1) MVP PRD — Climate Risk Insurance System (Wedge: Exposure + Accumulation Control Tower)

1.1 Product objective

Deliver a decision-operational platform that makes exposure trustworthy, accumulation visible, and portfolio drift explainable—with governance artifacts suitable for regulated insurer environments.

1.2 Target users and buyers

Primary personas (MVP)

1. **Head of Cat Risk / Cat Analyst (Primary user)**
   - Needs: accumulation rollups, breach alerts, drill-down lists, portfolio drift.
2. **Portfolio Management Lead / Risk Aggregation Owner**
   - Needs: live monitoring, threshold governance, reporting packages, repeatability.
3. **Underwriting Ops / Data Steward**
   - Needs: exceptions queue, data quality scoring, correction loop.
4. **Model Risk / Compliance / Internal Audit (Secondary user)**
   - Needs: lineage, versioning, audit logs, reproducible outputs.

Economic buyers (MVP motion)

- CRO or CUO (value: reduced tail surprises, improved control posture)
- CFO influence for reinsurance readiness (later expansion)

1.3 MVP scope (what we will build)

Core workflows (MVP)

1. **Exposure onboarding**
    - ○ Upload exposure file(s) → map fields → validate → canonicalize → create immutable exposure version.
2. **Geocoding + data quality scoring**
    - ○ Standardize address → geocode → assign confidence and quality tiers → produce exceptions.
3. **Hazard overlays**
    - ○ Join hazard datasets to locations → produce hazard attributes (bands/percentiles/categories) with dataset versioning.
4. **Accumulation rollups & alerts**
    - ○ Aggregate by region/segment/hazard bands → set thresholds → trigger breaches → drill-down.
5. **Portfolio drift**
    - ○ Compare exposure version A vs B → explain changes in counts/TIV mix/concentration/breaches.
6. **Governance**
    - ○ Version registry (inputs/configs/datasets) + run registry (reproducible executions) + append-only audit events.

Deliverables

- Web app (ops console + accumulation dashboards + drift report + governance views)
- API-first platform (all actions available by API)
- Exportable artifacts (exceptions report, breach report, drift report, run lineage report)

1.4 Out of scope (explicit non-goals for MVP)

- Full probabilistic catastrophe loss modeling engine (EP curves generation, event catalogs) beyond basic ingest of vendor outputs
- Pricing indication engine / rate adequacy at-location loss costs (Phase 2)
- Reinsurance structure optimization engine (Phase 3)
- Complex correlation models and dependency graphs beyond transparent aggregation (introduce later)
- Fully automated regulatory scenario reporting packages (Phase 4)

1.5 Success criteria (measurable)

Time-to-value

- Exposure file to accumulation dashboard in **< 24 hours** for first implementation, **< 1 hour** after configuration is established.

Data quality operations

- Reduce “unknown/ungeocoded” locations by **X%** over 30 days (tracked).
- Exceptions queue resolved cycle time decreases week-over-week.

Risk control outcomes

- Breach detection happens **on ingestion**, not quarterly.
- Drift reports used in monthly governance routines.

Governance acceptance

- Ability to reproduce any dashboard figure from immutable versions and run config.
- Audit log shows who changed what and when.

1.6 Product requirements (functional)

R1 — Exposure ingestion and canonicalization

- Accept CSV/XLSX (MVP) and optionally S3/object storage drop.
- Field mapping UI with reusable templates per tenant/LOB.
- Validation rules with severity: **ERROR / WARN / INFO**.
- Canonical output: `ExposureVersion` containing `Locations` and optional `Accounts/Policies` linkage.

Acceptance criteria

- User can upload, map, validate, and produce a versioned dataset.
- Validation output includes row-level errors and field-level summaries.
- Canonical dataset is immutable after version creation.

R2 — Geocoding and data quality scoring

- Geocode pipeline produces:
  - standardized address
  - lat/lon
  - geocode method and confidence
- Data quality score per location and aggregate:
  - completeness (required fields)
  - geocode confidence tier
  - financial sanity checks (e.g., TIV present, non-negative)

    - occupancy/construction validity (if present)

Acceptance criteria

- Each location has `quality_tier` and `quality_reasons[]`.
- Exceptions report export includes remediation hints (missing fields, invalid formats, low confidence).

R3 — Hazard overlays (dataset-versioned)

- Support multiple hazard datasets by peril proxy.
- Overlay results stored as:
    - dataset version
    - method (spatial join / raster sample / lookup)
    - hazard attributes (band/percentile/category)

Acceptance criteria

- Any overlay result references an immutable hazard dataset version.
- A portfolio run always records which hazard dataset versions were used.

R4 — Accumulation rollups and threshold alerts

- Rollups:
    - By geography: country/state/county/CRESTA (configurable)
    - By business segmentation: LOB/product/occupancy/COG (if provided)
    - By hazard: hazard band/category

  - By data quality tier
- Measures:
  - count locations
  - sum TIV
  - sum limit (if provided)
  - sum premium (optional)
- Alerts:
  - threshold definitions (e.g., TIV > $X in region/hazard band)
  - growth-rate thresholds vs prior exposure version
  - email/SIEM integration **not required** in MVP; in-app notifications required

Acceptance criteria

- Threshold breaches are computed deterministically and link to underlying locations.
- Breach lists are exportable.

R5 — Portfolio drift reporting

- Compare two exposure versions:
  - delta counts/TIV by rollup dimensions
  - new vs removed vs modified locations
  - breach changes (new breaches, resolved breaches, worsened breaches)

Acceptance criteria

- Drift report can attribute changes to a set of location IDs.

- Report can be reproduced later with stored versions/config.

R6 — Governance primitives

- Version registry: exposures, hazard datasets, configs
- Run registry: immutable “execution objects”
- Audit log: append-only event stream for sensitive actions

Acceptance criteria

- For any dashboard number, a lineage view shows:
  - exposure_version_id
  - hazard_dataset_version_ids
  - rollup_config_id
  - run_id
  - user_id + timestamp

1.7 UX surfaces (MVP screens)

1. Upload & Mapping
2. Validation Summary + Row-level error explorer
3. Exceptions Queue (filter/sort/assign/export)
4. Accumulation Dashboard (rollups + drill-down)
5. Threshold Builder + Breach List
6. Drift Report (vA vs vB)

7. Governance: Versions + Runs + Lineage

8. Audit Log Viewer (basic)

1.8 Rollout plan

- **Design partner pilot**: one LOB, one region, one hazard set.

- Week 1: ingestion + mapping templates

- Week 2: overlays + rollups

- Week 3: thresholds + drift

- Week 4: governance hardening + export/reporting
  (Exact weeks depend on team size; sequence is the key.)

1.9 Key risks and mitigations

- **Data heterogeneity:** enforce templates + strong validation + exceptions workflow.

- **Trust deficit:** quality scoring + deterministic pipelines + lineage.

- **Scope creep to cat modeling:** keep hazard overlays as proxies; ingest vendor results rather than replicate.

- **Security review friction:** ship baseline controls early (RBAC, audit, encryption, tenant isolation).