

OpenGeoAR

Project Foundation & Strategy

An open, community-built geospatial AR platform: a shared, spatially accurate digital layer over the real world — free to use, crowdsourced by its users, and structurally resistant to corporate capture.

Daniel Callicot · Project plan, v1.0 · Status: foundation brief — canonical source of truth for architecture, priorities, and constraints

About this document. This is the founding planning brief for OpenGeoAR, an open-source project I initiated and direct. It was developed through AI-assisted working sessions and does two jobs: align human contributors on strategy, and serve as the standing brief that constrains AI-assisted development (Section 11). It appears in my portfolio as an example of systems-level planning — strategy, architecture, governance, risk, and measurable phase gates in a single artifact.

Contents

- 1 Summary: the problem and the concept
- 2 Vision and non-negotiable principles
- 3 Research: prior art and what each precedent proves
- 4 The experience model: three roles, one accuracy ladder
- 5 System architecture
- 6 Technology stack
- 7 Roadmap: five phases, each gated on a demo
- 8 Decisions already made
- 9 Risks and mitigations
- 10 Success metrics per phase
- 11 Working with AI: standing guardrails

1 Summary: the problem and the concept

Augmented reality needs a shared spatial model of the real world — a persistent map that devices localize against and applications anchor content to. That layer is currently being built as proprietary infrastructure (Google's Geospatial API, Niantic's Lightship VPS). Whoever owns the map owns every experience built on top of it.

OpenGeoAR is the counter-model: the *OpenStreetMap of the AR cloud*. A low-poly, 1:1 digital twin of the world that anyone can build AR apps on. All data is open-licensed (ODbL), all code is open source, bulk data is mirrorable by anyone, and the heavy compute runs on contributors' own hardware.

Three strategic moves make this viable without funding, staff, or a compute farm:

1. **Useful on day one.** Launch with a global base model extruded from OpenStreetMap building footprints. Photogrammetry refines the model where contributors care; it never gates availability. This dissolves the cold-start problem.
2. **Users are the factory.** Reconstruction and feature-mapping run on contributors' own GPUs, sharded naturally by geography — people process their own neighborhoods. The project never needs central compute.
3. **Forkability is the anti-capture guarantee.** Bundles are immutable and content-addressed; the only central component is a tiny signed index the whole community can clone. If the steward ever turns hostile, the community forks the index and loses nothing.

Cost envelope (target): phases 0–2 run on one small VPS (~\$5–20/month) plus a consumer GPU the maintainer already owns.

2 Vision and non-negotiable principles

Vision. A persistent, shared, spatially accurate AR layer over the real world that anyone can build on. Users are placed in the model by GPS plus on-device motion tracking, later refined to centimeter accuracy by an open visual positioning system (VPS) — localization by matching what the camera sees against a community-built feature map.

Principles, in priority order. When principles conflict, the lower number wins.

1. **Open by construction.** ODbL data, Apache-2.0 code. No component may depend on a proprietary service for core function.
2. **No single point of control.** Bulk data is content-addressed and mirrorable by anyone; the one centralized element is a small, signed, append-only index the community can fork wholesale.
3. **Useful on day one, everywhere.** A global model derived from OpenStreetMap ships first. Usefulness is never gated on photogrammetry coverage.
4. **Users are the factory.** All heavy compute runs on contributors' hardware, sharded by geography.

5. **Privacy by default.** Low-poly abstraction rather than photorealism; automated face and license-plate scrubbing before anything derived from imagery is published; interiors only with explicit owner consent; a takedown process from day one.
1. **Minimal up-front cost.** The budget constraint is a design input, not an afterthought — it forces the architecture above.

Explicit non-goals for v1

Photorealistic rendering; indoor mapping at scale; anything blockchain or token-based; native mobile apps before the browser-based client is solid; inventing new file formats, coordinate systems, or reconstruction algorithms. The project's innovation is integration and governance, not algorithms.

3 Research: prior art and what each precedent proves

Every major design decision traces to an existence proof. Nothing here is speculative.

Precedent	What it proves	What OpenGeoAR takes
Photosynth / SeaDragon (Microsoft)	Crowd photos can be spatially registered into a navigable 3D model, and streaming only the detail the viewport needs makes huge models usable.	Multi-resolution streaming (via OGC 3D Tiles) and crowd-sourced reconstruction (via open SfM tools run by contributors).
OpenStreetMap	Community geodata at global scale withstands corporate pressure. Contributors map their own neighborhoods.	The governance model, the ODbL license, contributor motivation, and geographic sharding — copied directly.
Google Geospatial API / Niantic Lightship	Camera-based localization against a crowdsourced feature database reaches centimeter-level AR alignment.	Proof the goal is achievable. Their moat is data, not algorithms — the algorithms are published. OpenGeoAR rebuilds the data layer as a commons.
Mapillary	Crowdsourced street imagery works at scale.	The reference for imagery licensing and automated privacy scrubbing.

4 The experience model: three roles, one accuracy ladder

The platform serves three user roles, each with a distinct value loop:

- **App users** open a web link — no install — and see AR content correctly placed and occluded by real buildings, anywhere on Earth.
- **Contributors** walk their neighborhood with a phone, run one Docker command on their own PC overnight, and publish a signed, reviewed tile that upgrades their area for everyone. The motivation loop is OpenStreetMap's: visible, local, owned.
- **App developers** build against an SDK that gives them tiles, a live pose stream, and shared anchors — without asking anyone's permission.

Positioning is designed as a graceful ladder rather than a binary. Each rung works everywhere the previous one does; better data only ever improves the experience:

Rung	Mechanism	Experience
Coarse	GPS + compass place the user in the global model.	Content appears in roughly the right place (a few meters). Available everywhere, day one.
Smooth	The device's own visual-inertial tracking (WebXR) takes over frame-to-frame motion.	Content stays glued to the world as the user moves. No jitter from GPS noise.
Precise	Open VPS: camera frames matched against a community-built feature map.	Centimeter alignment — a virtual object fixed to a real doorway, persistent across sessions, devices, and users. Grows outward from where contributors care.

Accuracy claims always state their mechanism. GPS is a prior, not the localizer; conflating the layers is how AR demos overpromise.

5 System architecture

Five layers, decoupled so each can be mirrored, replaced, or forked independently:

Apps	Third-party AR experiences built on the SDK.
Client runtime (WebXR)	Tile streaming, rendering, positioning fusion (GPS → device tracking → VPS), shared anchors. Runs in the browser — zero install friction.
Index — federated, tiny	A signed, append-only log mapping tile address + version to content hashes and trust metadata. Megabytes at continental scale; anyone can clone all of it.
Bulk data — content-addressed	Immutable tile bundles distributed over HTTP mirrors, IPFS, and torrents. New data is a new version, never an overwrite.
Contributor pipeline	Dockerized: video in, signed bundle out. Runs on contributors' own GPUs.

The tile bundle: the core data contract

The bundle is the unit of everything — contribution, review, distribution, rendering. It is specified first, before any code, because every other component is written against it:

bundle/ manifest.json	schema version, tile address, contributor public key, signature, source, license, parent hashes
coarse.glb	REQUIRED — low-poly mesh (OSM extrusion or decimated recon)
refined.glb	OPTIONAL — higher-detail low-poly mesh from photogrammetry
vps/ provenance.json	OPTIONAL — localization feature map + registered camera poses capture dates, pipeline versions, QA and privacy-scrub flags

Bundles are immutable, signed with the contributor's key, and content-addressed (SHA-256 / IPFS). Immutability is what makes mirroring, caching, offline verification, and forking free.

Tiling, formats, and coordinates

- **Tiles are geohash-addressed** (~150 m cells), with level-of-detail subdivision inside OGC 3D Tiles — an existing standard with existing loaders, chosen precisely so no format needs inventing.
- **Meshes are glTF** with Draco compression; each tile keeps vertices in a local metric frame to avoid floating-point precision loss at planetary coordinates.

The honest position on decentralization

Pure no-authority P2P fails at exactly one layer: moderation. Spam, vandalism, and privacy takedowns are human-governance problems, and pretending otherwise is decentralization theater. The architecture is therefore decentralized bulk data plus a federated micro-index. Moderation is simply which index entries are marked canonical — it starts as maintainer-decides, moves to community review on the OpenStreetMap changeset model, and lands in nonprofit stewardship. Because anyone can clone the index and all data is content-addressed, the credible threat of a fork disciplines whoever holds it. That is the strongest anti-capture guarantee practically achievable.

6 Technology stack

Boring, proven, and entirely open source. Every row is a deliberate refusal to build something novel:

Concern	Choice	Why
Client runtime	WebXR + three.js + 3D Tiles renderer	Zero install friction; a link is the app.
Global base model	OpenStreetMap footprints + height tags, extruded to meshes	Planet-scale coverage before any contributor exists.
Reconstruction	COLMAP / OpenSfM, GPS + OSM geometry as priors	Standard open structure-from-motion; runs on one consumer GPU.
Meshing	Open3D — Poisson surface + aggressive decimation	Low-poly is the goal (bandwidth and privacy), achieved by simplifying real geometry.
Localization (VPS)	hloc: SuperPoint + LightGlue-class matching	Published, state-of-practice algorithms; the value is the open dataset.
Privacy scrubbing	Face / plate detectors inside the pipeline	Runs before anything imagery-derived is published — not optional.
Distribution	HTTPS mirror + IPFS + torrent seeds	No bundle lives in only one place.
Index	Git repository of signed JSON records	Append-only, clonable, reviewable — governance tooling for free.

Concern	Choice	Why
Signing	Ed25519 contributor keys	A key is a contributor identity; every artifact is verifiable offline.

7 Roadmap: five phases, each gated on a demo

Each phase must ship something demoable before the next begins. Costs are targets.

Phase 0 — Foundations weeks · \$0 · done when the contracts exist

- Repository scaffolding and licenses; the bundle and index schemas as versioned, validated contracts; signing and verification tooling; coordinate-math library with round-trip tests.
- Contribution guide, code of conduct, and privacy / takedown policy drafts — governance is a phase-0 deliverable, not a retrofit.

Phase 1 — Dumb but global 1–3 months · ~\$10/mo

- OpenStreetMap-to-tiles pipeline, batch-runnable for any bounding box; browser client that places the user by GPS, tracks with device motion, and renders coarse tiles with correct occlusion of virtual content.
- **Flagship demo: persistent geolocated notes** (“spatial graffiti”) — deliberately trivial content that exercises the hard things: shared anchors, persistence, and moderation.
- Honest accuracy target: building-scale occlusion, placement within a few meters. The wow moment is phase 2; phase 1 sets expectations rather than overpromising.

Phase 2 — Hyper-local proof of quality 3–6 months · one neighborhood

- Documented phone-video capture protocol; reconstruction and VPS pipelines running overnight on one consumer GPU; client relocalization to centimeter accuracy inside the covered zone.
- **Killer demo: a virtual object glued to a real doorway, persistent across sessions, devices, and users.** This demo is the project's recruiting and fundraising asset — it is prioritized as such.

Phase 3 — Users as the factory 6–12 months

- The whole pipeline as one Docker command; contributor onboarding (keys, capture guide, submission); community review on the changeset model; mapping-party seed events for the first towns; mirroring so no data lives in one place.

Phase 4 — Platform 12 months +

- App-developer SDK; guided capture app; nonprofit stewardship of the index with elected moderation; change detection and tile freshness scoring.

8 Decisions already made

Recorded so they are not casually relitigated — each states its mechanism, and changing one requires updating this document:

1. **Existing standards over custom formats.** 3D Tiles and glTF have loaders, tooling, and level-of-detail semantics already; a custom format would spend the project's scarcest resource on its least differentiating problem.
2. **Browser-first over native.** At this stage, distribution friction costs more than polish. A link outcompetes an install.
3. **OSM extrusion before photogrammetry.** Global usefulness on day one; refinement follows attention.
4. **Low-poly via decimation, not edge detection.** 2D edges lack depth. Low-poly remains a goal — for bandwidth and privacy — reached by aggressively simplifying real reconstructed geometry.
5. **GPS is a prior, not the localizer.** True 1:1 alignment comes only from the VPS layer, and the docs say so.
6. **Federated micro-index over pure P2P.** Moderation is human governance; forkability, not theater, is the anti-capture mechanism.
7. **Immutable, signed, content-addressed bundles.** Mirroring, caching, forking, and offline verification fall out for free.
8. **Geographic sharding of compute.** Contributors process their own neighborhoods; the workload is embarrassingly parallel by tile, so no distributed-compute framework is needed.

9 Risks and mitigations

Risk	Mitigation
Cold start — no contributors	The OSM-derived model is useful before any contribution exists; the notes app gives an immediate social payoff for placing content.
Coverage follows a power law — landmarks first	Accepted, not fought: precise zones grow outward from where people care, and the coarse model covers everywhere else.
Privacy backlash	Low-poly abstraction, mandatory scrubbing, no raw imagery published, interiors out of scope, takedown policy in place before any data ships.
GPS jitter undermines early demos	Expectations are set structurally: phase 1 promises building-scale accuracy only; the precision story is explicitly phase 2.
iOS browser limitations	Android-first, with the iOS path (viewer app / native wrapper) documented as a known gap rather than discovered late.
Maintainer burnout / bus factor	Everything forkable by design; documentation treated as a deliverable; co-maintainers recruited off the phase-2 demo.

Risk	Mitigation
Corporate embrace-and-extinguish	Share-alike license on data, early trademark on the name, and foundation stewardship as the end state.

10 Success metrics per phase

Each phase has a falsifiable exit condition. These are targets, set in advance:

Phase	Exit condition
1	The client renders occluding coarse tiles for any world location; a note placed by user A is seen by user B within ~3 meters.
2	Relocalization succeeds >80% of attempts in the covered zone in daylight; anchor drift under 10 cm across sessions; the pipeline processes ~5 city blocks overnight on one consumer GPU.
3	At least 3 external contributors have published accepted tiles from their own hardware; at least 2 independent full mirrors of the index and bundles exist.
4	At least 1 third-party app built on the SDK by someone outside the core team.

11 Working with AI: standing guardrails

The project is built AI-assisted, so the brief includes rules for the AI, written with the same precision as the schemas. These constrain every AI working session:

1. Prefer boring, proven libraries over novel code. This project's innovation is integration and governance, not algorithms.
2. Every pipeline stage must run headless from the command line with a sample fixture, covered by a smoke test in CI.
3. Never introduce a dependency on a paid or proprietary API for core function.
4. Schemas are contracts: version them, never break them silently, write migration notes.
5. When an accuracy claim is made anywhere, state its mechanism — GPS prior, device tracking, or VPS. Never conflate positioning layers.
6. Keep the client small; assume mid-range Android hardware.
7. When a recorded decision changes, update the decision record — the document stays the source of truth.