Pre-Coding Essentials (Component: crates/vm\_report/src/lib.rs, Version/FormulaID: VM-ENGINE v0) — 61/89

1. Goal & Success

Goal: Provide the high-level reporting API that consumes only **Result**, optional **FrontierMap**, and **RunRecord**, then renders reports that follow Doc 7’s fixed sections, precision, and data sources.

Success: Output respects one-decimal percentages and exact section order; includes the approval-denominator sentence for approval ballots; pulls all fields from the mandated artifacts; echoes tie policy/seed when applicable.

1. Scope

In scope: Build a ReportModel view from Result/RunRecord/FrontierMap; enforce Doc 7 section order, snapshot contents, and data mapping; handle invalid/gates-fail fallbacks; optional Sensitivity if CompareScenarios exists; formatting helpers (one-decimal).

Out of scope: I/O, file writing, and template assets (handled by renderers). No external data fetch (offline).

1. Inputs → Outputs (with schemas/IDs)

Inputs: **Result** (RES:…), optional **FrontierMap** (FR:…), **RunRecord** (RUN:…). All content must be derived from these.

Outputs:

* ReportModel struct (sections §1–§10) consumed by renderers; footer fields bound to RunRecord/Result.
* Public fns return serialized JSON/HTML strings; renderers add no data.

1. Entities/Tables (minimal)

(N/A – the report model mirrors Doc 7A sections/fields.)

1. Variables (rendered here; not computed)

Values are displayed from the ParameterSet snapshot embedded via Result/RunRecord. Domains shown for clarity:

* **VM-VAR-001 ballot\_type** ∈ {plurality, approval, score, ranked\_irv, ranked\_condorcet}
* **VM-VAR-010 allocation\_method** ∈ {winner\_take\_all, proportional\_favor\_big, proportional\_favor\_small, largest\_remainder, mixed\_local\_correction}
* **VM-VAR-012 pr\_entry\_threshold\_pct** ∈ % 0..10
* **VM-VAR-020 quorum\_global\_pct** ∈ % 0..100
* **VM-VAR-021 quorum\_per\_unit\_pct** ∈ % 0..100 (+ **VM-VAR-021\_scope** ∈ {frontier\_only, frontier\_and\_family} if set)
* **VM-VAR-022 national\_majority\_pct**, **VM-VAR-023 regional\_majority\_pct** ∈ % **50..75** (default 55)
* **VM-VAR-024 double\_majority\_enabled**, **VM-VAR-025 symmetry\_enabled** ∈ {on, off}
* **VM-VAR-028 roll\_inclusion\_policy** ∈ **{residents\_only, residents\_plus\_displaced, custom:list}**
* **VM-VAR-030 weighting\_method** ∈ {equal\_unit, population\_baseline}
* **VM-VAR-031 aggregate\_level** = **country** (v1 fixed)
* **VM-VAR-040 frontier\_mode** ∈ {none, sliding\_scale, autonomy\_ladder}
* **VM-VAR-042 frontier\_bands** (ordered, non-overlapping; structure only)
* **VM-VAR-047 contiguity\_edge\_types** ⊆ {land, bridge, water}
* **VM-VAR-048 island\_exception\_rule** ∈ {none, ferry\_allowed, corridor\_required}
* **VM-VAR-032 tie\_policy** ∈ {status\_quo, deterministic, random} (echoed in report provenance)
* **VM-VAR-033 tie\_seed** ∈ integer ≥0 (echoed only when tie\_policy = random)

1. Functions (signatures only)

pub struct ReportModel { /\* sections per Doc 7A; all fields sourced from Result/RunRecord/FrontierMap \*/ }

pub fn build\_model(

result: &ResultDb,

run: &RunRecordDb,

frontier: Option<&FrontierMapDb>,

compare: Option<&CompareScenarios>

) -> ReportModel;

pub fn render\_json(model: &ReportModel) -> String; // one-decimal formatting applied

pub fn render\_html(model: &ReportModel) -> String; // templates consume only fields from model

1. Algorithm Outline

**Cover & Snapshot**

* Fill label (“Decisive/Marginal/Invalid”), and snapshot items: **Ballot** (VM-VAR-001), **Allocation** (VM-VAR-010), **Weighting** (VM-VAR-030), **Thresholds** (VM-VAR-020/022/023), **Double-majority** (VM-VAR-024), **Symmetry** (VM-VAR-025), **Frontier mode** (VM-VAR-040). Values come from the ParameterSet snapshot (via Result/RunRecord).

**Eligibility & Rolls**

* Print **roll inclusion policy** from **VM-VAR-028** using fixed labels:
  + residents\_only → “Residents only”
  + residents\_plus\_displaced → “Residents + displaced”
  + custom:list → “Custom (see list)” and render the provided list.
* Show DivisionRegistry provenance {source, published\_date} and totals (Σ eligible\_roll, Σ ballots\_cast).
* If **VM-VAR-021 > 0**, add the per-unit quorum note, mentioning **VM-VAR-021\_scope**.

**Ballot (method paragraph)**

* Plain-English description per ballot type.
* **Mandatory sentence for approval ballots:** “For legitimacy gates, the support % is the **approval rate** = approvals for the Change option divided by **valid ballots**.”

**Legitimacy Panel**

* Show quorum/majority/double-majority/symmetry with Pass/Fail and values pulled from Result.

**Outcome/Label**

* “Decisive / Marginal / Invalid (reason)”.

**Frontier (if present)**

* Render statuses and counters from FrontierMap; note counts for quorum\_blocked, protected\_blocked, mediation, enclave.

**Sensitivity**

* If CompareScenarios exists, render 2×3 table; else “N/A”.

**Integrity / Provenance**

* Echo IDs and engine identifiers from RunRecord.
* If tie\_policy = random, display **tie\_seed (VM-VAR-033)**.

1. State Flow (very short)

Called after pipeline packaging; strictly offline; renderers consume only ReportModel.

1. Determinism & Numeric Rules

One-decimal at presentation; no double rounding; numbers come verbatim from artifacts; stable section order.

1. Edge Cases & Failure Policy

* Validation failed: render Cover/Eligibility/Ballot + “Why invalid”; mark ❌ Invalid; omit Frontier.
* Gates failed: up to the panel with ❌; outcome “Invalid (gate failed: …)”; omit Frontier.
* If an unknown roll\_inclusion\_policy value is encountered, render the raw value verbatim (no crash) and flag in a non-blocking footnote.

1. Test Checklist (must pass)

* Section order and one-decimal formatting.
* Approval-denominator sentence appears for approval ballots.
* Roll policy renders correctly for each allowed value (and gracefully for custom:list).
* Frontier shown only when FrontierMap exists; diagnostics mirror flags.
* Integrity identifiers match RunRecord + Result.id; **tie\_policy** echoed; **tie\_seed** shown only when random.