# **Pre-Coding Essentials (Component: vm\_cli/src/main.rs, Formula/Engine v0)**

## **1) Goal & Success**

**Goal:** Orchestrate the fixed pipeline and emit canonical artifacts (**Result**, **RunRecord**, optional **FrontierMap**) and reports, **offline** and **deterministic**.

**Success:** Same inputs (and same seed when used) ⇒ **byte-identical** outputs across OS/arch; ordering/rounding/RNG rules respected; reporting uses one-decimal presentation only.

## **2) Scope**

**In scope:** parse args; file I/O; pipeline step orchestration with **stop/continue** semantics; canonical serialization/hashing handoff; report rendering; exit code selection based on run label/outcomes.

**Out of scope:** core math/algorithms (tabulate/allocate/gates/frontier), schemas, UI rendering beyond Doc 7 rules.

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

**Inputs (all local):** paths to **DivisionRegistry**, **BallotTally**, **ParameterSet**, optional **Adjacency**; output dir; render flags; optional **rng\_seed**. **No network.**

**Outputs (canonical JSON; UTF-8; LF; sorted keys; UTC):** **result.json** RES:<hash>, **run\_record.json** RUN:<ts>-<hash>, optional **frontier\_map.json** FR:<hash>. Reports JSON/HTML per **Doc 7** (one-decimal display only).

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

(IDs & hashing rules per **Annex A** canonical serialization/IDs.)

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

*(All other VM-VARs come from the provided* ***ParameterSet*** *and are not interpreted by main.rs directly.)*

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

fn main() -> ExitCode — parse args, call run(); map errors to exit codes.

fn run(cfg: CliCfg) -> Result<RunSummary, CliError> — orchestrate pipeline.

fn write\_artifacts(res: &Result, rr: &RunRecord, fm: Option<&FrontierMap>, out: &Path) -> io::Result<()> — canonical JSON write (UTF-8, LF, sorted keys, UTC timestamps).

fn render\_reports(res: &Result, rr: &RunRecord, opts: RenderOpts) -> Result<(), CliError> — JSON/HTML per Doc 7 (one-decimal).

fn exit\_code(label: Label, gates\_failed: Option<&str>) -> ExitCode — choose exit code (see §10).

## **7) Algorithm outline (pipeline orchestration)**

**LOAD** inputs.

**VALIDATE** → on fail, build **Invalid** result + run record (reasons), skip counting.

**TABULATE** → **ALLOCATE** → **AGGREGATE**.

**APPLY\_DECISION\_RULES** → on **Fail**, mark **Invalid**, **skip MAP\_FRONTIER**.

If enabled, **MAP\_FRONTIER** (never invalidates; may cause **Marginal**).

**RESOLVE\_TIES** per policy; if random, use **rng\_seed** and log.

**LABEL\_DECISIVENESS** → **BUILD\_RESULT** → **BUILD\_RUN\_RECORD**.

**Render reports** (Doc 7A order; one-decimal).

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

args → run → {LOAD→…→BUILD\_RUN\_RECORD} → write artifacts → render reports → exit.  
 Stop/continue follows Doc 5A exactly.

## **9) Determinism & Numeric rules**

**No network**; inputs local. **Stable ordering** and **round-half-even** internal comparisons; **one-decimal** at report layer only. **RNG** only with explicit seed; seed recorded.

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

**Validation failed:** still emit **Invalid** Result/RunRecord and render fallback per Doc 7B; **omit Frontier**.

**Gates failed:** render up to Legitimacy panel; mark **Invalid**; **omit Frontier**.

**Exit codes:**

0 on **Decisive/Marginal** (artifacts & reports written).

2 on **Invalid (validation failed)**.

3 on **Invalid (gate failed)**.

1 on **CLI/config/IO error** before any artifacts.  
 *(Mapping is an implementation policy; labels and reasons come from artifacts per Docs 5/7.)*

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

Repeat runs on same OS and across OSes → **identical Result/RunRecord hashes** (VM-TST-019/020).

Report sections/wording match **Doc 7A/B** (order, one-decimal, mandatory approval-denominator sentence).

Frontier present only when produced; absence does **not** affect hashes.

Artifacts use canonical JSON (UTF-8, LF, sorted keys, UTC) and hash to the IDs they claim.