Pre-Coding Essentials (Component: crates/vm\_pipeline/src/map\_frontier.rs, Version/FormulaID: VM-ENGINE v0) — 55/89

1. Goal & Success

Goal: Translate per-Unit support into **FrontierMap** statuses using the selected **frontier mode**, contiguity policy, island/corridor rule, and scoped per-Unit quorum effects.

Success: Exactly one status per Unit; components computed only from allowed edge types; per-Unit quorum respected; **protected areas never change** (blocked with a flag); output matches FrontierMap fields; deterministic given same inputs.

1. Scope

In scope: **frontier\_mode ∈ {sliding\_scale, autonomy\_ladder}** (where “binary” is just a single cutoff band), component/contiguity computation, island/corridor handling, per-Unit quorum interaction, autonomy package tagging (via bands), summary counters.

Out of scope: Gate evaluation (must have **passed** already), tie resolution, report rendering.

1. Inputs → Outputs (with schemas/IDs)

Inputs

* **LoadedContext** (Units with protected\_area?, Adjacency {a,b,type}, ParameterSet).
* Per-Unit **support % for Change** (approval: **approval rate = approvals\_for\_change / valid\_ballots**).
* Optional map of per-Unit quorum pass/fail (if VM-VAR-021 > 0 with scope).

Variables (used here)

* **VM-VAR-040 frontier\_mode** ∈ {none, sliding\_scale, autonomy\_ladder} (we are called only if ≠ none).
* **VM-VAR-042 frontier\_bands** — ordered, non-overlapping bands; may carry **action/AP ids** for autonomy ladder.
* **VM-VAR-047 contiguity\_edge\_types** ⊆ {land, bridge, water}.
* **VM-VAR-048 island\_exception\_rule** ∈ {none, ferry\_allowed, corridor\_required}.
* **VM-VAR-021 quorum\_per\_unit\_pct** (+ **VM-VAR-021\_scope**) if provided by gates.

Output

* **FrontierMap (FR:…)** with per-Unit fields: {status, band\_id?, component\_id, flags{mediation,enclave,protected\_blocked,quorum\_blocked}} and summary counters.

1. Entities/Tables (minimal)

(Types align with Doc 1/5 sketches: FrontierUnit/FrontierOut; IDs are Unit IDs; band ids come from frontier\_bands.)

1. Variables (used here)

(See list under Inputs; no 041/045/046 variables are used.)

1. Functions (signatures only)

pub fn map\_frontier(

units: &UnitsView,

unit\_support\_pct: &BTreeMap<UnitId, Ratio>, // approval: approval rate; others per Doc 4

adjacency: &AdjacencyView,

p: &Params,

per\_unit\_quorum: Option<&BTreeMap<UnitId, bool>>

) -> FrontierMap;

fn build\_components(adjacency: &AdjacencyView, allowed: &ContiguityModes) -> Components;

fn apply\_island\_exception(components: &Components, rule: IslandRule) -> MediationFlags;

fn status\_by\_band(s: Ratio, bands: &[Band]) -> (Status, BandId); // single-cutoff = one band

1. Algorithm Outline

**Preconditions**

* Caller ensures frontier\_mode != none.
* frontier\_bands are validated earlier: ordered, non-overlapping, cover intended ranges; autonomy ladder bands carry AP ids if needed.

**Components & adjacency**

* Build connected components using only **VM-VAR-047** edge types (stable order).
* Apply **VM-VAR-048**:
  + none → water isolation does **not** connect; such isolated eligible units may become *mediation*.
  + ferry\_allowed → allow water to connect islands to mainland.
  + corridor\_required → bridges alone insufficient; require an explicit “corridor” classification to connect across water.

**Per-Unit quorum interaction**

* If VM-VAR-021 > 0: a **failing unit cannot change**; set quorum\_blocked=true and force status=no\_change. Family inclusion/exclusion was already handled in gates by **VM-VAR-021\_scope**.

**Protected areas**

* If unit.protected\_area == true and the band implies a change, **block change**; set protected\_blocked=true; final status=no\_change.

**Assign status by band**

* Compute (status, band\_id) via **status\_by\_band(support, frontier\_bands)**.
* “Binary” behavior is modeled by providing exactly **one cutoff band** (e.g., ≥60% ⇒ change); otherwise use multiple bands (sliding or ladder).
* For **autonomy\_ladder**, bands include the **AP id** to tag into the FrontierMap.

**Mediation & enclaves**

* If a unit meets a band but is **not** connected (per allowed edges) to a qualifying cluster, mark **mediation** (no change).
* Units fully surrounded by non-change areas after mapping can be flagged as **enclave** (informational).

**Emit FrontierMap**

* One record per Unit: status, band\_id?, component\_id, flags.
* Maintain summary counters by status/flags for reporting.

1. State Flow

APPLY\_DECISION\_RULES (Pass) → **MAP\_FRONTIER** → RESOLVE\_TIES (only if blocking) → LABEL. If gates **Fail**, caller skips frontier.

1. Determinism & Numeric Rules

Integer/rational comparisons; **≥** at cutoffs; no presentation rounding.  
Stable iteration orders (Units by ID). Same inputs ⇒ identical FrontierMap bytes.

1. Edge Cases & Failure Policy

* Missing adjacency when frontier\_mode active ⇒ ReferenceError.
* Bands unordered/overlapping should have been blocked in VALIDATE.
* Exact cutoff (support == threshold) counts as **meeting** the band.
* AP id missing on an autonomy ladder band ⇒ ReferenceError.

1. Test Checklist (must pass)

* **Cutoff band case (“binary”)**: single band ≥60%; allowed={land}; island separated by water ⇒ mainland units change; island units mediation under none, connect under ferry\_allowed.
* **Multi-band**: ordered bands map to exactly one status per Unit; ladder bands tag AP ids; no flags ⇒ Decisive.
* **Protected area**: protected Unit mapped to change by bands remains **no\_change** and sets protected\_blocked=true.
* **Quorum scope**: failing per-Unit quorum forces no\_change and quorum\_blocked=true.