# **Doc 1A — DB Definition: Entities & IDs (Skeleton)**

**Scope:** name the entities, fix their **stable ID formats**, and lock a few mandatory fields we added to prevent ambiguity later. This is implementation-neutral (no SQL/JSON).

**Rules:**

* **IDs are never re-used.**
* Provenance is required where noted.
* Names match Docs **2/4/5/7** (e.g., “**BallotTally label**”).

## **A) Canonical Entities (v1)**

**Core (always present)**

1. **DivisionRegistry** — versioned list of Units + hierarchy.
2. **Unit** — atomic decision unit within a registry.
3. **Option** — selectable outcome (A/B/C/D, Status Quo…).
4. **BallotTally** — per-unit tallies for a specific election context.
5. **ParameterSet** — frozen variables used for a run.
6. **Result** — computed outcomes (per-unit + aggregates + gates + label).
7. **RunRecord** — provenance/attestation for one run.

**Optional (when mapping borders/powers)** 8. **FrontierMap** — per-unit status + contiguity flags.  
 9. **AutonomyPackage** — named bundle of devolved powers.

**Support** 10. **Adjacency** — explicit neighbor graph for contiguity checks.

## **B) Stable ID Formats (and the few locked fields)**

Examples show shape; angle brackets are placeholders. All IDs are ASCII, case-sensitive, colon-separated. The **DivisionRegistry ID** is referenced inside several others.

### **1) DivisionRegistry**

* **ID:** REG:<name>:<version>  
  + *Example:* REG:UkraineAdmin:2021
* **Provenance (required fields):** source, published\_date, notes.

### **2) Unit *(includes new baseline fields)***

* **ID:** U:<REG\_ID>:<path> where <path> encodes the hierarchy (e.g., ISO/admin codes).  
  + *Example:* U:REG:UkraineAdmin:2021:UA:Donetsk:05
* **Locked fields:**
  + eligible\_roll *(integer, ≥0)* — count of eligible voters in the unit.
  + population\_baseline *(integer, ≥0)* — baseline population used for weighting when enabled.
  + population\_baseline\_year *(YYYY)* — provenance for the baseline.
* **Notes:** eligible\_roll + its provenance live at Unit level; a **Registry-level** note may state the global roll policy.

### **3) Option *(includes deterministic order field)***

* **ID:** OPT:<slug>  
  + *Example:* OPT:A, OPT:StatusQuo
* **Locked field:** order\_index *(integer; lower value = higher precedence in deterministic tie policy)*.

### **4) BallotTally *(dataset + label)***

* **ID:** TLY:<jurisdiction\_or\_event>:<label>:v<version>  
  + *Example:* TLY:UA:NationalPlebiscite2025:v1
* **Human label (for reports):** label (free text) — the **“BallotTally label”** referenced in Runs/Reports.
* **Links:** references **REG\_ID** and Option set used.

### **5) ParameterSet**

* **ID:** PS:<name>:v<semver>  
  + *Example:* PS:Baseline:v1.0.0
* **SemVer is part of the ID**; ParameterSets are immutable.

### **6) Result**

* **ID:** RES:<short-hash> *(derived from inputs + engine + formula lock)*

### **7) RunRecord**

* **ID:** RUN:<utc\_timestamp>-<short-hash>  
  + *Example:* RUN:2025-08-11T14-07-00Z-a1b2c3

### **8) FrontierMap *(optional)***

* **ID:** FR:<short-hash>

### **9) AutonomyPackage *(optional)***

* **ID:** AP:<name>:v<semver>  
  + *Example:* AP:LanguageTaxBase:v1.0

### **10) Adjacency *(support)***

* **Dataset ID:** ADJMAP:<REG\_ID>
* **Row identity (implicit):** ordered pair U1–U2 with a type field (land/bridge/water).

## **C) Minimal Field Lock-ins (to avoid drift)**

These are **intentionally included at the skeleton level** so downstream docs align:

* **DivisionRegistry**: id, name, version, provenance{source,published\_date,notes}.
* **Unit**: id, reg\_id, parent\_unit\_id|null, level, magnitude (≥1),  
   eligible\_roll, population\_baseline, population\_baseline\_year, flags {protected\_area?}.
* **Option**: id, display\_name, is\_status\_quo?, order\_index.
* **BallotTally**: id, label, reg\_id, ballot\_type, references to per-unit/option tallies (shape detailed in Doc 1B).
* **ParameterSet**: id, name, version, **variables snapshot** (values for VM-VAR-###).
* **Result**: id, references {reg\_id, ballot\_tally\_id, parameter\_set\_id}, pointer to **FrontierMap** (if any).
* **RunRecord**: id, identifiers {FormulaID, EngineVersion, reg\_id, ballot\_tally\_id, parameter\_set\_id}, determinism settings {rounding, ordering, rng\_seed?}, timestamps, pointers {result\_id, frontier\_map\_id?}.
* **FrontierMap**: id, per-unit status, flags {mediation,enclave,protected\_override\_used}, band met.
* **AutonomyPackage**: id, name, version, powers[], review\_period\_years.
* **Adjacency**: adjacency\_map\_id, rows {unit\_id\_a, unit\_id\_b, type}.

## **D) ID & Provenance Guarantees**

* **No ID reuse.** New versions/new sources ⇒ new IDs (e.g., REG:…:2026).
* **Traceability:** RunRecord must cite **all** input IDs and produce the **Result/FrontierMap** IDs.
* **Provenance required** for DivisionRegistry (source/date) and **population baselines** (year).
* **Deterministic order** comes from **Option.order\_index** (Doc 2/4/5 use it for deterministic tie policy).

**Done:** Entities named; **stable ID formats fixed**; new fields (eligible\_roll, population\_baseline(+year), Option.order\_index) locked and consistent with Docs **2/4/5/7**.

# **Doc 1B — DB Definition: Entity Details**

**Scope:** Per-entity definitions, key fields, constraints, relationships, and provenance for the voting machine. Names and semantics align with Docs **1A/2/4/5/7**.

**Global rules (apply to multiple entities):**

* **Hierarchy:** Units form a **tree** with a single root per **DivisionRegistry**.
* **Magnitude:** Unit.magnitude ≥ 1. If allocation\_method=winner\_take\_all, every Unit.magnitude = 1 (validated in VM-FUN-002).
* **Tally sanity:** For each Unit in a given BallotTally:  
   sum(valid tallies across options) + invalid\_or\_blank ≤ ballots\_cast.
* **Population weighting:** If weighting\_method=population\_baseline (VM-VAR-030), each aggregated Unit must provide a **positive** population\_baseline and population\_baseline\_year.
* **Deterministic order:** Option.order\_index sets precedence under **deterministic tie policy** (lower index wins before random; see VM-VAR-051 & VM-FUN-008).

## **VM-DB-001 DivisionRegistry**

**Definition.** Versioned catalogue of Units with their parent–child hierarchy for a run.

**Key fields.**

* id (REG::) · name · version
* levels[] (ordered labels, e.g., Country/Region/District/Neighborhood)
* constraints: e.g., “contiguity required” as a registry note (informational)
* **Provenance:** source (human-readable), published\_date (YYYY-MM-DD), notes

**Constraints.**

* Exactly one root Unit; no cycles; each Unit belongs to exactly one DivisionRegistry.

**Relationships.**

* 1—∞ to **Unit**, 1—∞ to **Adjacency** (rows scoped to this registry)
* Referenced by **BallotTally**, **Result**, **RunRecord**, **FrontierMap**

## **VM-DB-002 Unit**

**Definition.** Atomic decision unit within a DivisionRegistry.

**Key fields.**

* id (U:<REG\_ID>:) · reg\_id
* name (human label) · level (one of DivisionRegistry.levels[])
* parent\_unit\_id (nullable only for root)
* magnitude (integer ≥ 1; seats/power slots)
* **Weighting:** population\_baseline (int ≥ 0) · population\_baseline\_year (YYYY)
* **Roll:** eligible\_roll (int ≥ 0)
* **Flags:** protected\_area (bool)

**Constraints.**

* Parent must exist within same reg\_id; eligible\_roll ≥ 0; if used in population weighting, population\_baseline > 0.

**Relationships.**

* Belongs to **DivisionRegistry**
* Referenced by **BallotTally**, **Result**, **FrontierMap**, **Adjacency**

**Provenance.**

* population\_baseline\_year documents the baseline vintage; registry-level notes may cite roll policy (VM-VAR-028).

## **VM-DB-003 Option**

**Definition.** A selectable ballot option (e.g., A/B/C/D or Status Quo).

**Key fields.**

* id (OPT:) · display\_name
* is\_status\_quo (bool)
* **Deterministic tie:** order\_index (integer; **lower wins** when tie policy = deterministic)

**Constraints.**

* order\_index must be unique per election context.

**Relationships.**

* Referenced by **BallotTally**, **Result**, and tie handling in **RunRecord/TieLog**

## **VM-DB-004 BallotTally**

**Definition.** Per-Unit vote tallies consistent with the ballot type for a particular event/dataset.

**Key fields.**

* id (TLY:…:vX) · label (human-readable “BallotTally label”)
* reg\_id · ballot\_type (VM-VAR-001)
* **Turnout:** ballots\_cast (int ≥ 0) · invalid\_or\_blank (int ≥ 0)
* **Per-option tallies** (shape depends on ballot\_type):  
  + **plurality/approval:** count (int ≥ 0)
  + **score:** score\_sum (int ≥ 0), ballots\_counted (int ≥ 0)
  + **ranked:** rankings structure sufficient to derive **RoundLog**/**PairwiseMatrix** at run-time (not stored verbatim here)
* (Optional) notes, provenance (source, method, date)

**Constraints.**

* **Tally sanity rule** holds per Unit (see global rules).
* Ballot type in tallies must match ballot\_type.

**Relationships.**

* Input to **Result**; referenced by **RunRecord**
* Uses **Option** set; scoped to **DivisionRegistry**

**Provenance.**

* Required for public reporting integrity: who compiled tallies, from what original source, and when.

## **VM-DB-005 ParameterSet**

**Definition.** Frozen snapshot of variables (VM-VAR-###) that govern a run.

**Key fields.**

* id (PS::vSemVer) · name · version
* **Variables:** full key–value map of all used **VM-VAR-###** (Docs 2A/2C)
* description / intent note

**Constraints.**

* **Immutable** once published; coherent combinations enforced in validation (VM-FUN-002).

**Relationships.**

* Read by the engine to produce a **Result**
* Cited in **RunRecord** and **Report**

## **VM-DB-006 Result**

**Definition.** Official computed outcome bundle for a run.

**Top-level fields.**

* id (RES:)
* **Inputs:** reg\_id, ballot\_tally\_id, parameter\_set\_id
* **Aggregates by level:** totals/shares per Option; turnout; weighting used
* **Decision gates:** pass/fail for quorum, majority/supermajority, double-majority, symmetry, with computed denominators/thresholds
* **TieLog**: entries {context, candidates, policy, order/seed, winner}
* **Label:** Decisive | Marginal | Invalid + rationale
* Optional pointer: frontier\_map\_id

**Per-Unit block (Result.UnitBlock[]).**

* unit\_id
* **Tabulation:** scores (by Option; natural totals), turnout {ballots\_cast, invalid\_or\_blank, valid\_ballots}
* **Allocation:** seats\_or\_power (by Option; sums to magnitude or 100%)
* **Flags (validity) — enumerate exactly:**
  + unit\_data\_ok (bool) — structural/tally checks passed for this Unit
  + unit\_quorum\_met (bool) — if per-unit quorum applies (VM-VAR-021)
  + unit\_pr\_threshold\_met (bool) — if PR threshold applied & met
  + protected\_override\_used (bool) — true only if VM-VAR-045 allowed a protected change
  + mediation\_flagged (bool) — contiguity/island mediation affected this Unit’s status

**Constraints.**

* Seats sum equals Unit.magnitude (PR) or 100% power (WTA).
* Aggregates are consistent with per-Unit data.

**Relationships.**

* Written by pipeline; referenced by **RunRecord** and **Report**; optionally linked to **FrontierMap**.

## **VM-DB-007 RunRecord**

**Definition.** Provenance/attestation for reproducing a run.

**Key fields.**

* id (RUN:-)
* **Identifiers:** FormulaID (hash of normative Doc 4 sections), EngineVersion (Doc 3), reg\_id, ballot\_tally\_id, parameter\_set\_id
* **Determinism settings:** rounding mode (fixed), ordering policy, rng\_seed (if used), option order source
* **Timestamps:** start/end in UTC
* **Outputs:** result\_id, optional frontier\_map\_id
* **Environment (optional):** brief platform string

**Constraints.**

* Sufficient to **reproduce** results byte-for-byte with the same engine.

**Relationships.**

* 1—1 with **Result**; optional 1—1 with **FrontierMap**

## **VM-DB-008 FrontierMap (optional)**

**Definition.** Per-Unit status after applying frontier mapping (binary/sliding/ladder) and contiguity checks.

**Key fields.**

* id (FR:) · reg\_id · parameter\_set\_id
* **Per-Unit status:** one of {no\_change, autonomy(AP:id), phased\_change, immediate\_change}
* **Band met** (if sliding/ladder)
* **Contiguity diagnostics:**
  + contiguity\_component\_id (cluster label)
  + mediation\_flag (bool) — this unit is in an island/violates contiguity policy
  + enclave\_flag (bool) — enclave detected under policy
  + protected\_override\_used (bool) — if a change occurred with override
* **Counters (summary):** number of mediation zones/enclaves/protected overrides

**Constraints.**

* Exactly one status per Unit.
* Contiguity evaluation uses **Adjacency** and contiguity policies (VM-VAR-047/048).

**Relationships.**

* Derived from **Result** & **ParameterSet**; referenced by **RunRecord** and **Report**.

## **VM-DB-009 AutonomyPackage (optional)**

**Definition.** Named bundle of devolved powers used in frontier outcomes.

**Key fields.**

* id (AP::vSemVer) · name · version
* powers[] (e.g., education, language, taxation, policing, judiciary)
* review\_period\_years · escalation\_triggers / de-escalation\_triggers (informational text)

**Constraints.**

* Stable semantics across runs for the same version.

**Relationships.**

* Referenced by **FrontierMap** statuses when action = autonomy(AP:...)
* Mentioned in **ParameterSet** bands (VM-VAR-046)

## **VM-DB-010 Adjacency (support)**

**Definition.** Explicit neighbor relationships between Units for contiguity checks.

**Key fields.**

* adjacency\_map\_id (ADJMAP:<REG\_ID>)
* Rows: unit\_id\_a, unit\_id\_b, type ∈ {land, bridge, water}, optional notes

**Constraints.**

* Symmetric: if (A,B) exists, treat (B,A) equivalently.
* Both Units must belong to the same reg\_id.

**Relationships.**

* Owned by **DivisionRegistry**; consumed by **FrontierMap** logic and validation.

## **Cross-references (where these are used)**

* **Variables:** VM-VAR-030 (weighting uses population\_baseline), VM-VAR-028 (roll policy uses eligible\_roll), VM-VAR-047/048 (contiguity rules), VM-VAR-045 (protected overrides).
* **Functions:** VM-FUN-002 (ValidateInputs), -003 (TabulateUnit), -004 (AllocateUnit), -005 (AggregateHierarchy), -007 (MapFrontier), -010 (BuildResults), -011 (BuildRunRecord).
* **Report:** Doc 7 reads Result, RunRecord, and FrontierMap fields verbatim; per-unit flags drive the **Legitimacy Panel** and **Frontier** notes.

**Done:** Each entity now has a self-contained definition with fields, constraints, relationships, and provenance; validity flags are enumerated; adjacency types are fixed (land/bridge/water); global sanity and determinism constraints are stated.

# **Doc 1C — DB Definition: Relationships & Global Constraints**

**Scope:** Entity–relationship map for the Voting Machine data model and the invariants that must hold across all runs. Terminology matches Docs **4/5/7**; entities match Doc **1A/1B**.

## **1) Entity–Relationship Map (cardinalities)**

### **Core graph**

* **DivisionRegistry (REG)**
  + **1 → ∞ Units (Unit)** — Units belong to exactly one REG; Units form a **tree** (see §2).
  + **1 → ∞ Adjacency rows (Adjacency)** — Each row links two Units within the same REG.
  + **1 → ∞ BallotTally datasets (BallotTally)** — Tallies are scoped to the REG and its Option set.
  + **1 → ∞ Results (Result)** — Multiple Results over time can reference the same REG via different inputs/ParameterSets.
* **Unit**
  + **∞ → 1 DivisionRegistry** (owner).
  + **Referenced by** BallotTally tallies, Result.UnitBlocks, FrontierMap status, Adjacency rows.
* **Option**
  + **Many-to-many** with BallotTally (tallies per Unit×Option).
  + **Many-to-many** with Result (allocations per Unit×Option).
  + Ordered by **Option.order\_index** (used in deterministic ties).
* **BallotTally (TLY)**
  + **∞ Units × ∞ Options** tallies (logical rows).
  + **1 → 1 DivisionRegistry** (by reg\_id).
  + **∞ → 1 Result** (as input; a single TLY can feed many runs/results).
* **ParameterSet (PS)**
  + **1 → ∞ Results** (each run freezes a PS).
  + **1 → ∞ RunRecords** (each run produces a record).
* **Result (RES)**
  + **1 → 1 RunRecord** (provenance).
  + **0..1 → 1 FrontierMap** (optional link when mapping is enabled).
  + **∞ UnitBlocks** (one per Unit), each with per-Option scores/allocations and flags.
* **RunRecord (RUN)**
  + **1 → 1 Result** (the run it attests).
  + **0..1 → 1 FrontierMap** (the map produced in the run, if any).
  + **References:** REG, TLY, PS, FormulaID, EngineVersion, RNG seed (if used).
* **FrontierMap (FR)** *(optional)*
  + **∞ Unit statuses** (exactly one status per Unit).
  + **∞ → 0..∞ AutonomyPackage** references via actions where applicable.
  + **1 → 1 ParameterSet** (values used to derive it).
  + **1 → 1 DivisionRegistry** (scope).
* **AutonomyPackage (AP)** *(optional)*
  + **0..∞ FrontierMap** entries may reference a given AP version.
* **Adjacency (ADJMAP:REG)** *(support)*
  + Rows {Unit A, Unit B, type∈{land, bridge, water}}; symmetric by interpretation.

## **2) Hierarchy & Ownership Rules**

* **Unit tree:** Exactly **one root** Unit per REG; every non-root Unit has **one parent** within the same REG; no cycles; path-encoding stable (Doc 1A).
* **Adjacency ownership:** All Adjacency rows are **scoped to one REG** and must reference existing Units in that REG.
* **Result/RunRecord/FrontierMap linkage:**
  + Each **RunRecord** must point to **exactly one Result** (and optionally one FrontierMap).
  + Each **Result** must point back to the **exact inputs** used: reg\_id, tally\_id, parameter\_set\_id.
  + Each **FrontierMap** must point to the **REG** and **PS** it used; **Result** optionally points to the produced **FrontierMap**.

## **3) Global Constraints (invariants across the DB)**

### **Identity, versioning, provenance**

1. **IDs never reused.** New sources/versions create new IDs (REG, TLY, PS, AP, etc.).
2. **Provenance required:** DivisionRegistry (source, published\_date); population baselines (population\_baseline\_year).
3. **ParameterSet immutability:** PS content is frozen by id (SemVer in ID).

### **Unit, tallies, and magnitudes**

1. **Magnitude:** Unit.magnitude ≥ 1. If allocation\_method = winner\_take\_all, then **every Unit.magnitude = 1** for the run (else the run is Invalid).
2. **Tally sanity (per Unit per TLY):** Σ(valid tallies over Options) + invalid\_or\_blank ≤ ballots\_cast (all non-negative integers).
3. **Ballot type coherence:** BallotTally.ballot\_type must match the run’s VM-VAR-001. Ranked tallies present when needed; score scale consistent with VM-VAR-002..003.

### **Weighting & rolls**

1. **Population weighting readiness:** If VM-VAR-030 = population\_baseline, every aggregated Unit must have **positive** population\_baseline and a population\_baseline\_year.
2. **Eligible roll readiness:** If VM-VAR-020 > 0 (quorum in effect) or VM-VAR-021 > 0, each aggregated Unit must have eligible\_roll with eligible\_roll ≥ ballots\_cast. The **math of turnout** always uses eligible\_roll (Doc 4C).

### **Contiguity & protections**

1. **Adjacency type domain:** Adjacency.type ∈ {land, bridge, water} only.
2. **Contiguity evaluation:** Frontier contiguity must use **only** the edge types allowed by VM-VAR-047; islands handled per VM-VAR-048.
3. **Protected areas:** Units flagged protected\_area = true **cannot change status** via FrontierMap unless protected\_override\_allowed is set in the ParameterSet; any override must be flagged in **Result.UnitBlock** and **FrontierMap**.

### **Determinism & ties**

1. **Stable ordering:** Any operation that depends on ordering uses a **total order** (Unit IDs; Options by order\_index then ID).
2. **Rounding policy:** Internal comparisons use **round half to even**; presentation rounding happens only in reporting.
3. **Randomness isolation:** Randomness is allowed **only** for tie resolution when tie\_policy = random; the **rng\_seed** must be recorded in **RunRecord** and the **TieLog** must appear in **Result**.
4. **Option order uniqueness:** Option.order\_index must be **unique** within the Option set for the run.

### **Frontier & double-majority scoping (consistency rules)**

1. **Single frontier mode:** At most **one** frontier mode per run (or none).
2. **Double-majority without frontier:** If double\_majority\_enabled = on **and** frontier\_mode = none, the affected-region family **must** be provided via by\_list or by\_tag (not by\_proposed\_change).

## **4) Integrity links to reporting (Doc 7)**

* Everything shown in the Report must be derivable from **Result**, optional **FrontierMap**, and **RunRecord**.
* **Legitimacy panel values** (turnout/support/thresholds) must appear in **Result** with the gate Pass/Fail flags used by the Report templates.
* **Frontier diagnostics** (mediation/enclaves/protected overrides) must be present in **FrontierMap** and mirrored in **Result.UnitBlock.mediation\_flagged / protected\_override\_used** for consistency.

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

* Cardinalities and ownership rules above cover **all** entity links.
* The hierarchy/tree, adjacency scoping, and Result↔RunRecord↔FrontierMap links are unambiguous.
* The global constraints enumerate magnitude, tally sanity, weighting & roll readiness, contiguity & protections, and determinism/ties—consistent with Docs **4/5/7**.

**Status:** ER map and invariants are fixed and implementation-ready.