**01-db\_schema.md — *Canonical Database Schema for EkoH & Smart Vote*** *(PostgreSQL + Django-ORM conventions; standard created\_at / updated\_at timestamp columns are implicit unless noted otherwise.)*

## **0 Overview & Conventions**

* **Scope** – only the new tables owned by the standalone EkoH (expertise & ethics) and Smart Vote modules. Shared tables such as auth\_user live in the Konnaxion core schema and are referenced by FK.
* **Naming** – singular CamelCase model → snake\_case table; FK columns carry \_id; composite uniqueness enforced via UniqueConstraint.
* **Immutability** – reference tables (expertise\_category, vote\_modality, ENUM types) are append-only.
* **Retention & Partitioning** – see §6.
* **ERD** – text diagram in §7 plus a placeholder for a PNG/SVG in docs/assets/.

Note – Module-local tables: The following audit / integrity tables are local to the EkoH + Smart Vote module and therefore do not appear in the global Konnaxion canonical table list. They are safe to keep here as long as their names do not clash with core-schema objects.

## **1 EkoH (Expertise & Ethics)**

| **Table (Model)** | **Purpose** | **Key columns & constraints** |
| --- | --- | --- |
| **expertise\_category**  *(ExpertiseCategory)* | UNESCO / ISCED-F domain catalogue. | id PK, name VARCHAR UNIQUE |
| **user\_expertise\_score**  *(UserExpertiseScore)* | Per-user merit score per domain. | id PK, user\_id FK→auth\_user, category\_id FK→expertise\_category, raw\_score NUMERIC, weighted\_score NUMERIC, **UNIQUE**(user\_id,category\_id) |
| **user\_ethics\_score**  *(UserEthicsScore)* | Behaviour-based multiplier ≥ 0. | user\_id PK+FK→auth\_user, ethical\_score NUMERIC |
| **score\_configuration**  *(ScoreConfiguration)* | Tunable coefficients for the merit engine. | id PK, weight\_name VARCHAR, weight\_value NUMERIC, field VARCHAR NULLABLE |
| **context\_analysis\_log**  *(ContextAnalysisLog)* | Explainable-AI adjustment audit. | id PK, entity\_type VARCHAR, entity\_id UUID, field VARCHAR, input\_metadata JSONB, adjustments\_applied JSONB |
| **confidentiality\_setting**  *(ConfidentialitySetting)* | User privacy level. | user\_id PK+FK→auth\_user, level ekoh\_privacy\_level\_enum |
| **score\_history**  *(ScoreHistory)* | Immutable audit of every score change. | id PK, merit\_score\_id FK→user\_expertise\_score, old\_value NUMERIC, new\_value NUMERIC, change\_reason TEXT |

*Index notes* – composite index (category\_id, weighted\_score DESC) accelerates “top experts per domain”.

## **2 Smart Vote (Weighted Voting Engine)**

| **Table (Model)** | **Purpose** | **Key columns & constraints** |
| --- | --- | --- |
| **vote**  *(Vote)* | One raw ballot cast by a user. | id PK, user\_id FK→auth\_user, target\_type VARCHAR, target\_id UUID, modality\_id FK→vote\_modality, raw\_value NUMERIC, weighted\_value NUMERIC, **UNIQUE**(user\_id,target\_type,target\_id) |
| **vote\_modality**  *(VoteModality)* | Defines ballot formats & JSON parameters. | id PK, name VARCHAR UNIQUE, parameters JSONB |
| **vote\_result**  *(VoteResult)* | Aggregated outcome per target. | id PK, target\_type VARCHAR, target\_id UUID, sum\_weighted\_value NUMERIC, vote\_count INT, **UNIQUE**(target\_type,target\_id) |
| **emerging\_expert**  *(EmergingExpert)* | Flags users with rapid merit growth. | id PK, user\_id FK→auth\_user, detection\_date DATE, score\_delta NUMERIC |
| **integration\_mapping**  *(IntegrationMapping)* | Cross-module glue (e.g., Ethikos topic → vote target). | id PK, module\_name VARCHAR, context\_type VARCHAR, mapping\_details JSONB |

*Index notes* – BTREE (target\_type,target\_id) on vote for fast tallies; partial index on (user\_id) for “my votes”.

## **3 Audit / Ledger Extension *(transparency requirement)***

| **Table** | **Purpose** | **Columns** |
| --- | --- | --- |
| **vote\_ledger** | Immutable log of every ballot & weight hash. | ledger\_id BIGSERIAL PK, vote\_id FK→vote, sha256\_hash BYTEA, block\_height BIGINT, logged\_at TIMESTAMP DEFAULT now() |

## **4 Integrity & Anomaly Logs**

| **Table** | **Purpose** | **Columns** |
| --- | --- | --- |
| **integrity\_event** | Detects collusion, Sybil rings, spam etc. | id PK, user\_id FK→auth\_user NULLABLE, event\_type ekoh\_integrity\_event\_enum, score JSONB, handled BOOLEAN DEFAULT FALSE, created\_at TIMESTAMP DEFAULT now() |

## **5 ENUM Appendix**

*Defined in ekoh\_enums.sql; referenced by table DDL.*

| **Enum type** | **Allowed values** |
| --- | --- |
| ekoh\_privacy\_level\_enum | public, pseudonym, anonymous |
| ekoh\_integrity\_event\_enum | sybil, ring, rapid\_growth, spam |
| vote\_modality\_name\_enum | approval, ranking, rating, preferential, budget\_split |
| stance\_scale\_enum | -3, -2, -1, 0, 1, 2, 3 |

## **6 Retention & Partitioning Policies**

* GDPR-aligned and performance-oriented:  
  + vote & vote\_ledger – hot 5 years, then archived cold; monthly partitions on created\_at.
  + score\_history – 7 year retention; monthly partitions.
  + integrity\_event – retain 3 years rolling; auto-purge handled=true events after 12 months.

## **7 Entity-Relationship Diagram (text)**

text

CopyEdit

auth\_user (PK id)

│1

├─< confidentiality\_setting

├─< user\_ethics\_score

├─< user\_expertise\_score ─┐

│ └─< score\_history

└─< emerging\_expert

expertise\_category

└─< user\_expertise\_score

vote\_modality

└─< vote ──> vote\_result (1:1 per target)

│

└─< vote\_ledger

integration\_mapping

integrity\_event (optional FK to user)

context\_analysis\_log (polymorphic entity\_type/entity\_id)

*A vector version (erd\_ekoh\_smartvote\_v1.svg) lives in docs/assets/.*

## **8 Glossary of Column Abbreviations**

| **Abbrev.** | **Meaning** |
| --- | --- |
| target\_type | Model label of the entity being voted on |
| score\_delta | % or absolute change triggering emerging-expert flag |
| block\_height | Position of ledger entry on external blockchain |
| event\_type | Categorical anomaly detected by Integrity Guard |

### **Change-control Checklist**

1. **Add column/table** → update this file *and* 02-parameter\_reference.md if tunables change.
2. **Rename/drop** → create reversible migration; mark old name here under “Deprecated” until purge.
3. **New ENUM value** → append to §5 and regenerate ERD.

This document is now the *single source of truth* for every table, key, and enum powering the EkoH reputation engine and Smart Vote weighted-voting system. All subsequent specs, code, analytics, and tests must reference these canonical names and structures.

Sources

##### **You said:**

write 2/11

###### **ChatGPT said:**

Thought for 28 seconds

**02-parameter\_reference.md — Definitive Parameter Registry for EkoH & Smart Vote (v1.0)** *This file freezes every knob that integrators, ops teams, and data-scientists may tune.  
 All other documents must hyperlink here rather than restating values.*

## **0 Document structure**

1. Global conventions & units
2. **EkoH engine** parameters
3. **Smart Vote engine** parameters
4. Choice/enum sets (shared)
5. Environment-variable matrix

## **1 Global conventions**

* **Type** — SQL or Python primitive (Decimal(4,3), float, enum, etc.)
* **Range** — inclusive unless otherwise stated.
* **Location** — where code reads the value (ScoreConfiguration table, settings.py, .env).
* **Defaults** — MUST NOT change except by PR that updates **this file first**.

## **2 EkoH engine parameters**

| **Name** | **Location** | **Type & Range** | **Default** | **Description** |
| --- | --- | --- | --- | --- |
| raw\_weight\_quality | ScoreConfiguration | Decimal(4,3) 0 – 5 | **1.000** | Multiplier on *quality* axis in multidimensional scoring |
| raw\_weight\_expertise | ScoreConfiguration | Decimal(4,3) 0 – 5 | **1.500** | Multiplier on *expertise depth* axis |
| raw\_weight\_frequency | ScoreConfiguration | Decimal(4,3) 0 – 5 | **0.750** | Dampening factor for contribution frequency (prevents spam inflation) |
| ethical\_multiplier\_floor | settings.EKOH | float 0 – 1 | **0.20** | Lower bound applied to unethical users; anything below is auto-suspension |
| ethical\_multiplier\_cap | settings.EKOH | float 1 – 2 | **1.50** | Upper bound for exemplary conduct; prevents runaway influence |
| EXPERTISE\_DOMAIN\_CHOICES | ExpertiseCategory fixtures | enum[26] | UNESCO ISCED-F codes | Frozen list used by all domain tagging |

## **3 Smart Vote engine parameters**

| **Name** | **Location** | **Type & Range** | **Default** | **Description** |
| --- | --- | --- | --- | --- |
| VOTE\_MODALITY\_CHOICES | VoteModality | enum | "approval", "ranking", "rating", "preferential" | Allowed ballot types; engine must reject any other string |
| EMERGING\_EXPERT\_THRESHOLD | settings.SMART\_VOTE | % Ekoh Δ over rolling 30 d | **+15 %** | Triggers EmergingExpert flag and mentorship prompts |
| CONSENSUS\_STRONG\_THRESHOLD | settings.SMART\_VOTE | % weighted agreement | **75 %** | Above this, a consultation is labeled “Strong Consensus” in UI |

**Note – soft weight cap**: while the tech spec allows implementation-level caps (e.g. max 10× average weight), no constant is yet frozen; if introduced it must be appended here and cited to the governance minutes.

## **4 Shared choice & enum sets**

| **Enum** | **Values** | **Consumed by** |
| --- | --- | --- |
| ekoh\_privacy\_level\_enum | public, pseudonym, anonymous | ConfidentialitySetting.level |
| ekoh\_integrity\_event\_enum | sybil, ring, rapid\_growth, spam | IntegrityEvent.event\_type |
| vote\_modality\_name\_enum | *(see VOTE\_MODALITY\_CHOICES above)* | VoteModality.name |
| stance\_scale\_enum | -3 … 0 … +3 | Ethikos integration (read-only for Smart Vote) |

All enum literals are **case-sensitive** and must never be localized.

## **5 Environment-variable matrix (excerpt)**

| **Env var** | **Consumes** | **Maps to parameter** | **Default** |
| --- | --- | --- | --- |
| EKOH\_MULTIPLIER\_CAP | settings.EKOH | ethical\_multiplier\_cap | 1.50 |
| SMARTVOTE\_CONSENSUS\_STRONG | settings.SMART\_VOTE | CONSENSUS\_STRONG\_THRESHOLD | 0.75 |
| SMARTVOTE\_EMERGING\_DELTA | settings.SMART\_VOTE | EMERGING\_EXPERT\_THRESHOLD | 0.15 |
| RAW\_WEIGHT\_QUALITY | settings.EKOH | raw\_weight\_quality | 1.000 |
| RAW\_WEIGHT\_EXPERTISE | settings.EKOH | raw\_weight\_expertise | 1.500 |
| RAW\_WEIGHT\_FREQUENCY | settings.EKOH | raw\_weight\_frequency | 0.750 |
| PROMETHEUS\_BASE\_URL | analytics-settings | - | http://prometheus:9090 |

**Convention** – env var names mirror the parameter in UPPER\_SNAKE; values are parsed as float or JSON as appropriate.

PROMETHEUS\_BASE\_URL – analytics-settings CM – http://prometheus:9090

EXPORT\_MAX\_ROWS – analytics-settings CM – 100 000

### **Change-control checklist**

1. **Propose new parameter** → draft PR adding row to correct section + env-var (if needed).
2. **Modify default** → governance vote required; update this file *and* config templates.
3. **Rename / drop** → breaking change; version bump this document and all dependent code.

This registry is the **single source of truth** for runtime tunables in the EkoH reputation engine and Smart Vote weighted-voting system. All services, Helm charts, and CI pipelines must read values through the paths documented above.

Sources

##### **You said:**

3/11

###### **ChatGPT said:**

Thought for 7 seconds

**03-technical\_spec.md — *Layer-by-Layer System Specification for EkoH & Smart Vote (v1.0)*** *(All identifiers must match the names frozen in 01-db\_schema.md and 02-parameter\_reference.md.)*

## **0 Purpose & Scope**

This document translates the high-level architecture into implementable, testable contracts across:

1. **Backend micro-services** (Django + DRF)
2. **Frontend UI & state flow** (React 18 + TanStack Query)
3. **Data / persistence** (PostgreSQL 15, Redis 7, Kafka 3)
4. **DevOps / runtime** (Docker + Kubernetes, Prometheus, Grafana)

The spec is normative; deviations require a version bump and PR that patches every impacted reference doc.

## **1 High-Level Component Diagram**

text

CopyEdit

┌─────────── UI Layer (SPA) ───────────┐

│ /ekoh/\* widgets /smart-vote/\* pages │

└──────────────┬────────────────────────┘

│ HTTPS / JWT

┌───────────────────────▼───────────────────────────┐

│ API Gateway (Konnaxion Core) │

└──┬────────────────────────┬───────────────────────┘

│ │

┌────▼─────┐ ┌─────▼────┐

│ EkohSvc │ gRPC/REST │ VoteSvc │

└────┬─────┘ └────┬─────┘

│ │ Kafka events

│ ▼

┌────▼───────────┐ ┌──────────────┐

│ Auth & Profile │ │ Integrity │

│ Service │ │ Guard │

└────┬───────────┘ └────┬─────────┘

│ │

▼ ▼

┌─────────────────┐ ┌─────────────────┐

│ PostgreSQL 15 │ │ Redis 7 │

│ ekoh & vote DB │ │ cache / locks │

└─────────────────┘ └─────────────────┘

## **2 Backend Micro-Services**

### **2.1 EkohSvc (reputation engine)**

| **Endpoint** | **Method** | **Auth** | **Purpose** | **Status codes** |
| --- | --- | --- | --- | --- |
| /ekoh/profile/:userId | GET | JWT (public if level = public) | Return expertise & ethics scores. | 200, 404 |
| /ekoh/score/recalc | POST | ROLE\_ADMIN | Trigger async full recalculation. | 202 |
| /ekoh/score/bulk | POST | ROLE\_SERVICE | Stream credentials / impacts for batch ingest. | 207, 400 |

*Implementation notes*

* Recalc job enqueues Kafka topic ekoh.score.recalc; Celery worker processes in shards of 500 users.
* Scores stored in user\_expertise\_score & user\_ethics\_score tables .

#### **2.1.1 Algorithm — multidimensional \_scoring**

python

CopyEdit

def weighted\_domain\_score(raw\_quality, raw\_expertise, raw\_freq):

return (

raw\_quality \* RAW\_WEIGHT\_QUALITY

+ raw\_expertise \* RAW\_WEIGHT\_EXPERTISE

+ raw\_freq \* RAW\_WEIGHT\_FREQUENCY

)

Constants map to parameters frozen in §2 of the registry .

### **2.2 VoteSvc (weighted voting)**

| **Endpoint** | **Method** | **Auth** | **Purpose** | **Status** |
| --- | --- | --- | --- | --- |
| /smart-vote/cast | POST | JWT | Submit ballot {target, modality, value}. | 201, 400, 409 |
| /smart-vote/result/:targetType/:id | GET | JWT/Anon | Retrieve current weighted result snapshot. | 200, 404 |
| /smart-vote/modality | GET | JWT | List supported modalities. | 200 |

#### **2.2.1 Weight-calculator pseudocode**

python

CopyEdit

def compute\_weight(user\_id, consultation):

base = 1

dot\_product = sum(

user\_scores[d] \* consultation.relevance[d]

for d in consultation.relevance

)

weight = (base + dot\_product) \* user\_ethics[user\_id]

return min(weight, HARD\_CAP) # optional cap

user\_scores from user\_expertise\_score, ethics multiplier from user\_ethics\_score.  
 HARD\_CAP to be introduced only after governance vote (not yet in registry).

#### **2.2.2 VoteEngine flow (sequence)**

1. Frontend POST /cast → API gateway → VoteSvc.
2. VoteSvc checks idempotency (vote table UNIQUE(user,target)), else 409.
3. Calls **WeightCalculator** → returns W.
4. Stores raw\_value, weighted\_value = raw\_value \* W in vote.
5. Emits Kafka vote.cast event; **Aggregator** service consumes, UPSERTs into vote\_result.
6. Writes hash to vote\_ledger + (optionally) layer-1 blockchain.

Error codes map to standard JSON problem+ details (code, detail, instance).

## **3 Frontend (SPA)**

### **3.1 Route map (superset of Navigation Map entries)**

| **Path** | **Component** | **Auth** | **Data source** |
| --- | --- | --- | --- |
| /ekoh/profile/:uid | <EkohProfile> | Public/Private | /ekoh/profile/:uid |
| /smart-vote/:entity/:id | <ConsultationPage> | JWT/Anon | /smart-vote/result + /cast |
| /reports/smart-vote | <SmartVoteDashboard> | ROLE\_ADMIN | /reports/smart-vote (Insights API) |

*State management* — TanStack Query caches per-route; WebSocket channel ws://.../smart-vote/live/:id pushes partial tally updates every 5 s.

## **4 Data Layer Contracts**

| **Table** | **Access pattern** | **Service owner** |
| --- | --- | --- |
| user\_expertise\_score | R/W by EkohSvc; read-only by VoteSvc. |  |
| vote | Insert by VoteSvc; read by Aggregator. |  |
| vote\_result | Upsert by Aggregator; read by VoteSvc & UI. |  |
| vote\_ledger | Append-only by VoteSvc; read by Audit Lambda. |  |

All table names & columns must match 01-db\_schema.md.

## **5 Integration & Messaging**

| **Kafka topic** | **Producer** | **Consumer** | **Payload schema** |
| --- | --- | --- | --- |
| ekoh.score.recalc | API Gateway | Celery worker | {batch\_id, user\_ids[]} |
| vote.cast | VoteSvc | Aggregator, IntegrityGuard | {vote\_id, user\_id, target, w\_value} |
| integrity.alert | IntegrityGuard | Ops Slack hook | {event\_id, type, user\_id?} |

Topic names are frozen; message payloads versioned (v1).

## **6 Security & Auth**

* JWT issued by Konnaxion Auth (HS256 + rotating keys).
* Role matrix: ROLE\_USER, ROLE\_EXPERT, ROLE\_ADMIN.
* Vote endpoints accept anonymous read but require JWT for cast.
* EKOH profile obeys confidentiality\_setting.level (401 vs 403 vs redacted fields).
* Rate-limit: 60 req/min/user via API Gateway Leaky-Bucket filter.

## **7 Error & Event Codes (excerpt)**

| **Code** | **HTTP** | **Message** | **Source** |
| --- | --- | --- | --- |
| EKOH\_40401 | 404 | “User not found” | EkohSvc |
| VOTE\_40901 | 409 | “Duplicate ballot” | VoteSvc |
| INTG\_20001 | 202 | “Recalc queued” | EkohSvc |
| AUTH\_40101 | 401 | “JWT expired” | Gateway |

Full list lives in /common/error\_codes.yaml.

## **8 DevOps / Runtime**

* **Containers** – each service in Alpine-based image; tag v1.0.{git-sha}.
* **Kubernetes** – namespace konnaxion-intel; HPA 2-10 replicas per service on CPU 70 %.
* **Observability** – Prometheus / Grafana dashboards:  
  + vote\_tps (target ≥ 200/s)
  + ekoh\_recalc\_duration\_ms (p95 < 120 000)
* **CI pipeline** – GitHub Actions → pytest + doc-lint + vulnerability scan; block merge if coverage < 90 %.

## **9 Non-Functional Requirements**

| **Category** | **Target** |
| --- | --- |
| **Latency** | P95 vote cast < 150 ms (same region) |
| **Throughput** | 100 k ballots / min burst |
| **Uptime** | 99.9 % per rolling 30 d |
| **Data integrity** | Zero lost ledger entries; SHA-256 mismatch alert in < 1 min |
| **Scalability** | Horizontal; DB partitioning strategy defined in 01-db\_schema.md §6 |
| **Compliance** | GDPR, CCPA; ISO 27001 controls inherited from Konnaxion core |

### **Change-Control Checklist**

1. Add/modify endpoint → update §2 tables + Integration Mapping doc.
2. Change message schema → bump topic payload version & reflect here.
3. New role/permission → update Role matrix in §6 and Parameter Registry.

This technical specification is now the contract binding code, tests, and operations for the EkoH reputation engine and Smart Vote weighted-voting subsystem. All engineering work must conform to the interfaces, flows, and thresholds defined herein.

##### **You said:**

4/11

###### **ChatGPT said:**

Thought for 15 seconds

**04-inventory\_of\_features.md — Feature Catalogue for EkoH & Smart Vote (v1.0)** *Each row is a “contract”; code, DB, tasks and UI must implement the exact code-name.*

## **0 How to read this file**

* **Display Name → Code Name** — UI wording on the left, immutable identifier on the right.
* **Layer tags** —  
  + **Svc** = backend service / function
  + **API** = REST/gRPC endpoint
  + **Task** = Celery / Cron job
  + **DB** = table(s) updated
  + **UI** = React component / page hook
* **Depends on** — tables, parameters, or other features that must exist first.
* All tables & params referenced are frozen in 01-db\_schema.md and 02-parameter\_reference.md.

## **1 EkoH engine (Kollective Intelligence)**

| **#** | **Display Name → Code Name** | **Purpose / behaviour** | **Layer tags** | **Depends on** |
| --- | --- | --- | --- | --- |
| 1 | Multidimensional Scoring → **multidimensional\_scoring** | Computes per-user merit along axes (quality, frequency, expertise) | Svc services/multidimensional\_scoring.py  Task ekoh\_score\_recalc | DB: user\_expertise\_score  Params: raw\_weight\_\* |
| 2 | Criteria Customization → **configuration\_weights** | Admins adjust axis weights at runtime | API PATCH /admin/ekoh/weights  DB score\_configuration | Param registry |
| 3 | Automatic Contextual Analysis → **contextual\_analysis** | AI fine-tunes scores by topic & complexity | Svc services/contextual\_analysis.py  Task contextual\_analysis\_batch | DB context\_analysis\_log |
| 4 | Dynamic Privacy → **privacy\_settings** | Apply anonymity / pseudonym while still showing scores | API PUT /ekoh/privacy  UI toggle  DB confidentiality\_setting | Enum ekoh\_privacy\_level\_enum |
| 5 | History & Traceability → **score\_history** | Persist every score recalculation for audit | Svc logic wrapper  DB score\_history | Audit policy |
| 6 | Interactive Visualizations → **score\_visualization** | Serve live dashboards & skill-maps | API /reports/ekoh  UI <SkillMap> | Materialised view mat\_expert\_domain\_top10 |
| 7 | Expertise Classification by Field → **expertise\_field\_classification** | Bind each sub-score to formal UNESCO domain | Svc helper  DB expertise\_category | Fixtures EXPERTISE\_DOMAIN\_CHOICES |

## **2 Smart Vote engine**

| **#** | **Display Name → Code Name** | **Purpose / behaviour** | **Layer tags** | **Depends on** |
| --- | --- | --- | --- | --- |
| 1 | Dynamic Weighted Voting → **dynamic\_weighted\_vote** | Re-weights every ballot using voter’s EkoH score | Svc services/weight\_calculator.py  API POST /smart-vote/cast | DB vote, user\_expertise\_score, user\_ethics\_score |
| 2 | Flexible Voting Modalities → **voting\_modalities** | Support approval, ranking, rating, preferential ballots | DB seed vote\_modality  UI <BallotModal> | Enum vote\_modality\_name\_enum |
| 3 | Emerging Expert Detection → **emerging\_expert\_detection** | Flag users with sharp EkoH growth | Task detect\_emerging\_expert  DB emerging\_expert | Param EMERGING\_EXPERT\_THRESHOLD |
| 4 | Transparency of Results → **vote\_transparency** | Publish raw + weighted values & context (no PII) | API GET /smart-vote/result  DB vote\_ledger | Privacy policy |
| 5 | Advanced Result Visualizations → **vote\_result\_visualization** | Histograms, network graphs, interactive maps | UI <VoteGraphs>  Query vote\_result | Reporting conduit |
| 6 | Cross-Module Integration → **cross\_module\_vote\_integration** | Make Smart Vote callable from all modules | API gateway mapping  DB integration\_mapping | Auth scopes, event bus |

## **3 Feature → Dependency Matrix (quick lookup)**

| **Code Name** | **Tables** | **Params** | **Scheduled Task** | **UI Hook** |
| --- | --- | --- | --- | --- |
| multidimensional\_scoring | user\_expertise\_score | raw\_weight\_\* | ekoh\_score\_recalc | n/a |
| contextual\_analysis | context\_analysis\_log | — | contextual\_analysis\_batch | n/a |
| privacy\_settings | confidentiality\_setting | — | — | <PrivacyToggle> |
| dynamic\_weighted\_vote | vote, user\_expertise\_score, user\_ethics\_score | — | — | <BallotForm> |
| vote\_transparency | vote\_ledger, vote\_result | — | — | <VoteLedgerTable> |
| emerging\_expert\_detection | emerging\_expert | EMERGING\_EXPERT\_THRESHOLD | nightly Celery | <MentorBadge> |

*(If a square is blank, that layer does not touch the feature.)*

### **Change-control reminders**

1. **Add a new feature** → register Display Name → Code Name here **first**.
2. **Rename a feature** → migrate code, DB seed, tasks, and update this catalogue in the same PR.
3. **Delete a feature** → deprecate here, ensure no DB FK or API route still references the code-name, then remove.

This catalogue is the authoritative cross-reference between UX wording, backend services, scheduled tasks, database artefacts, and configuration parameters for EkoH and Smart Vote. All future specs, code and documentation must use these exact identifiers to stay coherent.

Sources

##### **You said:**

5/11

###### **ChatGPT said:**

Thought for 25 seconds

**05-reporting\_analytics.md — Reporting & Analytics Slice for EkoH + Smart Vote (layer 5)** *(Frontend dashboards → read-only API → star-schema warehouse → ETL & DevOps)*

## **0 Scope & Position in the Stack**

The Insights (Reporting & Analytics) module sits **read-only** atop the operational Konnaxion databases.  
 It provides:

* React dashboards under /reports/\* (5.1)
* reports-api micro-service (5.2)
* A dedicated PostgreSQL 16 **star-schema** cluster + Redis cache (5.3)
* Airflow-driven ETL + monitoring/backup infrastructure (5.4)

These four sub-layers map 1-to-1 to the headings of *Document 5* referenced below.

## **1 Frontend Dashboards (5.1)**

| **Route** | **React page** | **Key components** |
| --- | --- | --- |
| /reports/smart-vote | SmartVoteDashboard.tsx | SmartVoteChart, TimeRangePicker |
| /reports/usage | UsageDashboard.tsx | UsageBigNumbers, DomainHeatMap |
| /reports/perf | PerfDashboard.tsx | LatencySLOGauge, ErrorRateSparkline |

Component and file naming must follow the package layout shown in the design spec.

All charts use Chart.js 4, Ant-Design 5 widgets, and React-Query for data fetch / cache (no extra state library).

## **2 Backend Service – reports-api (5.2)**

| **Endpoint** | **Method** | **Filters** | **Cache-TTL** | **Notes** |
| --- | --- | --- | --- | --- |
| /reports/smart-vote | GET | range (24h/7d/30d/custom), grouping (day/week) | 600 s | Returns {labels[], votes[], avg\_score[]} |
| /reports/usage | GET | identical | 600 s | MAU, docs, projects |
| /reports/perf | GET | range, endpoint | 300 s | P95 latency & error-rate |
| /reports/export | GET | report, format | 60 s | CSV/JSON stream, ADMIN-only |

*Service tech-stack*: Django 4.2 + DRF 3.15, Redis-cached, JWT-protected, 60 req/min throttle.

## **3 Star-Schema Warehouse (5.3)**

### **3.1 Overview**

Fact tables **smart\_vote\_fact**, **usage\_mau\_fact**, **api\_perf\_fact** sit at the centre, linked to **dim\_date**, **dim\_domain**, **dim\_endpoint**.

### **3.2 Core dimensions**

| **Dim** | **PK** | **Note** |
| --- | --- | --- |
| dim\_date | date\_id | pre-populated 2000-2035 |
| dim\_domain | domain\_id | maps UNESCO codes |
| dim\_endpoint | endpoint\_id | REST path catalogue |

All surrogate keys = SMALLINT, zstd-compressed.

### **3.3 Fact smart\_vote\_fact (partitioned monthly)**

*Columns*: id UUID, date\_id, domain\_id, question\_id, user\_id (SHA-256), vote\_value, score\_normalised  
 *Indexes*: (domain\_id, date\_id) BRIN.  
 *ETL*: **etl\_smart\_vote** writes delta every 10 min; nightly full load.

### **3.4 Materialised views**

| **View** | **Scope** | **Refresh** |
| --- | --- | --- |
| vw\_smart\_vote\_30d | last 30 days aggregates | incremental via ETL trigger |

All views are created WITH NO DATA during migration; first populate via Airflow.

### **3.5 Retention & purge**

* smart\_vote\_fact — keep 5 yrs, drop oldest partition yearly.  
   Hashes replace user IDs; cohorts < 10 rows are discarded for k-anonymity.

## **4 ETL & Airflow (5.4)**

| **DAG id** | **Schedule** | **Task summary** |
| --- | --- | --- |
| etl\_smart\_vote | \*/10 \* \* \* \* | ingest weighted votes → smart\_vote\_fact |
| etl\_usage | 0 \* \* \* \* | hourly MAU update → usage\_mau\_fact |
| etl\_perf | \*/15 \* \* \* \* | pull Prometheus metrics → api\_perf\_fact |
| refresh\_mat\_views | 5 \* \* \* \* | REFRESH MATERIALIZED VIEW vw\_\* |
| cleanup\_cache | @hourly | purge Redis keys > 12 h |
| purge\_old\_partitions | 0 4 \* \* 0 | drop partitions past retention |

Airflow images & resources are detailed in the infra table.

## **5 DevOps & Runtime**

| **Resource** | **Image** | **Replicas (HPA)** | **CPU m (req/lim)** | **RAM Mi (req/lim)** |
| --- | --- | --- | --- | --- |
| **reports-api** | ghcr.io/konnaxion/reports-api:<sha> | 3 (2 → 6) | 300 / 600 | 384 / 768 |
| **reports-etl-worker** | ghcr.io/konnaxion/reports-etl:<sha> | 2 (manual) | 400 / 800 | 512 / 1024 |

Environment variables & secrets (Vault / ConfigMap) are listed in the infra section.

### **Monitoring & alerts**

Prometheus rules e.g. reports\_latency\_p95 (> 400 ms) and etl\_task\_fail (Airflow) drive PagerDuty.

Performance baseline: 100 M-row partition attach/detach ≤ 2 s; /reports/\* P95 ≤ 400 ms.

Backups: pgBackRest nightly + 30-min incrementals; 14-day retention; quarterly restore drill.

## **6 Security & Access**

* All user\_id fields are irreversible SHA-256 hashes.
* Role **reports\_reader** = SELECT on dims/views only; facts restricted to ETL account.
* CSV export capped at **100 000 rows** and ADMIN-only.

## **7 CI/CD Extensions**

Jobs schema-diff, dag-lint, perf-budget, export-guard enforce schema drift, DAG quality, latency budget, and export limits before deploying images via Argo CD.

## **8 Change-Control Checklist**

1. **Add column to fact/dim** → new migration + update §3 tables + schema-diff test.
2. **New dashboard** → register React page & add OpenAPI path in §2.
3. **Modify DAG schedule** → edit table in §4 + redeploy Airflow chart.
4. **New Prometheus alert** → append to §5 metrics list and HPA triggers.

This document now completes layer 5, binding dashboards, APIs, warehouse schema, ETL jobs and DevOps contracts into a single authoritative reference for all Reporting & Analytics functions in the EkoH / Smart Vote ecosystem.

Sources

##### **You said:**

6/11

###### **ChatGPT said:**

Thought for 5 seconds

**06-insights\_module.md — Analytics & ML Reference for EkoH + Smart Vote (v 1.0)** *This file freezes every dataset, feature, and configuration knob consumed by downstream statistical dashboards, machine-learning models, and insight services. Anything that feeds or reads the star-schema warehouse (§3 of the Reporting doc) must respect these contracts.*

## **0 Scope**

* **In-scope** – feature dictionaries, predictive-bonus rules, drift-detection thresholds, Airflow DAG IDs, environment variables, and privacy guarantees that govern the “Insights” slice.
* **Out-of-scope** – operational reporting queries (see 05-reporting\_analytics.md), API routes (03-technical\_spec.md), or raw OLTP schema (01-db\_schema.md).

## **1 Canonical Data Inputs**

| **Source table / view** | **Grain** | **Primary keys** |
| --- | --- | --- |
| smart\_vote\_fact | *vote × day × domain* | id (UUID) |
| vw\_smart\_vote\_30d | *aggregated view* – last 30 d | (date\_id, domain\_id) |
| usage\_mau\_fact | *org × month* | (org\_id, month) |
| api\_perf\_fact | *endpoint × 5 min* | (endpoint\_id, ts\_5m) |

All user identifiers are one-way SHA-256 hashes with secret salt, and cohorts smaller than **10** rows are discarded to preserve k-anonymity .

## **2 Feature Dictionary**

| **Feature group** | **Feature name (snake\_case)** | **Definition / transformation** |
| --- | --- | --- |
| **Vote metrics** | weighted\_yes\_ratio | sum(weighted\_yes) / sum(weighted\_total) over 30 d |
|  | expert\_participation\_rate | voters with EkoH score ≥ 80 ÷ total voters |
|  | weight\_gini\_index | inequality of weights in a consultation |
| **User trajectory** | ekoh\_score\_delta\_30d | latest EkoH – EkoH 30 d ago |
|  | predictive\_bonus\_points | accumulated per §4 rule |
| **System health** | api\_p95\_latency\_ms | P95 latency per endpoint (5 min bins) |
|  | etl\_dag\_success\_rate | successful Airflow task instances / total |

Surrogate keys map to dim\_date, dim\_domain, dim\_endpoint (see star schema). All features are materialised in nightly Airflow task **etl\_feature\_store** (DAG spec below).

## **3 Predictive-Insight Bonus Logic (EkoH)**

1. Identify minority votes (< 25 % weighted share at close).
2. After a **validation event** (expert panel decision or real-world KPI) proves the minority correct, award **+b** bonus points:  
    b=2×Wu,c b = 2 \times \sqrt{W\_{u,c}}b=2×Wu,c​​
3. Persist to predictive\_bonus\_points and score\_history.
4. Cap cumulative bonus at **+10 %** of user’s current domain score to prevent gaming.

Design rationale documented in the EkoH overview .

## **4 Model Training & Drift Detection**

| **Model ID** | **Target** | **Algorithm** | **Retrain cadence** | **Drift signal** |
| --- | --- | --- | --- | --- |
| sv\_outcome\_regressor\_v1 | Consultation outcome within ±5 % | Gradient-boosted trees | weekly | KS-stat on weighted\_yes\_ratio vs training |
| ekoh\_growth\_forecast\_v1 | ekoh\_score\_delta\_30d | Prophet | daily | MAPE > 15 % on latest week |
| expert\_emergence\_classifier\_v1 | Binary flag (emerging\_expert) | Logistic + SMOTE | monthly | F1 drop > 5 % |

Drift thresholds and retrain timers are defined in Airflow DAG defaults (section 5).

## **5 Airflow DAG Registry (extracted from *Insights Parameter Reference*)**

yaml

CopyEdit

etl\_dags: # frozen list (do not rename ids)

- id: etl\_smart\_vote # every 10 min

- id: etl\_usage # hourly

- id: etl\_perf # every 15 min

- id: refresh\_mat\_views # hourly :05

- id: cleanup\_cache # hourly

- id: purge\_old\_partitions # Sundays 04:00 UTC

- id: etl\_feature\_store # nightly 01:00 UTC (new)

- id: retrain\_models # varies (see model spec)

Schedules and tasks for the first six DAGs are fixed per document lines 16-51 and 41-48 .

## **6 Configuration Parameters (superset of 02-parameter\_reference.md)**

| **Env var** | **Type** | **Default** | **Notes** |
| --- | --- | --- | --- |
| REPORTS\_DB\_URL | string | *(none)* | Vault; Analytics Postgres conn |
| REDIS\_URL | string | redis://redis:6379/0 | Shared Redis cache |
| EXPORT\_MAX\_ROWS | int | **100 000** | Hard cap enforced in reports-api |
| PREDICTIVE\_BONUS\_CAP | float | **0.10** | 10 % ceiling of domain score |
| MODEL\_DRIFT\_KS\_THRESHOLD | float | **0.07** | KS-stat threshold before retrain |

All new variables follow Cookiecutter-Django naming, and are injected via ConfigMap or Vault. No overlap with global matrix (§7 of Definitive Parameter Reference) .

## **7 Resource & Autoscaling Defaults**

* **reports-api**: 3 replicas, HPA 2→6, CPU 300 m / 600 m, RAM 384 Mi / 768 Mi
* **reports-etl-worker**: 2 fixed pods, 400 m / 800 m, 512 Mi / 1 Gi
* **Airflow**: 1 scheduler, 2 workers, 250 m CPU each

These values must remain in sync with the Helm chart in 09-config\_deploy.md.

## **8 Privacy & Governance Guarantees**

* SHA-256 hashing of user\_id; no raw PII in analytics DB
* Cohort suppression for counts < 10 rows
* Read-only role **reports\_reader** restricted to dims/views; raw facts accessible only by ETL account
* Full API request audit retained 6 months in OpenSearch

### **Change-Control Checklist**

1. **Add new feature column** → update §1 table list + §2 feature dictionary + regenerate etl\_feature\_store.
2. **New DAG** → append YAML in §5 and add schedule to 08-tasks\_schedule.md.
3. **Tweak resource limits** → sync with 09-config\_deploy.md and re-cite lines in §7.
4. **Alter privacy rules** → governance approval + update §8 and the Parameter Reference.

This reference ensures that data scientists, ML engineers, and BI developers work from a single, immutable contract covering data structures, feature semantics, schedules, privacy, and runtime limits for the Insights stack powering EkoH and Smart Vote.

Sources

##### **You said:**

7/11

###### **ChatGPT said:**

**07-integration\_mapping.md — Cross-Module Route, Event & Auth Map (v 1.0)** *This document is the “wiring diagram” that binds EkoH / Smart Vote to the rest of the Konnaxion platform. Any change here forces a parallel patch in the Technical Spec, DevOps manifests, and Parameter Reference.*

## **0 Reference sources**

* **Navigation Map** – canonical list of top-level routes
* **Definitive Parameter Reference** – route invariants & env-var prefixes
* **Technical Specification v14** – gateway, namespaces, auth roles
* **Docs 01-05** in this series – DB tables, features, analytics.

## **1 REST / gRPC Route Matrix**

/reports GET reports-api read:reports Analytics hub landing

| **Path** | **Method** | **Owner svc** | **Auth scope** | **Purpose** |
| --- | --- | --- | --- | --- |
| /ekoh/profile/:uid | GET | **EkohSvc** | read:profile (public if level=public) | Fetch expertise & ethics profile |
| /ekoh/score/recalc | POST | **EkohSvc** | admin:ekoh | Queue async full recalculation |
| /smart-vote/cast | POST | **VoteSvc** | write:vote | Submit weighted ballot |
| /smart-vote/result/:type/:id | GET | **VoteSvc** | none (public) | Retrieve current weighted result |
| /reports/smart-vote | GET | **reports-api** | read:reports | Time-series JSON for dashboards |
| /reports/export | GET | **reports-api** | admin:reports | CSV/JSON bulk export (≤100 k rows) |

*All new routes inherit API-gateway prefix /api/v1/ and version header X-Konnaxion-API: 1.  
 Path strings* ***must exactly match*** *Navigation Map slots to avoid regressions.*

## **2 Event-Bus (Kafka 3) Topics**

ws.smart-vote.\* aliases /ws/reports/\* until v15 migration

| **Topic** | **Producer** | **Consumer(s)** | **Schema ID (Avro)** | **Notes** |
| --- | --- | --- | --- | --- |
| ekoh.score.recalc | API-Gateway | Celery shard\*\* | ekoh\_recalc\_v1 | Batch of user IDs |
| vote.cast | **VoteSvc** | Aggregator, IntegrityGuard | vote\_cast\_v1 | Weight & hash |
| integrity.alert | IntegrityGuard | Ops-Slack-Hook | integrity\_alert\_v1 | Collusion / Sybil |
| analytics.warehouse.load | Airflow | Warehouse PG | etl\_load\_v1 | Fact inserts |
| ws.smart-vote.\* | - | - | - | WebSocket alias for /ws/reports/\* |

Topic names & schemas are immutable once released; new fields → new schema version with \_v2 suffix.

## **3 Frontend Route → Component Map**

| **SPA path** | **React component** | **Data source / hook** |
| --- | --- | --- |
| /ekoh/profile/:uid | <EkohProfile> | useQuery('/ekoh/profile/:uid') |
| /smart-vote/:entity/:id | <ConsultationPage> | useQuery('/smart-vote/result/...') + <BallotForm> |
| /reports/smart-vote | <SmartVoteDashboard> | useQuery('/reports/smart-vote') |

Component filenames must align with UI conventions in the v14 spec.

## **4 Role & Permission Matrix**

| **Role** | **OAuth2 scope(s)** | **Capabilities** |
| --- | --- | --- |
| **ROLE\_USER** | read:profile, write:vote | View any public profile, cast votes |
| **ROLE\_EXPERT** | *User* + read:reports | Access aggregate dashboards |
| **ROLE\_ADMIN** | *Expert* + admin:ekoh, admin:reports | Trigger score recalcs, exports, view integrity alerts |

Scopes are issued by Auth Svc JWT and enforced by API-gateway filter inline with the Parameter Reference.

## **5 Integration Mapping Table (integration\_mapping)**

| **Field** | **Meaning** | **Example** |
| --- | --- | --- |
| module\_name | Source module requesting a Smart-Vote (e.g. Ethikos) | "Ethikos" |
| context\_type | Object type in that module | "discussion\_thread" |
| mapping\_details | JSON: { "external\_id": "abc", "vote\_target\_id": "uuid-123" } | – |

*Rule*: external module **never** stores raw vote IDs; must insert a mapping row then call /smart-vote/cast with the returned vote\_target\_id.

## **6 Webhooks / External Bridges**

| **Direction** | **URL / Target** | **Event** | **Payload** |
| --- | --- | --- | --- |
| Outbound | **Ops Slack** (https://hooks.slack.com/...) | integrity.alert | { event\_id, type, user\_id?, severity } |
| Outbound | Layer-1 blockchain RPC | vote\_ledger batch | [ { sha256, block\_height }... ] |
| Inbound | Credentials Registry webhook | new credential | { user, credential\_type, domain } → EkohSvc /score/bulk |

All webhook secrets in Vault; L1 chain configurable via LEDGER\_RPC\_URL.

## **7 Environment-Variable Alignment**

| **Env var** | **Consumed by** | **Mirrors parameter** |
| --- | --- | --- |
| EKOH\_MULTIPLIER\_CAP | EkohSvc | ethical\_multiplier\_cap |
| SMARTVOTE\_CONSENSUS\_STRONG | VoteSvc | CONSENSUS\_STRONG\_THRESHOLD |
| REPORTS\_DB\_URL | reports-api, Airflow | warehouse Postgres DSN |

Must match both 02-parameter\_reference.md and Helm chart values.

## **8 Sequence Diagram (cast vote)**

mermaid

CopyEdit

sequenceDiagram

participant UI

participant Gateway

participant VoteSvc

participant EkohSvc

participant DB as Postgres

participant Kafka

UI->>Gateway: POST /smart-vote/cast

Gateway->>VoteSvc: JWT + payload

VoteSvc->>EkohSvc: gRPC getWeight(user, target)

EkohSvc->>DB: SELECT score, ethics

EkohSvc-->>VoteSvc: weight W

VoteSvc->>DB: INSERT vote, vote\_ledger

VoteSvc->>Kafka: produce vote.cast

VoteSvc-->>Gateway: 201 Created

Gateway-->>UI: 201

## **9 Change-Control Checklist**

1. **Add route** → update §1 + Role matrix, cite Navigation Map.
2. **New Kafka topic** → append §2 + Avro schema registry + Helm propagation.
3. **Modify auth scope** → revise §4 and sync with Keycloak realm config.
4. **Alter IntegrationMapping** → create reversible DB migration + update §5.

Once merged, this mapping becomes the authoritative bridge spec; all services, front-end routes, and external modules must comply verbatim. Any divergence requires a version bump across this doc and affected layers.

##### **You said:**

8/11

###### **ChatGPT said:**

Thought for 9 seconds

**08-tasks\_schedule.md — Central Job-&-Scheduler Registry for EkoH + Smart Vote (v 1.0)** *All recurring jobs—whether Celery beat, Airflow DAG, CronJob, or Kubernetes Cron—are declared here.  
 If it’s not on this page, it should not run in prod.*

## **0 Scheduler Landscape**

| **Layer** | **Engine** | **Clock source** | **Time-zone** | **Container image** |
| --- | --- | --- | --- | --- |
| **Application jobs** | Celery Beat 5.4 | K8s time | UTC | ghcr.io/konnaxion/app-worker:<sha> |
| **Analytics / ETL** | Airflow 2.9 (K8sExecutor) | Airflow DB | UTC | ghcr.io/konnaxion/reports-etl:<sha> |
| **K8s Infra** | k8s CronJob | cluster | UTC | bitnami/kubectl:latest |

All schedules below are expressed in **cron-syntax UTC**.

## **1 Celery Periodic Tasks  *(operational EkoH / Smart Vote)***

| **Task ID (code-name)** | **Queue** | **Schedule** | **SLA** | **Purpose** |
| --- | --- | --- | --- | --- |
| ekoh\_score\_recalc | ekoh | 0 2 \* \* \* (02:00 UTC daily) | 2 h | Recompute all user\_expertise\_score & user\_ethics\_score |
| contextual\_analysis\_batch | ekoh | \*/30 \* \* \* \* (every 30 min) | 15 min | Apply AI context adjustments; log to context\_analysis\_log |
| detect\_emerging\_expert | ekoh | 30 2 \* \* \* | 30 min | Flag users meeting EMERGING\_EXPERT\_THRESHOLD |
| vote\_result\_aggregator | vote | \*/5 \* \* \* \* (every 5 min) | 2 min | Consume vote.cast topic; upsert vote\_result |
| integrity\_guard\_scan | vote | \*/30 \* \* \* \* | 10 min | Scan for Sybil / ring patterns; emit integrity.alert |
| ledger\_batch\_to\_chain | vote | 0 \* \* \* \* (top of hour) | 15 min | Anchor vote\_ledger hashes on L1 blockchain |

**Concurrency**: each queue runs with --concurrency=4; Beat sends max 1 task instance per schedule.

## **2 Airflow DAG Schedule  *(analytics warehouse)***

| **DAG ID** | **Schedule** | **Owner** | **Success SLA** | **Notes** |
| --- | --- | --- | --- | --- |
| etl\_smart\_vote | \*/10 \* \* \* \* | data-eng | 10 min | Incremental load → smart\_vote\_fact |
| etl\_usage | 0 \* \* \* \* | data-eng | 30 min | Update usage\_mau\_fact |
| etl\_perf | \*/15 \* \* \* \* | sre-team | 20 min | Pull Prometheus → api\_perf\_fact |
| refresh\_mat\_views | 5 \* \* \* \* | data-eng | 10 min | Refresh vw\_\* materialised views |
| etl\_feature\_store | 0 1 \* \* \* | ml-ops | 1 h | Materialise nightly feature dataset |
| retrain\_models | 30 1 \* \* 1 (Mon) | ml-ops | 3 h | Conditional; skips if drift < threshold |
| cleanup\_cache | @hourly | sre-team | 10 min | Purge Redis keys > 12 h |
| purge\_old\_partitions | 0 4 \* \* 0 (Sun) | dba | 2 h | Drop facts older than retention window |

**DAG fail alerts** → Airflow’s on\_failure\_callback → integrity.alert Slack webhook.

## **3 Kubernetes CronJobs  *(infrastructure / housekeeping)***

| **CronJob name** | **Schedule** | **Image tag** | **Command** | **Retention** |
| --- | --- | --- | --- | --- |
| pgbackrest-full | 0 0 \* \* \* | pgbackrest:2.48 | full backup to S3 | 14 days |
| log-rotate | 15 \*/6 \* \* \* | bitnami/kubectl | rotate & compress container logs | 7 days |
| prometheus-snapshot-export | 45 0 \* \* \* | prom/prometheus | snapshot + push to long-term bucket | 30 days |

## **4 Alerting & Monitoring Rules**

| **Metric** | **Threshold** | **Alert channel** | **Owner** |
| --- | --- | --- | --- |
| task\_duration{task='ekoh\_score\_recalc',p95} | > 2 h | PagerDuty SRE-P1 | sre-team |
| dag\_failure\_rate{dag\_id='etl\_smart\_vote'} | > 5 % over 1 h | #data-alerts | data-eng |
| integrity\_event\_unhandled\_total | > 50 | #moderation | trust-ops |
| vote\_queue\_lag\_seconds | > 300 s | PagerDuty SRE-P2 | sre-team |

Alerts configured in PrometheusRule CRDs; escalation policies in Ops runbook.

## **5 Environment Variables (Scheduler)**

| **Variable** | **Default** | **Consumed by** | **Description** |
| --- | --- | --- | --- |
| CELERY\_BEAT\_SCHEDULE\_TZ | UTC | all app-workers | Clock TZ |
| AIRFLOW\_\_CORE\_\_DEFAULT\_TIMEZONE | UTC | Airflow | Global TZ |
| EKOH\_TASK\_CONCURRENCY | 4 | Celery | Overrides worker concurrency |
| EXPORT\_MAX\_ROWS | 100000 | reports-api | Enforced by DAG etl\_export\_guard |

Values must mirror Parameter Registry (§5 of 02-parameter\_reference.md) and Helm overrides.

## **6 Change-Control Checklist**

1. **Add / modify Celery task**
   * Patch code; register in CeleryBeat config; append row here (Section 1).
2. **Add / modify Airflow DAG**
   * PR must include DAG file, YAML schedule in 06-insights\_module.md, and row in Section 2.
3. **Alter SLA or alert threshold**
   * Update Section 4, plus PrometheusRule manifest, then notify SRE slack.
4. **Rename CronJob**
   * Change K8s manifest + Section 3 table; preserve old job logs 7 days.

This schedule is **source-of-truth** for all automated jobs in EkoH & Smart Vote. CI will fail if a task in code isn’t listed here, or a cron expression here is missing from manifests.

Sources

##### **You said:**

9/11

###### **ChatGPT said:**

Thought for 11 seconds

**09-config\_deploy.md — Deployment & Configuration Guide for EkoH + Smart Vote (v 1.0)** *(This is the single source of truth for container images, Helm values, env vars, secrets, resource sizing, and CI/CD flow. Any change here* ***must*** *be reflected in the codebase, GitHub Actions, and Argo CD manifests before it can reach production.)*

## **0 Scope & Philosophy**

* **Immutable images** – every build is tagged v<major>.<minor>.<git-sha> and never overwritten.
* **Config-as-code** – *all* Kubernetes, Helm, and Argo CD objects live in Git. Runtime changes are carried by a manifest patch, never by kubectl exec.
* **Twelve-factor env-vars** – no secrets or parameters hard-coded in images; everything injectable via ConfigMap or Vault-backed Secret.
* **One namespace per logical stack** – konnaxion-intel for EkoH & Smart Vote, konnaxion-analytics for Reporting/Insights.

## **1 Container Images & Tagging**

| **Service / Job** | **Image repository** | **Tag pattern** | **Build trigger** |
| --- | --- | --- | --- |
| **EkohSvc** | ghcr.io/konnaxion/ekoh-svc | v1.0.<sha> | Merge to main in services/ekoh |
| **VoteSvc** | ghcr.io/konnaxion/vote-svc | v1.0.<sha> | Merge to main in services/vote |
| **reports-api** | ghcr.io/konnaxion/reports-api | v1.0.<sha> | Merge to main in services/reports |
| **Celery worker** | ghcr.io/konnaxion/app-worker | v1.0.<sha> | Merge in services/common |
| **Airflow** | ghcr.io/konnaxion/reports-etl | v1.0.<sha> | Merge in infra/airflow |

Tag promotion flow: **dev → staging → prod** via Argo CD wave-files. No latest.

## **2 Kubernetes Namespaces & RBAC**

| **Namespace** | **Purpose** | **ServiceAccounts (short)** |
| --- | --- | --- |
| konnaxion-intel | Ekoh & Smart Vote runtime | ekoh-svc, vote-svc, app-worker |
| konnaxion-analytics | Reporting, Airflow, warehouse | reports-api, airflow-scheduler, airflow-worker |
| konnaxion-ops | Monitoring & backups | prometheus, pgbackrest |

ClusterRoles follow least-privilege: each SA can *get/list/watch* its own CRDs plus secrets in its namespace. Image pull uses a shared read-only registry secret.

## **3 Helm Chart Layout**

arduino

CopyEdit

charts/

ekoh/

Chart.yaml

values.yaml # overridable per env

templates/

deployment.yaml

hpa.yaml

service.yaml

ingress.yaml

configmap-env.yaml

smart-vote/

reports/

airflow/

infra-backup/

Each chart ships opinionated defaults; per-env values overlay lives in clusters/<env>/<stack>.yaml.

## **4 Environment Variables & Secrets**

### **4.1 Shared (ConfigMap: intel-common-env)**

| **Env var** | **Default** | **Notes** |
| --- | --- | --- |
| DJANGO\_SETTINGS\_MODULE | config.settings |  |
| LOG\_LEVEL | INFO | Map to gunicorn & Celery |
| REDIS\_URL | redis://redis:6379/0 | Shared cache |

### **4.2 Ekoh / Smart Vote (ConfigMap: ekoh-sv-env)**

| **Env var** | **Maps to parameter** | **Default** |
| --- | --- | --- |
| EKOH\_MULTIPLIER\_CAP | ethical\_multiplier\_cap | 1.50 |
| RAW\_WEIGHT\_QUALITY | raw\_weight\_quality | 1.000 |
| RAW\_WEIGHT\_EXPERTISE | raw\_weight\_expertise | 1.500 |
| RAW\_WEIGHT\_FREQUENCY | raw\_weight\_frequency | 0.750 |
| SMARTVOTE\_CONSENSUS\_STRONG | CONSENSUS\_STRONG\_THRESHOLD | 0.75 |
| SMARTVOTE\_EMERGING\_DELTA | EMERGING\_EXPERT\_THRESHOLD | 0.15 |
| PROMETHEUS\_BASE\_URL | PROMETHEUS\_BASE\_URL | http://prometheus:9090 |

### **4.3 Secrets (Vault-CSI or SealedSecret)**

| **Secret key** | **Service** | **Purpose** |
| --- | --- | --- |
| POSTGRES\_DSN | all | R/W DB connection |
| JWT\_PUBLIC\_KEY | gateway & all | verify auth tokens |
| LEDGER\_RPC\_URL | vote-svc | anchor vote\_ledger hashes |
| AIRFLOW\_FERNET\_KEY | airflow | DAG encryption |

## **5 Resource Requests, Limits & Autoscaling**

| **Deployment** | **Req CPU / RAM** | **Lim CPU / RAM** | **HPA** | **Scale trigger** |
| --- | --- | --- | --- | --- |
| EkohSvc | 250 m / 256 Mi | 500 m / 512 Mi | 2 → 6 | 70 % CPU |
| VoteSvc | 300 m / 384 Mi | 600 m / 768 Mi | 2 → 8 | 70 % CPU or 400 req/s |
| app-worker | 400 m / 512 Mi | 800 m / 1 Gi | 3 → 10 | Celery queue > 500 |
| reports-api | 300 m / 384 Mi | 600 m / 768 Mi | 2 → 6 | 70 % CPU |
| airflow-worker | 400 m / 512 Mi | 800 m / 1 Gi | manual | Airflow pools |

Values mirror §5 of 05-reporting\_analytics.md; prod sizing is doubled for HA.

## **6 Ingress & API Gateway**

* **North-south traffic** hits NGINX Ingress Controller with TLS from LetsEncrypt.
* All paths routed to Gateway (/api/v1/\*) where per-service **internal** host names are resolved by header.
* Rate-limit: 60 req/min/user; burst 200.
* WebSockets (/smart-vote/live/\*) proxied with sticky sessions.

## **7 Observability Stack**

* **Prometheus Operator** – scrapes /metrics on port 9090.
* **Grafana Dashboards** – EkoH Overview, VoteSvc Latency, Celery Queues, Airflow DAGs.
* **Alertmanager** – routes to PagerDuty (severity P1/P2) and Slack #intel-alerts.
* **Loki** + **Grafana Loki** – central log aggregation, 7-day retention.
* **OpenTelemetry** traces exported to Jaeger in staging, Tempo in prod.

## **8 CI/CD Pipeline (GitHub Actions → Argo CD)**

1. **Push / PR** – run pytest, black --check, doc-lint, and SCA.
2. **Image build** – docker buildx bake, tag v1.0.<sha>, push to GHCR.
3. **Chart bump** – update Chart.yaml.version; commit in PR.
4. **Argo CD auto-sync** – watches clusters/<env>; deploys new image tag & chart revision.
5. **Post-deploy smoke test** – GitHub Action workflow calls /healthz & verifies Prometheus targets up.

Rollback = Argo CD sync --revision <prev-commit>.

## **9 Change-Control Checklist**

1. **New env-var** → add to Section 4, patch ConfigMap, reference in 02-parameter\_reference.md.
2. **Resource tweak** → Section 5 & Helm values; update Grafana alert thresholds if needed.
3. **New service** → add image repo (Section 1), namespace, RBAC, chart scaffold.
4. **Ingress path change** → Section 6 + Navigation Map + API Gateway routes.
5. **Secret rotation** → update Vault path & Secret spec; bump service Deployment for reload.

Failure to update this guide **blocks the PR**: CI contains a linter that compares manifests against the tables above.

This document now locks deployment topology, environment configuration, resource sizing, secrets mapping, and the CI/CD conveyor for the EkoH and Smart Vote modules. All operators, SREs, and developers must treat it as the canonical reference before touching infrastructure.

##### **You said:**

10/11

###### **ChatGPT said:**

Thought for 18 seconds

**10-testing\_plan.md — Quality-Engineering & CI Gate Reference (v 1.0)** *Every test artefact, coverage target, load goal and CI guard for EkoH + Smart Vote lives here. If the check isn’t listed, it must not block—or sneak past—production deployments.*

## **0 Purpose & scope**

This plan enforces a **90 % line-coverage baseline** across back-end, front-end and ETL code, plus deterministic integration, e2e, load and security checks before a container tag may roll to staging or prod.

## **1 Test-pyramid & coverage targets**

| **Level** | **Tooling** | **Coverage goal** | **Key subjects** |
| --- | --- | --- | --- |
| **Unit** | pytest, pytest-django, vitest | ≥ 90 % lines per package | score formulas, serializers, React hooks |
| **Integration** | Django APIClient, pytest-asyncio, Dredd contract tests | pass rate 100 % | API routes, DB migrations, cache invalidation |
| **End-to-End** | Playwright | green run on Chrome + Firefox + WebKit | user journeys *home → Smart Vote → CSV export* |
| **Load / Perf** | k6, Locust | p95 < 300 ms @ 200 RPS, error < 1 % |  |
| **Security** | Semgrep, Bandit, OWASP ZAP | 0 high-sev findings | JWT, SQLi, XSS, RBAC |
| **Data / ETL** | pytest + Great Expectations | all expectations green | smart\_vote\_fact, dim\_date, materialised views |

## **2 Unit-test checklist by module**

| **Module** | **Must test…** | **Example cases** |
| --- | --- | --- |
| **EkohSvc** | multidimensional\_scoring maths | weight override, negative raw values |
| **VoteSvc** | WeightCalculator + DUP vote guard | ethics multiplier edge, UNIQUE constraint |
| **reports-api** | serializers, range validation, cache decorator | invalid range > 90 d, CSV header match |
| **Airflow DAGs** | task graphs via airflow dags test | etl\_smart\_vote SQL row count > 0 |
| **React** | hooks & UI components | useReport cache expiry, <BallotForm> error state |

Mock external services with responses, Redis fakeredis, Kafka aiokafka fixtures.

## **3 Integration tests**

| **Contract** | **Tool** | **Approach** |
| --- | --- | --- |
| **OpenAPI** (Vote, Ekoh, reports) | **Dredd** | validate every example payload against /openapi.yaml |
| **DB migrations** | pytest-django | migrate forward+backward; snapshot schema diff clean |
| **Cache** | custom | POST vote → expect Redis set; TTL ≤ 600 s |
| **Kafka flow** | testcontainers-kafka | produce vote.cast, expect vote\_result row |

## **4 End-to-End scenarios**

| **ID** | **Path** | **Assertions** |
| --- | --- | --- |
| SV001 | login → open /smart-vote/foo → cast ballot | toast “Vote saved”, row in vote |
| SV002 | anonymous → open /smart-vote/... | result chart visible, cast button disabled |
| REP001 | /reports/smart-vote 30 d → export CSV | file download ≤ 100 000 rows (config guard) |

Runs headless in CI on chrome; screenshots saved on failure.

## **5 Performance & load**

* **k6** script hits /smart-vote/cast and /smart-vote/result at 200 RPS for 5 min; test fails if p95 > 150 ms (cast) or > 400 ms (result/api) in line with API SLOs.
* **Celery stress** – enqueue 10 k votes; expect aggregator latency < 2 min.

Results upload to Prometheus custom metric ci\_load\_pass{branch="main"}.

## **6 Security & privacy**

| **Check** | **Tool** | **Threshold** |
| --- | --- | --- |
| Static analysis | **Semgrep** coroutine | no high-sev rule violation |
| Dependency scan | GitHub Dependabot + pip-audit | 0 critical CVE |
| DAST | OWASP ZAP scripted | no auth bypass; same-site cookies enforced |

Data-protection tests ensure SHA-256 user hashes persist in analytics DB.

## **7 CI pipeline quality-gates**

On every PR GitHub Actions run:

1. **lint** (ruff, prettier, eslint)
2. **unit + integration tests** (pytest -q, vitest run) – block if coverage < 90 % lines
3. **contract tests** (Dredd)
4. **load/perf quick-shot** (k6 50 RPS, 1 min)
5. **schema-diff** for analytics DB – fail on drift
6. **export-guard** – abort if CSV export would exceed cap
7. **image build** – only after all checks green; then Argo CD sync (see 09-config\_deploy.md).

## **8 Test data management**

* Factory Boy for Django models; pytest-snapshot for deterministic JSON.
* Analytics fixtures generated nightly from staging anonymised dump; truncated to last 7 d.
* Playwright seeds browser local-storage with JWT signed by test RSA key.

## **9 Ownership & review**

| **Area** | **Code-owner** | **Review label** |
| --- | --- | --- |
| EkohSvc / VoteSvc tests | @konnaxion/kollective-intel | INTEL-TEST |
| reports-api, ETL | @konnaxion/data-eng | DATA-TEST |
| Front-end e2e | @konnaxion/ux-guild | UX-TEST |
| Security scans | @konnaxion/sec-ops | SEC-TEST |

A PR must have at least **one** approving review from each affected owner label.

## **10 Change-control checklist**

1. **New endpoint / feature** → add unit + contract tests; update pyramid table.
2. **Coverage threshold change** → governance vote + edit §1 and CI job.
3. **New load SLO** → adjust §5 numbers + Prometheus alert.
4. **Deprecate route/data** → drop relevant tests in same PR; ensure quality gate still ≥ 90 %.

This plan is now the *single source of truth* for all testing and quality gates across EkoH and Smart Vote. CI enforces the rules above; an out-of-date MR will fail fast until the document and reality converge.

Sources

##### **You said:**

11/11

###### **ChatGPT said:**

Thought for 7 seconds

**00-docs\_index.md — Master Documentation Directory (EkoH + Smart Vote, release v1.0)** *(Bookmark this page: every other file hangs off the anchors below. If you rename a doc or a section heading, update the map immediately or CI will fail.)*

## **1 Library at a Glance**

| **#** | **File** | **What question it answers** | **Last-edited / version** |
| --- | --- | --- | --- |
| **00** | **00-docs\_index.md** | “Where do I find X?” — this table of contents | *2025-08-07 / v1.0* |
| **01** | 01-db\_schema.md | Exact tables, PK/FK, ERD, retention rules | v1.0 |
| **02** | 02-parameter\_reference.md | All tunable weights, caps, enums, env-vars | v1.0 |
| **03** | 03-technical\_spec.md | Layer-by-layer APIs, algorithms, flows | v1.0 |
| **04** | 04-inventory\_of\_features.md | Feature catalogue → code-name cross-ref | v1.0 |
| **05** | 05-reporting\_analytics.md | Dashboards, star-schema, ETL, metrics | v1.0 |
| **06** | 06-insights\_module.md | Feature store, ML models, drift rules | v1.0 |
| **07** | 07-integration\_mapping.md | Routes, Kafka topics, auth scopes, Webhooks | v1.0 |
| **08** | 08-tasks\_schedule.md | Celery, Airflow, Cron—when and how they run | v1.0 |
| **09** | 09-config\_deploy.md | Helm values, env-var matrices, resource sizes | v1.0 |
| **10** | 10-testing\_plan.md | Test pyramid, coverage gates, CI workflow | v1.0 |

**Tip for new contributors** *Need a constant?* — look in **02**.  
 *Need an API payload or path?* — check **03** then **07**.  
 *Want to know why a scheduled job failed?* — see **08** then Grafana links in **09**.

## **2 Navigation Rules & Stable Anchors**

* **Filenames are immutable** once shipped. Patch this index first, then apply the rename across code & CI.
* **Section IDs** (markdown headings) serve as intra-doc anchors; keep wording stable or add an explicit <a id="">.
* **CI doc-lint** (docs-link-checker) resolves every link on PR; a broken anchor blocks merge.

## **3 Versioning & Change-Control**

| **Change type** | **Required actions** |
| --- | --- |
| **Add new doc** | Choose next number (11, 12…), add row here, create file skeleton with {#} header. |
| **Major breaking change** (schema, route, env-var) | Increment minor version on *all affected docs*; update date stamps in table above. |
| **Minor clarifications** | Edit doc, bump patch in front-matter, no index change needed. |
| **Deprecation** | Mark row as ***deprecated***, point to replacement, remove after 2 minor versions. |

Front-matter template for every doc:

yaml

CopyEdit

---

title: 03-technical\_spec

version: v1.0

updated: 2025-08-07

---

## **4 Ownership & Review Matrix**

| **File pattern** | **Code-owner GitHub team** |
| --- | --- |
| 0\*-\*.md (docs in this library) | **@konnaxion/doc-guild** |
| services/ekoh/\*\* services/vote/\*\* | **@konnaxion/kollective-intel** |
| services/reports/\*\* infra/airflow/\*\* | **@konnaxion/data-eng** |
| charts/\*\* clusters/\*\* | **@konnaxion/sre-team** |

At least one approving review from the relevant owner is mandatory; doc PRs without an index update will be auto-rejected.

## **5 Quick-jump Markdown Links**

* DB schema → EkoH tables
* Parameter registry → Smart Vote section
* Tech spec → Weight calculator pseudocode
* Report ETL schedule
* Airflow DAG registry
* Kafka topic list
* Celery task timetable
* Helm env-var matrix
* Load test thresholds

## **6 Onboarding Checklist (1-hour crash path)**

1. Clone repo → run make init-docs → open this index.
2. Skim **01** schema & **02** parameters (10 min).
3. Read **03** tech spec weight section (10 min).
4. Fire up /docs/architecture.drawio for high-level component map (5 min).
5. Run pytest -q & npm test (15 min).
6. Open Grafana at /d/ekoh-overview to see live metrics (10 min).

With this index in place, the EkoH + Smart Vote knowledge base is fully self-navigable.  
 Keep links alive, headers consistent, and versions current—everything else will follow.