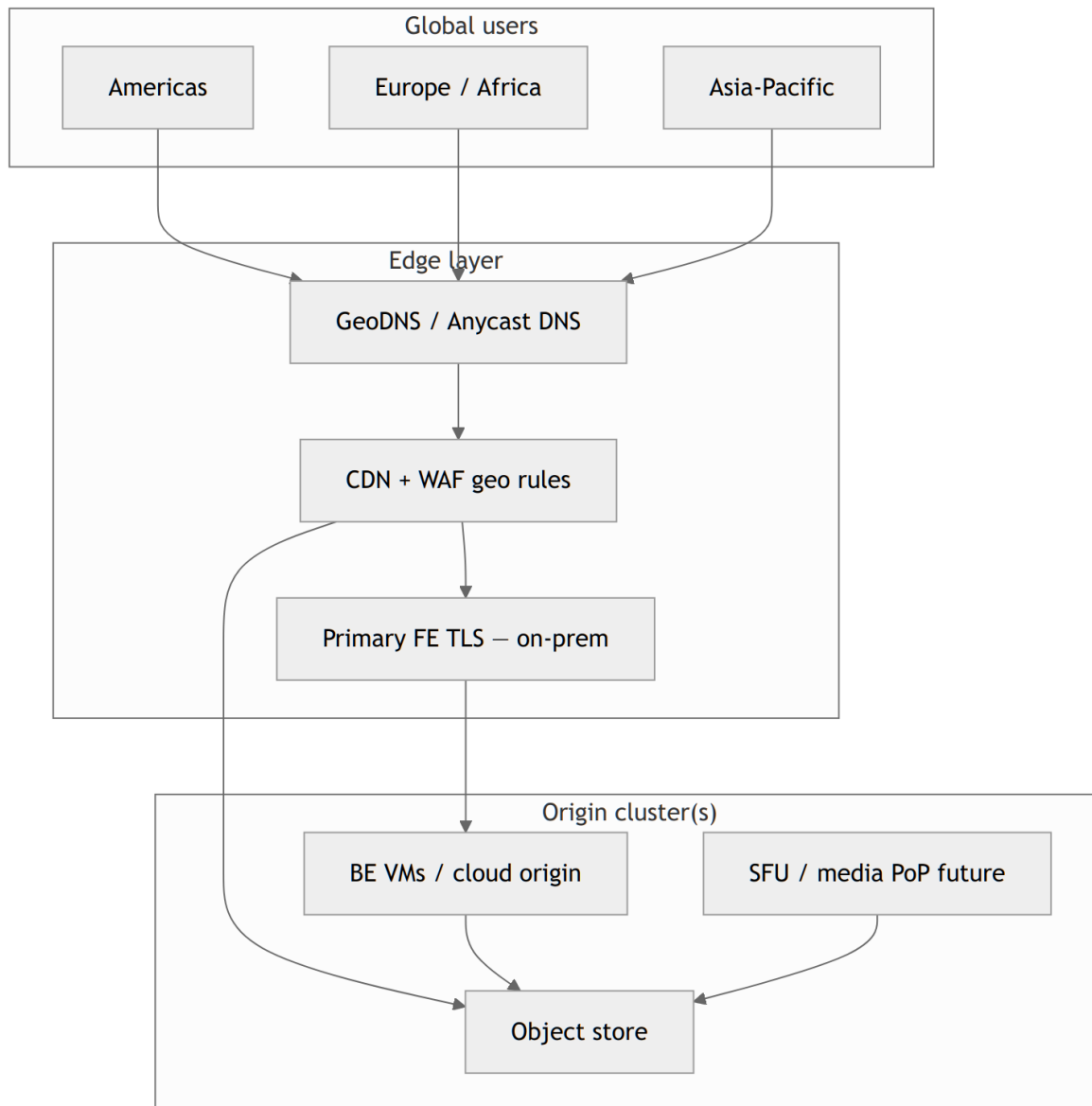
Egress rule: Keep **one** canonical public URL per service; DNS/CNAME moves — no app hard-coded IP changes.

Bandwidth rule: Model **cloud egress pricing** in phase 1 object-store migration; prefer **CDN in front of origin** for any artifact > 1 MB served to > 100 users.

14. Global availability and GEO policy

Draft intent: **anyone in the world can use the platform** (latency permitting), while **specific content or features may be excluded by region** — e.g. a recording or OTA channel must **not** be published or downloadable in **Zambia (zm)** when PO sets that policy.

14.1 Reach model



Phase	Reach	Latency target (control plane)
Now	Single FE/BE (EU-centric)	Best-effort worldwide HTTPS
Phase 5–6	CDN for static + OTA	p95 < 300 ms TTFB via edge
Future	Optional regional media PoP + signaling	RT cast < 150 ms LAN-equivalent where PoP exists

Normative: Public APIs and consoles remain **globally reachable** unless PO declares a **maintenance** or **sanctions** block at edge (separate from per-content deny).

14.2 Segment isolation

Concept	Definition
Segment	ISO country code, optional region group (EU, AFRICA, custom list)
Default	Allow serve worldwide for control plane; content inherits tenant default
deny_publish	Content must not be made available in listed segments (upload/mirror/index)
deny_serve	Existing object must not be delivered (HTTP 451 / empty manifest / VPN reject)
allow_only	Exclusive list — only these segments (stricter catalogues)

Example (illustrative):

```

{
  "resource": "cast_recording:abc123",
  "geo_policy": {
    "default": "allow",
    "deny_serve": ["ZM"],
    "deny_publish": ["ZM"]
  }
}

```

Same pattern applies to OTA channels, short-link targets (optional), and paid stream entitlements (F5).

Content class	Typical policy
Crash/ticket consoles	Global (operator VPN/RBAC unchanged)
OTA stable channel	Global unless PO restricts beta regions
Cast recording (F3)	Per-object deny segments
HLS rebroadcast (F9)	Per-channel geo + copyright rules
Commercial VPN exit (F4)	egress region choice, not arbitrary bypass of deny_serve

14.3 Enforcement layers

Layer	Mechanism	When
L1 DNS / GeoDNS	Route to nearest healthy edge	Multi-PoP phase
L2 CDN / WAF	Geo block or cache vary-by-country	OTA, static, recording download
L3 FE nginx	geo module or auth subrequest	Optional coarse blocks
L4 BE API	Resolve client country (CDN header CF-IPCountry, GeoIP) → geo_policy check	Normative for dynamic content
L5 Object store	Bucket policy / signed URL with segment claim	Large artifacts
L6 App / entitlements	Mobile checks BE before playback	F3, F5, F9

Headers (normative when behind CDN): trust X-Geo-Country or provider equivalent only from **FE/CDN** — strip at edge from client.

Audit: deny_serve events logged to audit table (who, resource, country, time) for operator review.

Dev/staging: may use X-Debug-Country: ZM header to simulate segments without VPN.

15. RBAC and operator model evolution

Phase	Model
Now	Developer-centric; shared ac_crash_sess; privilege sets for tickets, RA, short links
Staging scale	Separate operator vs developer orgs in Gitea; audit log export
Prod commercial	Tenant-scoped RBAC; customer admin roles; SSO optional

Platform **MUST** support **per-environment** user tables or row-level environment_id before multi-tenant commercial features (F5–F9).

16. Commercial and paid services (platform hooks)

No implementation in this SPEC — only **interfaces**:

Hook	Requirement
Billing	Webhook idempotency; tenant_id on all paid resources
Metering	VPN minutes, storage GB, transcode minutes, egress GB — export to metrics
Entitlements	Mobile app checks BE /api/v1/entitlements (future); includes geo_allowed flag
Add-ons	Feature flags per user; GPU queue for masks/background

Payment provider **SHOULD** have **no monthly minimum** at low volume (draft: “near 0 for developer”).

17. Observability and SLO

Service	Health	SLO (prod, initial)
Control APIs	/api/... + upload smoke	99.5% monthly
Consoles	HTTP 200 login page	99.5%
URL shortener	/api/v1/health	99.9%
RSSH heartbeat	device poll success rate	99%
Gitea	mirror sync age < 2× interval	Best effort

Bandwidth metrics (add to graphs / future central metrics):

Metric	Source	Alert
wan_egress_mbps	router or FE interface	> 70% cap 15 min
fe_be_backhaul_mbps	FE→BE link	> 60% cap
origin_egress_gbytes_month	nginx / object store	> 80% budget
cdn_cache_hit_ratio	CDN analytics	< 85% during OTA
geo_deny_serve_total	BE audit	anomaly spike
sfu_media_mbps	SFU (future)	per-PoP capacity plan

Use existing graphs console + future centralized metrics (GRAFANA_vs_others_graphvis_pivot.md).

18. Safety constraints (current topology)

From bottomline_reminder.txt and INFRA.md:

Rule	Detail
No risky WAN changes	Router 134.17.26.161 → FE 10.7.0.10 chain stays until DR-approved migration
Port 8089	Janus/other — not androidcast vhost
BE listen 80	Canonical for FE proxy_pass
ssh access	ssh alpine-be, ssh f0xx-monstro, ssh f0xx.org — inspect tmp/FE_gentoo, tmp/BE_alpine before infra edits
Simulation only	New HVMS on intra 10.7.0.0/8; no public IP experiments without PO

19. Phased delivery map

Phase	Scope	Alpha relation
0	Document + simulation design (this SPEC/DR)	Parallel to alpha
1	Split build Docker to dedicated VM; OTA on object disk	Post-alpha
2	be-web-2 + FE upstream; staging cluster on HVMS	Post-alpha
3	Managed backup; MariaDB replica	Post-alpha
4	SFU + ffmpeg workers (separate SPEC)	Post-alpha
5	Hybrid cloud staging + CDN for OTA/static	Strategic
5b	Bandwidth dashboards + egress budgets on graphs	Strategic
6	Commercial hooks F5–F9	Product-dependent
7	GEO policy enforcement (F10) for recordings/OTA/catalog	After F3 or PO mandate

20. Out of scope (this SPEC revision)

- Concrete Terraform/Pulumi modules
- Pricing quotes for GCP/Azure
- Legal/compliance for payments and VPN resale
- Mobile app changes for commercial features
- Replacing MariaDB with Postgres
- Sanctions-law mapping per country — PO/legal owns lists; platform stores **deny segments** only

21. Related docs

Doc	Link
Infra topology	INFRA.md
BE service map	20260608_BE_SERVICES_and_infra.md
Build / Docker	BUILD_DEPLOY.md
Remote access	REMOTE_ACCESS_IMPL.md, 20260602_REVERSE_SSH_proposals_summary.md
Open tasks graph	OPEN_TASKS_GRAPH.md
Alpha priorities	20260608_ALPHA_PRIORITIES.md
URL shortener	specs/20100611_3_url_shortener.md
Design review	DRs/20100612_1_scaling.md

Changelog

Date	Change
2026-06-12	Initial SPEC from draft 20100612_1_scaling.txt
2026-06-12	Rev. 2 — network bandwidth (§12.3), global availability + GEO policy (§14), metrics