# **Doc 5A — Pipeline: State Machine & Data Exchanges**

**Scope:** The fixed run flow (state machine) and the artifacts passed between stages. Names align with **Doc 1** entities and with logic in **Doc 4**. Determinism constraints align with **Doc 3**.

## **1) Determinism & Naming (binding)**

* **Same inputs + same engine ⇒ identical outputs.**
* **Ordering:** any iteration/reduction uses a **stable total order** on IDs (Unit IDs, Option IDs via Option.order\_index, etc.).
* **Rounding:** internal comparisons use **round half to even**; **presentation rounding** happens only in the Report (Doc 7).
* **Randomness:** only in tie resolution when tie\_policy = random, with an explicit **rng\_seed** (recorded in the **RunRecord**).
* **Offline:** no network calls; all inputs are local (Doc 3).
* **Names:** Artifacts and entities use **exact labels** below; DB entities are those in **Doc 1**.

## **2) State Machine (fixed order & stop/continue semantics)**

1. **LOAD**
2. **VALIDATE**
3. **TABULATE**
4. **ALLOCATE**
5. **AGGREGATE**
6. **APPLY\_DECISION\_RULES**
7. **MAP\_FRONTIER** *(only if enabled)*
8. **RESOLVE\_TIES** *(only if blocking)*
9. **LABEL\_DECISIVENESS**
10. **BUILD\_RESULT**
11. **BUILD\_RUN\_RECORD**

**Stop/continue rules (must implement exactly):**

* If **VALIDATE fails** → mark run **Invalid**; **skip 3–8**; still do **LABEL\_DECISIVENESS** (Invalid), **BUILD\_RESULT**, **BUILD\_RUN\_RECORD** with reasons.
* If **APPLY\_DECISION\_RULES** has any **Fail** (quorum, majority, double-majority, symmetry) → mark run **Invalid**; **skip MAP\_FRONTIER**; continue to **RESOLVE\_TIES** only if a blocking tie must be logged; then label & build outputs.
* **MAP\_FRONTIER** never invalidates a run; contiguity/protection/per-unit-quorum conflicts yield **Mediation/Protected flags** and can change the **label to Marginal** (Doc 4C).
* **RESOLVE\_TIES** is entered only if a decision is blocked (e.g., WTA tie, last-seat tie, IRV elimination tie). If policy is status\_quo or deterministic\_order, no RNG is used; if random, the **rng\_seed** must be applied and logged.

## **3) Canonical Data Exchanges (artifacts)**

These are implementation-neutral contracts between stages. Where a name equals a DB entity in **Doc 1**, it is noted. Field sketches indicate intent; full per-field lists live in Doc 1B / 5B / 5C.

### **3.1 LoadedContext (ephemeral)**

**Produced by:** LOAD  
 **Contains:**

* Chosen **DivisionRegistry** (REG id), **Units**, **Options** (with order\_index), **Adjacency** (if any)
* **BallotTally** label & dataset (TLY id)
* **ParameterSet** (PS id; full VM-VAR map)
* Engine identifiers for determinism (FormulaID, EngineVersion) for later echo  
   **Notes:** Immutable snapshot for the run.

### **3.2 UnitScores (ephemeral, per Unit)**

**Produced by:** TABULATE  
 **Contains (per Unit):**

* scores{ Option → natural tally } (plurality=counts; approval=approvals; score=score\_sum; ranked → method tallies)
* turnout{ ballots\_cast, invalid\_or\_blank, valid\_ballots }
* Audit hooks: **RoundLog** (IRV) or **PairwiseMatrix** (Condorcet) to be emitted later  
   **Used by:** ALLOCATE, MAP\_FRONTIER, AGGREGATE

### **3.3 UnitAllocation (ephemeral, per Unit)**

**Produced by:** ALLOCATE  
 **Contains (per Unit):**

* seats\_or\_power{ Option → int seats or % power } (sums to Unit.magnitude or 100%)
* Tie notes if last seat required policy application  
   **Used by:** AGGREGATE

### **3.4 AggregateResults (ephemeral, by level)**

**Produced by:** AGGREGATE  
 **Contains (per level: District/Region/Country):**

* Totals/seats per Option (sums over child Units)
* Shares per Option
* Carried turnout denominators required for gates
* **weighting\_method** applied (equal\_unit / population\_baseline)  
   **Used by:** APPLY\_DECISION\_RULES; also referenced in reporting

### **3.5 LegitimacyReport (ephemeral)**

**Produced by:** APPLY\_DECISION\_RULES  
 **Contains:**

* **Quorum:** national turnout vs **VM-VAR-020**; per-unit quorum outcomes (if **VM-VAR-021 > 0**) and **VM-VAR-021\_scope** effect
* **Majority/Supermajority:** national support vs **VM-VAR-022** (denominators per Doc 4A; approval uses **approval rate**)
* **Double-majority:** affected-family definition (**VM-VAR-026/027**) and result vs **VM-VAR-023**; enforced rule when frontier\_mode=none
* **Symmetry:** **VM-VAR-025** result; **symmetry\_exceptions (VM-VAR-029)** if any
* Overall gate **Pass/Fail** flags with reasons and raw computed values  
   **Used by:** LABEL\_DECISIVENESS; informs Result writing

### **3.6 FrontierMap (DB entity, optional)**

**Produced by:** MAP\_FRONTIER  
 **Contains (per Unit):**

* status ∈ { no\_change, phased\_change, immediate\_change, autonomy(AP:...) }
* band\_met (if sliding/ladder)
* Contiguity diagnostics: component id, **mediation/enclave** flags (via **VM-VAR-047/048**)
* protected\_override\_used (when **VM-VAR-045** permits change in protected areas)  
   **Used by:** LABEL\_DECISIVENESS, BUILD\_RESULT, BUILD\_RUN\_RECORD

### **3.7 TieLog (embedded in Result)**

**Produced by:** RESOLVE\_TIES  
 **Contains entries:**

* context (e.g., “WTA winner in Unit U:…”, “last seat in Unit …”, “IRV elimination in Unit …”)
* candidates (Option IDs), policy (status\_quo / deterministic\_order / random), order\_or\_seed, winner  
   **Used by:** BUILD\_RESULT (audit); Report Annex E

### **3.8 DecisivenessLabel (ephemeral)**

**Produced by:** LABEL\_DECISIVENESS  
 **Contains:**

* label ∈ {Decisive, Marginal, Invalid}
* reason (verbatim phrase for report)
* Inputs considered: gate outcomes, **national margin** vs **VM-VAR-062**, and existence of mediation/enclave/protected-override flags  
   **Used by:** BUILD\_RESULT

### **3.9 Result (DB entity)**

**Produced by:** BUILD\_RESULT  
 **Contains:**

* **Top-level:** id (RES:…), references to reg\_id, ballot\_tally\_id, parameter\_set\_id
* **Per-Unit blocks:** tabulation summaries, allocation, per-unit **validity flags**:  
   unit\_data\_ok, unit\_quorum\_met, unit\_pr\_threshold\_met, protected\_override\_used, mediation\_flagged
* **Aggregates by level:** totals & shares, turnout metrics, weighting used
* **Legitimacy gates:** values & Pass/Fail (from **LegitimacyReport**)
* **TieLog** (from 3.7)
* **Label** (from 3.8)
* Optional frontier\_map\_id  
   **Used by:** BUILD\_RUN\_RECORD; Report (Doc 7)

### **3.10 RunRecord (DB entity)**

**Produced by:** BUILD\_RUN\_RECORD  
 **Contains:**

* id (RUN:…), timestamps (UTC)
* Identifiers: **FormulaID**, **EngineVersion**, reg\_id, ballot\_tally\_id, parameter\_set\_id
* Determinism settings: rounding mode, ordering basis, **rng\_seed** (if used)
* Pointers: result\_id, optional frontier\_map\_id
* Environment summary (optional)  
   **Used by:** audit/repro; Report “Integrity & Reproducibility”

## **4) Data Flow at a Glance**

LOAD → LoadedContext

↓

VALIDATE (fail ⇒ Invalid path)

↓

TABULATE → UnitScores

↓

ALLOCATE → UnitAllocation

↓

AGGREGATE → AggregateResults

↓

APPLY\_DECISION\_RULES → LegitimacyReport (Fail ⇒ skip MAP\_FRONTIER)

↓

MAP\_FRONTIER → FrontierMap (optional)

↓

RESOLVE\_TIES → TieLog (only if blocking)

↓

LABEL\_DECISIVENESS → DecisivenessLabel

↓

BUILD\_RESULT → Result

↓

BUILD\_RUN\_RECORD → RunRecord

## **5) Acceptance for this part**

* Stage order and stop/continue semantics match §2 exactly.
* All artifacts above are produced/consumed as specified; names align with **Doc 1**.
* Determinism constraints (ordering, rounding, RNG-seed use) match **Doc 3/4**.
* The content outlines are sufficient for 5B/5C to define function-level contracts and for Doc 7 to map report fields.

**Status:** Flow and artifacts are crystal clear.

# **Doc 5B — Pipeline: Functions 001–006 (contracts)**

**Scope:** Function-level contracts for stages **LOAD → VALIDATE → TABULATE → ALLOCATE → AGGREGATE → APPLY\_DECISION\_RULES**.  
 **Artifacts & names:** must match **Doc 5A** and **Doc 1**.  
 **Determinism:** ordering, rounding, RNG rules per **Doc 3/4**.  
 **Standard errors:** SchemaError, ReferenceError, ConstraintError, MethodConfigError, TieError, ContiguityError, DeterminismError, QuorumError (as recorded status).

## **VM-FUN-001 — LoadInputs**

**Purpose** Create an immutable **LoadedContext** from the selected inputs.

**Inputs**

* IDs: reg\_id (DivisionRegistry), tally\_id (BallotTally), parameter\_set\_id (ParameterSet)
* Local files/data blobs for those IDs

**Preconditions**

* All three IDs exist and are readable.

**Output**

* **LoadedContext** containing: Registry (Units, Adjacency), Options (with order\_index), BallotTally (with ballot\_type), ParameterSet (full VM-VAR map), and engine identifiers (FormulaID, EngineVersion) for echo later.

**Postconditions**

* Snapshot is read-only for the run.

**Errors**

* ReferenceError (missing ID), SchemaError (malformed payloads)

**Audit**

* Echo selected IDs and brief counts (units/options/adjacency rows); record ballot\_type and ParameterSet version.

## **VM-FUN-002 — ValidateInputs**

**Purpose** Perform **structural and semantic** validation before any math.

**Inputs**

* **LoadedContext**

**Preconditions**

* None beyond VM-FUN-001 success.

**Output**

* ValidationReport { pass|fail, issues[] } (issues have severity, code, message, where)

**Postconditions**

* If pass=false, the run must be labeled **Invalid** later and stages 3–8 are skipped (Doc 5A).

**Errors**

* Throw only for unrecoverable loader problems already covered in VM-FUN-001. Prefer to **report** issues in ValidationReport. May raise SchemaError for contradictions that prevent even packaging an Invalid result.

**Checks (must implement exactly)**

**Registry & hierarchy**

* Units form a **tree** (single root, no cycles). (ConstraintError)
* Each Unit has magnitude ≥ 1. (ConstraintError)

**Ballot & tallies**

* BallotTally.ballot\_type == VM-VAR-001. (MethodConfigError)
* **Tally sanity:** per Unit, sum(valid option tallies) + invalid\_or\_blank ≤ ballots\_cast. (ConstraintError)
* Ranked data present if ballot\_type ∈ {ranked\_irv, ranked\_condorcet}. (MethodConfigError)
* Score data consistent with [VM-VAR-002..003]; normalization flag valid. (MethodConfigError)

**WTA constraint**

* If VM-VAR-010 = winner\_take\_all ⇒ every Unit.magnitude = 1. (MethodConfigError)

**Weighting**

* If VM-VAR-030 = population\_baseline ⇒ every aggregated Unit has **positive** population\_baseline and population\_baseline\_year. (ConstraintError)

**Quorum data**

* If VM-VAR-020 > 0 (global quorum) ⇒ every aggregated Unit has eligible\_roll and eligible\_roll ≥ ballots\_cast. (ConstraintError)
* If VM-VAR-021 > 0 (per-unit quorum) ⇒ all Units have eligible\_roll. (ConstraintError)

**Double-majority scoping**

* If VM-VAR-024 = on and frontier\_mode = none ⇒ **require** VM-VAR-026 ∈ {by\_list, by\_tag} and ensure VM-VAR-027 resolves to a **non-empty** family. (MethodConfigError / ReferenceError)

**Frontier prerequisites (if VM-VAR-040 ≠ none)**

* Adjacency exists for the Registry. (ReferenceError)
* Bands VM-VAR-042 (if used) are **ordered, non-overlapping, contiguous** over 0–100. (MethodConfigError)
* VM-VAR-047 is a non-empty subset of {land, bridge, water}; VM-VAR-048 is in its domain. (MethodConfigError)
* If autonomy actions are present, VM-VAR-046 maps them to valid **AP:** IDs. (ReferenceError)

**PR threshold range**

* VM-VAR-012 ∈ [0..10]. (MethodConfigError)

**Deterministic order source**

* Every **Option** has a unique order\_index (for deterministic ties). (ConstraintError)

**Audit**

* Full issue list with codes (e.g., HIERARCHY\_NOT\_TREE, WTA\_MAGNITUDE\_VIOLATION, MISSING\_ELIGIBLE\_ROLL, FRONTIER\_BANDS\_OVERLAP, …).

## **VM-FUN-003 — TabulateUnit**

**Purpose** Compute **UnitScores** per Unit according to ballot\_type and Doc 4A tabulation rules.

**Inputs**

* **LoadedContext**
* Unit slice (one or many Units)

**Preconditions**

* Validation passed or the caller is intentionally collecting partial data for an Invalid run report.

**Consumes variables**

* **VM-VAR-001..007**, **VM-VAR-012** (threshold applied later in allocation)

**Output**

* **UnitScores** per Unit:  
  + scores{ Option → natural tally } (plurality counts; approval approvals; score score\_sum; ranked method tallies)
  + turnout{ ballots\_cast, invalid\_or\_blank, valid\_ballots }
  + Audit payloads: **RoundLog** (IRV) or **PairwiseMatrix** (Condorcet)

**Postconditions**

* Denominator policy matches Doc 4A: **approval gate uses approval rate over valid ballots** (record both counts).

**Errors**

* MethodConfigError (missing ranked preferences; score scale mismatch)

**Audit**

* Per Unit: totals by Option, counts of exhausted ballots (IRV), pairwise edges (Condorcet).

## **VM-FUN-004 — AllocateUnit**

**Purpose** Transform **UnitScores** into **UnitAllocation** according to allocation\_method.

**Inputs**

* **UnitScores** for one Unit
* Unit metadata (magnitude)
* **LoadedContext.ParameterSet** (allocation fields)

**Preconditions**

* If winner\_take\_all then magnitude must be 1 (already validated).

**Consumes variables**

* **VM-VAR-010..015**, **VM-VAR-012** (apply PR entry threshold **before** seat math)

**Output**

* **UnitAllocation** { Option → seats\_or\_power } summing to magnitude (PR/LR) or 100% (WTA)

**Postconditions**

* For MMP, this function handles **local tier** seats if local magnitudes are defined here; top-up seats are assigned at correction level (Doc 4B).
* Last-seat ties resolved later (ResolveTies) or recorded as pending if policy requires.

**Errors**

* MethodConfigError (incoherent method + data), TieError (if the implementation chooses to surface a blocking last-seat tie here)

**Audit**

* Divisors/remainders trail (for D’Hondt/Sainte-Laguë/LR); thresholded-out options list; any tie candidate set.

## **VM-FUN-005 — AggregateHierarchy**

**Purpose** Roll **UnitAllocation** up the hierarchy to produce **AggregateResults** for District/Region/Country levels.

**Inputs**

* All **UnitAllocation**
* Registry hierarchy (parent pointers)
* Weighting data (population\_baseline if used)

**Preconditions**

* Weighted aggregation permitted only if all required baselines are present.

**Consumes variables**

* **VM-VAR-030 (weighting\_method)**, **VM-VAR-031 (aggregate\_level)**

**Output**

* **AggregateResults** per level: totals/seats by Option, shares, turnout metrics carried for gates, and the weighting method used.

**Postconditions**

* Reduction order is **stable** (by Unit ID then Option order) to maintain determinism.

**Errors**

* ConstraintError (missing/zero baseline under population weighting)

**Audit**

* For each level: child count, total seats by Option, notes on weighting.

## **VM-FUN-006 — ApplyDecisionRules**

**Purpose** Evaluate gates in fixed order and produce a **LegitimacyReport**.

**Inputs**

* **AggregateResults** (country and, if needed, regional)
* **LoadedContext.ParameterSet**
* Optionally per-Unit quorum results from Tabulate/Aggregate (turnout per Unit)

**Preconditions**

* None beyond prior stages.

**Consumes variables**

* **VM-VAR-020..027** (quorum, majority, double-majority & family), **VM-VAR-025** (symmetry), **VM-VAR-007** (denominator include blanks), **VM-VAR-029** (symmetry\_exceptions)
* *(Executive note)*: honor executive-specific settings; double-majority only if **VM-VAR-073=on**.

**Output**

* **LegitimacyReport** with:  
  + **Quorum:** national turnout vs **VM-VAR-020**; per-unit quorum flags if **VM-VAR-021>0**, noting **VM-VAR-021\_scope** effects
  + **Majority/Supermajority:** national support vs **VM-VAR-022** using denominators per Doc 4A (**approval rate** for approval)
  + **Double-majority:** affected-family definition/result vs **VM-VAR-023**; enforce rule: if frontier\_mode=none, family must be by\_list/by\_tag
  + **Symmetry:** respected/not respected; list **symmetry\_exceptions** if any
  + Overall **Pass/Fail** with explicit reasons and the raw numbers used

**Postconditions**

* If any gate **Fail** ⇒ mark run **Invalid** and signal the state machine to **skip MAP\_FRONTIER** (Doc 5A).

**Errors**

* Do **not** throw; record a QuorumError status internally when quorum fails (still a normal “Fail” in report). Only misconfiguration should have been caught in validation.

**Audit**

* Exact denominators used, computed percentages (pre-presentation rounding), thresholds, affected-region family membership, and symmetry exception list (if any).

### **Explicit dependencies & hidden-input ban**

* Every function above declares the **VM-VAR-###** it consumes and the artifacts it reads/writes.
* No function may rely on undeclared globals or hidden inputs; any external effect must be reflected in Inputs/Consumes/Audit.

**Done.** Functions 001–006 have complete contracts with Purpose, Inputs, Preconditions, Output, Postconditions, Errors, and Audit. Required validations (eligible roll, WTA→magnitude=1, frontier=none + double-majority=on ⇒ by\_list/by\_tag, bands non-overlapping, population baselines) are explicitly enforced in **VM-FUN-002**.

# **Doc 5C — Pipeline: Functions 007–013 (Frontier → Compare)**

**Scope:** Function-level contracts for **MAP\_FRONTIER → RESOLVE\_TIES → LABEL\_DECISIVENESS → BUILD\_RESULT → BUILD\_RUN\_RECORD → COMPARE\_SCENARIOS**.  
 **Alignment:** Logic per **Doc 4C**, artifacts per **Doc 5A**, entities per **Doc 1**.  
 **Determinism:** Same inputs + same engine (+ same seed, if used) ⇒ identical outputs (Docs **3/4**).

## **VM-FUN-007 — MapFrontier**

**Purpose** Translate per-Unit support into **FrontierMap** statuses using the chosen frontier mode, contiguity policies, protected-area rules, and per-Unit quorum scope.

**Inputs**

* **LoadedContext** (REG/Units/Adjacency/ParameterSet)
* **UnitScores** (for support %)
* Optional: per-Unit quorum outcomes from prior stages (turnout per Unit)

**Consumes variables**

* **VM-VAR-040..046** (frontier mode, cutoff/bands, autonomy mapping, protected overrides)
* **VM-VAR-047/048** (contiguity modes allowed, island exception rule)
* **VM-VAR-021 / VM-VAR-021\_scope** (per-Unit quorum and its scope)

**Preconditions**

* If frontier\_mode = none, caller must skip this function.
* **Adjacency** present for the Registry.
* Bands (if used) are ordered, non-overlapping, contiguous (validated earlier).

**Output**

* **FrontierMap** DB entity with per-Unit: status, band\_met?, contiguity diagnostics component\_id, mediation\_flag, enclave\_flag, and protected\_override\_used.

**Postconditions**

* Exactly **one** status per Unit.
* Contiguity computed using only edges in **VM-VAR-047**; islands handled per **VM-VAR-048**.
* If VM-VAR-021 > 0 and the Unit failed its per-Unit quorum, status = no\_change and flag is recorded (scope effects per 4C).
* Protected Units change only if **VM-VAR-045 = on**; overrides flagged.

**Errors**

* ReferenceError (missing Adjacency or AP mapping for autonomy bands)
* ConstraintError (attempted change in a protected Unit without override)
* ContiguityError (graph inconsistency); the function should **degrade to Mediation** where possible rather than abort.

**Audit**

* For each Unit: input support %, assigned status, band, whether quorum blocked change, protected override, mediation/enclave flags.
* Summary: number of components per action, count of mediation zones/enclaves, list of Units affected by protected overrides.

## **VM-FUN-008 — ResolveTies**

**Purpose** Resolve only **blocking** ties using the declared policy; log deterministic details (including seed when used).

**Inputs**

* Tie contexts emitted by earlier stages (e.g., WTA winner ties, last-seat ties, IRV elimination ties)
* **LoadedContext.ParameterSet** (tie policy, deterministic order, rng seed)
* Option metadata (including Option.order\_index)

**Consumes variables**

* **VM-VAR-050..052** (policy, deterministic order, rng\_seed)

**Preconditions**

* There exists at least one blocking tie to resolve.
* If tie\_policy = random, **rng\_seed** must be present.

**Output**

* Resolved allocations/decisions and a **TieLog** (to be embedded in **Result**), with entries:  
   {context, candidates[], policy, order\_or\_seed, winner}

**Postconditions**

* **Deterministic:** with the same inputs and **same seed**, output winners and TieLog are **identical** across OS/arch.
* Policy order enforced: status\_quo → deterministic\_order (by Option.order\_index, lower wins) → random(seed).

**Errors**

* TieUnresolvedError (should not occur with a valid policy/seed)
* MethodConfigError (random policy without seed)

**Audit**

* One TieLog entry per resolved tie; include candidate set order before resolution.

## **VM-FUN-009 — LabelDecisiveness**

**Purpose** Assign the final **DecisivenessLabel** (Decisive / Marginal / Invalid) with a verbatim reason string for the report.

**Inputs**

* **LegitimacyReport** (gate pass/fail + values)
* National margin (pp) computed at aggregation
* **FrontierMap** flags (if present): any mediation/enclave/protected overrides

**Consumes variables**

* **VM-VAR-062** (marginal band threshold, in pp)

**Preconditions**

* None (works for both valid and invalid runs).

**Output**

* **DecisivenessLabel**: {label, reason}

**Postconditions**

* If any gate failed (or validation failed earlier), label = **Invalid**.
* Else if national margin < **VM-VAR-062** **or** any frontier flags present, label = **Marginal**.
* Else label = **Decisive**.
* Reason text is concise and ready for Doc 7.

**Errors**

* — (pure computation)

**Audit**

* Margin value used; list of frontier flags that triggered “Marginal” (if any).

## **VM-FUN-010 — BuildResults**

**Purpose** Assemble the canonical **Result** DB entity from all prior artifacts.

**Inputs**

* **LoadedContext** identifiers (REG/TLY/PS),
* **UnitScores**, **UnitAllocation**, **AggregateResults**, **LegitimacyReport**, optional **FrontierMap**, **TieLog**, **DecisivenessLabel**

**Preconditions**

* All required artifacts present; when gates failed, **FrontierMap** may be absent by design.

**Output**

* **Result** DB entity containing: per-Unit blocks (with validity flags), level aggregates, gates (values + pass/fail), TieLog, label, pointer to FrontierMap (if any)

**Postconditions**

* Seat totals per Unit match Unit.magnitude (or 100% for WTA).
* Per-Unit validity flags are set exactly as enumerated in Doc 1B:  
   unit\_data\_ok, unit\_quorum\_met, unit\_pr\_threshold\_met, protected\_override\_used, mediation\_flagged.
* All values reflect the same denominators as used in gate calculations (Doc 4A/4C).

**Errors**

* — (assembly only; upstream stages guarantee coherence)

**Audit**

* Checksums/hashes of major sections (informational); counts of Units/Options/levels; references to input IDs.

## **VM-FUN-011 — BuildRunRecord**

**Purpose** Create the **RunRecord** attesting to reproducibility and inputs used.

**Inputs**

* IDs & versions: **FormulaID**, **EngineVersion**, reg\_id, tally\_id, parameter\_set\_id
* Determinism settings (rounding policy, ordering basis, **rng\_seed** if used)
* Pointers: result\_id, optional frontier\_map\_id
* Timestamps (UTC)

**Preconditions**

* A **Result** exists.

**Output**

* **RunRecord** DB entity

**Postconditions**

* Contains all identifiers required to reproduce the run offline; **rng\_seed** recorded if any tie used random policy.

**Errors**

* — (assembly only)

**Audit**

* Human-readable summary line mirroring Doc 7 “Integrity & Reproducibility” section.

## **VM-FUN-012 — BatchRun *(helper; unchanged)***

**Purpose** Execute VM-FUN-001…011 across multiple ParameterSets and/or tallies; collect Results & RunRecords for comparison.

**Note**

* Helper; not required for single-scenario execution.

## **VM-FUN-013 — CompareScenarios (REQUIRED)**

**Purpose** Produce the **sensitivity outputs** used in Doc 7’s “±1/±5 pp” table and side-by-side comparisons of scenarios.

**Inputs**

* A baseline **Result** (with its **ParameterSet**)
* A set of **delta ParameterSets** derived from the baseline by applying **±1 pp** and **±5 pp** adjustments to the relevant threshold variables (e.g., **VM-VAR-020, 022, 023, 041**, and band boundaries in **VM-VAR-042** where applicable)

**Preconditions**

* The baseline run completed (Decisive or Marginal or Invalid).
* Deltas are well-formed and differ only in the intended variables.

**Output**

* A **ComparisonBundle** containing:  
  + Per-scenario **Result IDs** and labels
  + **Flip report**: which thresholds (±1/±5 pp) flipped any gate, changed the label, or altered the seat/power outcome
  + **Frontier diffs**: counts of Units whose status changed

**Postconditions**

* All comparisons are deterministic (same deltas ⇒ same diffs).
* The bundle is sufficient for the Report layer to render the sensitivity mini-table (Doc 7A/7B).

**Errors**

* MethodConfigError if deltas change variables outside the allowed set for sensitivity.
* ReferenceError if a delta references an unknown variable ID.

**Audit**

* List of deltas applied; per-delta hash; brief notes on the first flip-point for each dimension (e.g., national threshold, regional threshold, cutoff).

## **Determinism guarantees (for this part)**

* **Stable iteration order** at every step (Unit IDs, then Option order\_index).
* **Round half to even** at defined comparison points only.
* **Random policy** in ties uses only **VM-VAR-052 rng\_seed**; given the same seed, the **TieLog** and outputs are **byte-identical** across OS/arch.
* No network or time-dependent data enters any function’s logic (timestamps only in **RunRecord**).

**Done.** Functions **007–013** are fully specified, enforce contiguity modes and island rules, respect per-Unit quorum scope and protected overrides, log tie policy/seed, and make **CompareScenarios** **required** to power the Doc 7 sensitivity table.