# **Pre-Coding Essentials (Component: vm\_cli/Cargo.toml, Version/FormulaID: VM-ENGINE v0)**

## **1) Goal & Success**

Problem this component solves (1–2 lines): Define the **CLI binary crate** (vm) and its deterministic, offline build surface. It wires the CLI to pipeline/report crates without introducing networked deps or nondeterminism.

Success criteria: cargo build -p vm\_cli --locked compiles on Win/macOS/Linux; no runtime network; outputs match pipeline stages per Docs 4–5.

## **2) Scope**

In scope: crate metadata, [[bin]] name/path, dependency pins, optional features (pass-through to report/frontier), profiles for deterministic build.

Out of scope: argument parsing logic (src/args.rs) and main flow (src/main.rs)—those are 67/68.

## **3) Inputs → Outputs (with schemas/IDs)**

Inputs: workspace toolchain & lockfile; deps: vm\_pipeline, vm\_io, vm\_report, clap (derive); optional: serde\_json for --print-json.

Outputs: binary **vm** that orchestrates: **LOAD → VALIDATE → TABULATE → ALLOCATE → AGGREGATE → APPLY\_DECISION\_RULES → MAP\_FRONTIER → RESOLVE\_TIES → LABEL\_DECISIVENESS → BUILD\_RESULT → BUILD\_RUN\_RECORD**.

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

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

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

(manifest has no functions)

## **7) Algorithm Outline (bullet steps)**

Declare [[bin]] name="vm" path src/main.rs.

Set [dependencies] on vm\_pipeline, vm\_io, vm\_report, clap with derive.

Expose pass-through [features] (frontier, report-html) mapping to downstream crates.

Profiles: release with lto, codegen-units=1, panic="abort".

Ensure no net-using build scripts; rely on local data only.

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

The CLI binary, when invoked, will sequentially drive the pipeline stages in the fixed order; manifest just enables this wiring.

## **9) Determinism & Numeric Rules**

Ordering/rounding/RNG rules live in core crates; CLI must not inject nondeterminism (no time-based flags, no OS RNG).

Canonical JSON, sorted keys, LF, UTC timestamps are preserved by downstream crates; CLI shouldn’t alter bytes prior to hashing.

## **10) Edge Cases & Failure Policy**

Feature combos: frontier requires adjacency data; if absent, downstream will skip map step without invalidating run.

If gates fail, CLI still packages Result/RunRecord on the “Invalid path”; the manifest must not hide this path behind feature flags.

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

cargo build -p vm\_cli --locked (dev & release).

cargo run -p vm\_cli -- --help exits 0; no network I/O at runtime.

With --features frontier, binary links and runs; pipeline produces FrontierMap only when data present.

End-to-end smoke: invoking CLI over Annex B Part 0 fixtures yields Result + RunRecord with correct stage order.