AE BRAWL TECHNICAL MANUAL

Architecture, protocol, privacy, safety, resilience, verification, and the path from competition proof to responsible production.

10 SECTIONS / ONE ACCOUNTABLE SYSTEM

COMPETITION EDITION / JULY 2026

ATOM EONS

01 1. PRODUCT THESIS

Local combat-sport capacity expires nightly. Search products flatten that capacity into addresses, ratings, and maps. AE BRAWL instead treats a supervised trial session as a time-bound experience: intent, signal, venue, consent, DoorPass, referral settlement, and single redemption.

The visitor creates no permanent account. The venue receives no hidden medical claims or guarantee of readiness. The coach remains the final authority over participation and contact.

02 2. SYSTEM BOUNDARY

The browser owns presentation, ephemeral location, camera-local scouting, consent, and session-only pass display. Google Places can supply likely legitimate venue candidates when a restricted key is present. Keyless judge mode uses visibly fictional venues. LeadMCP owns referral state, simulated settlement, receipt integrity, and replay rejection.

Production integration would add a protected server boundary for verified venue inventory, signed consent exchange, payment-provider settlement, email delivery, abuse prevention, and data-retention controls. Those systems are not falsely represented as live in this build.

03 3. EXPERIENCE ARCHITECTURE

The core loop is:

01. Cold open creates curiosity.
02. Signal tuning converts location search into play.
03. Venue doors expose expiring choices without a map.
04. The dossier makes safety and session details legible.
05. Consent interrupts momentum at the correct risk boundary.
06. DoorPass creates a tangible one-night reward.
07. LeadMCP proof reveals the machine transaction behind the magic.

The reward cadence moves from atmosphere to choice, confidence, artifact, and proof. Judge mode removes permission and credential friction while preserving every conceptual step.

04 4. LEADMCP PROTOCOL

LeadMCP is a zero-dependency Node.js MCP stdio server. It handles initialize, ping, tools/list, tools/call, resources/list, and resources/read as newline-delimited JSON-RPC 2.0.

The seven tools are fight_pass_issue, fight_pass_offer, fight_pass_accept_and_pay, fight_pass_collect, fight_pass_redeem, fight_pass_status, and fight_pass_receipt.

The immutable transition order is ISSUED -> OFFERED -> PAID -> COLLECTED -> REDEEMED. Each mutation validates the current state. Payment is represented by a unique simulated transaction. Consent is represented only by a one-way token hash. The receipt returns a verified SHA-256 event chain.

05 5. PRIVACY ARCHITECTURE

AE BRAWL follows data minimization:

- Location remains in browser memory for the active search.
- Judge mode requires no credential.
- No email, phone, legal name, or biometric record is collected.
- Camera frames are neither uploaded nor persisted.
- DoorPass data lives in sessionStorage and disappears with the browser session.
- LeadMCP stores identifiers, protocol state, amount, expiry, hashes, and venue ID-not raw personal information.

A production referral handoff would require explicit user consent, a published retention schedule, venue agreements, access controls, deletion operations, and jurisdiction-specific review.

06 6. SAFETY ARCHITECTURE

The app never promises a fight or sparring. It routes to supervised combat-sport trial sessions. The safety checkpoint makes three conditions explicit: training is supervised, injuries are disclosed, and contact requires venue approval. The manifesto prohibits unsanctioned fighting and unauthorized filming.

Live Google results are candidates, not guaranteed sessions. Product copy requires confirmation with the venue. Generated artwork shows legitimate gyms, protective context, and coaches; it avoids blood and injury.

07 7. GOOGLE PLACES ADAPTER

The production adapter loads Maps JavaScript API only when VITE_GOOGLE_MAPS_API_KEY exists. It uses nearby gym candidates and filters names and addresses for combat-sport terms. A real deployment must restrict the browser key by HTTP referrer, monitor quota, confirm current Places types, and provide venue verification and opt-out processes.

No map is rendered. This is a deliberate product choice: the interface ranks expiring experiences rather than asking the visitor to interpret pins.

08 8. PWA AND RESILIENCE

The manifest provides standalone installation, theme, icon, and shortcuts. The service worker uses a small app-shell cache, network-first updates, same-origin runtime caching, and a navigation fallback. The deterministic SVG layer makes essential art available even before generated bitmaps load.

Reduced-motion media queries eliminate nonessential animation. More-contrast preferences strengthen boundaries and text. Keyboard focus is visible; touch targets are sized for mobile use.

09 9. VERIFICATION

npm run verify runs five LeadMCP invariants, the complete demo, PDF generation, and the Vite production build. Release verification additionally requires secret scanning, desktop/mobile browser traversal, public HTTP checks, and inspection of published GitHub Actions.

Thirteen tests cover tool count, complete lifecycle, real child-process stdio exchange, request-ID replay, payment and redemption replay, expiration, invalid transitions without mutation, absence of raw consent/PII, intact-chain verification, and tamper detection. They do not claim production payment, venue inventory, email delivery, or legal clearance.

10 10. COMMERCIAL MODEL AND ROADMAP

The initial model charges legitimate venues for verified one-night referrals, not visitors for a recurring membership. The user sees the one-night session price before consent. Future work includes venue onboarding, signed availability, refundable settlement rules, confirmed attendance, fraud controls, provider-neutral machine payments, and service categories beyond combat sports.

The durable innovation is not a fighting directory. It is the combination of perishable local capacity, zero-account intent, consent-scoped referral, machine settlement, and single-use redemption expressed as an MCP protocol.