# **Lean pre-coding sheet — 79/89**

**Component:** crates/vm\_app/Cargo.toml (App meta-manifest)  
 **Version/FormulaID:** VM-ENGINE v0

## **1) Goal & success**

**Goal:** Define a minimal **Cargo package** for the desktop app wrapper that owns packaging metadata and *does not* add runtime/network deps. The actual Tauri backend lives in src-tauri/.

**Success:** cargo metadata -p vm\_app works; cargo build -p vm\_app/src-tauri --locked produces the app backend when explicitly targeted; offline requirements upheld (no telemetry; assets bundled).

## **2) Scope**

**In scope:** Package name/version/license; publish = false; pointing to src-tauri crate (documented relationship); optional feature flags mirrored to the backend (HTML reporting, frontier map); deterministic build profile hints.

**Out of scope:** Backend Rust code (in src-tauri/), UI build (npm/vite), map assets; those are separate files (80–89).

## **3) Inputs → outputs**

**Inputs:** Workspace toolchain pin; backend crate src-tauri; UI bundle under ui/ (consumed by Tauri at runtime, not by this manifest).

**Outputs:** None directly (meta-manifest). Building the app targets **vm\_app/src-tauri** which emits an offline desktop binary per Doc 3.

## **4) Entities/Tables (minimal)**

## **5) Variables (only ones used here)**

## **6) Functions (signatures only)**

N/A (manifest).

## **7) Algorithm outline (what this file enforces)**

Declare package as **private** (publish=false).

Define **passthrough features**: report-html, frontier (forwarded to src-tauri and downstream).

Document that the build target is vm\_app/src-tauri to avoid accidental workspace default builds (Tauri toolchain not required unless explicitly built).

## **8) State flow (very short)**

Acts as a **container package**; actual app build runs in 80/89 with Tauri, which packages the offline UI, fonts, and **MapLibre** tiles/styles locally.

## **9) Determinism & numeric rules**

Follows workspace rules: **no network at runtime**, canonical JSON in artifacts, stable ordering; the app layer must not introduce telemetry or online fonts.

## **10) Edge cases & failure policy**

If built without Node/vite present, this meta package should still parse; only **src-tauri** requires those assets at run/pack time.

Feature combo errors are deferred to backend crates (e.g., frontier without adjacency data skips map step rather than failing).

## **11) Test checklist (must pass)**

cargo metadata -p vm\_app OK.

cargo build -p vm\_app/src-tauri --locked succeeds on supported OS/arch (when explicitly targeted).

Enabling --features report-html,frontier links downstream features and still respects offline constraints.