### **Document 5.1 – Reporting & Analytics · Frontend layer**

*(“Insights” module UI, aligned with Konnaxion Technical Spec v14)*

#### **1 . Portée**

Décrit **toutes les interfaces utilisateur React** permettant de consulter les indicateurs analytiques globaux : Smart Vote, activité projet, performance d’auth, adoption générale. Seules les vues lecture (read‑only) sont couvertes ; l’ingestion et le calcul relèvent des couches 5.2 – 5.4.

#### **2 . Routes & navigation**

| **Route** | **Page** | **Description** | **API(s) consommée(s)** |
| --- | --- | --- | --- |
| /reports | InsightsHomePage | Hub cartes : Smart Vote, Activité, Santé API | — |
| /reports/smart-vote | SmartVoteDashboard | Tendance des votes & corrélations | GET /reports/smart-vote |
| /reports/usage | UsageDashboard | MAU / projets / docs par domaine | GET /reports/usage |
| /reports/perf | PerfDashboard | Latence & erreurs SLO | GET /reports/perf |
| /reports/custom | CustomBuilderPage | (beta) builder de requêtes | WebSocket /ws/reports/custom |

*La barre latérale “Insights” s’ouvre depuis l’icône graphique du layout principal (0.1).*

#### **3 . Composants principaux**

| **Composant** | **Rôle** | **Lib. graphique** |
| --- | --- | --- |
| **<SmartVoteChart>** | Bar + line mix, 2 axes (votes & score) | Chart.js 4 |
| **<UsageBigNumbers>** | 3 cartes “Projects / Docs / Active users” | Tailwind utilities |
| **<DomainHeatMap>** | Carte thermique docs × domaines | Chart.js matrix |
| **<LatencySLOGauge>** | Gauge latence P95 vs SLO | Chart.js doughnut |
| **<ErrorRateSparkline>** | Sparkline 24 h erreurs % | Chart.js line |
| **<TimeRangePicker>** | Sélecteur absolu / relatif (24 h, 7 j, 30 j) | AntD RangePicker |
| **<ExportCSVButton>** | Télécharge dataset courant (?format=csv) | — |

Tous composants respectent la palette tokens (0.1 §2) ; pas de couleurs codées en dur.

#### **4 . Gestion d’état & données**

* **React Query 5** :  
  + useReport(endpoint, params) (cache 5 min ou variable si auto‑refresh actif).
  + Invalidation sur changement de plage temps.
* **WebSocket** (page custom) : hook useReportStream() (Channel group reports\_user\_<id>).
* **Redux** non requis – navigation locale par context.

#### **5 . Flux utilisateur (Smart Vote)**

sequenceDiagram

U->>F: Open /reports/smart-vote

F->>API: GET /reports/smart-vote?range=30d

API-->>F: JSON dataset

F-->>U: Charts render

U->>F: Change range → 7d

F->>API: GET /reports/smart-vote?range=7d ← cached?

API-->>F: JSON

Latence cible UI < 200 ms (chargement chart après réponse).

#### **6 . Accessibilité & i18n**

* Charts accompagnés de tableaux aria‑labelledby (pour lecteurs d’écran).
* Couleurs vérifiées WCAG AA.
* Tous labels dans namespace reports.

#### **7 . Tests**

| **Niveau** | **Couverture** |
| --- | --- |
| Unit | Calcul axes, fallback “no data” |
| Integration | API fetch, cache invalidation, time‑range picker |
| e2e | Parcours home → SmartVote → export CSV |

Cible ≥ 90 % lignes.

#### **8 . Dépendances**

* **Chart.js 4** (import tree‑shaken).
* **Ant Design 5** pour formulaires / pickers.
* No third‑party state lib en plus de React Query.

#### **9 . Structure dossier**

/packages/reports-frontend

/components

SmartVoteChart.tsx

UsageBigNumbers.tsx

DomainHeatMap.tsx

LatencySLOGauge.tsx

ErrorRateSparkline.tsx

TimeRangePicker.tsx

/pages

InsightsHomePage.tsx

SmartVoteDashboard.tsx

UsageDashboard.tsx

PerfDashboard.tsx

CustomBuilderPage.tsx

hooks/

useReport.ts

useReportStream.ts

index.ts

Build publie @konnaxion/reports-ui (ESM + types).

#### **10 . Conformité correctifs**

* Composant **SmartVoteChart** implémenté (bar + line).
* Pages perf & usage incluent SLO gauges et heatmap.
* Export CSV bouton présent sur chaque dashboard.
* Aucune mention TBD ; tout aligné documentation v14.

### **Document 5.2 – Reporting & Analytics · Backend layer**

*(service* ***reports‑api*** *aligned with Konnaxion Technical Specification v14; complements Frontend 5.1 and DevOps 5.4)*

#### **1 . Scope**

reports‑api is a **read‑only micro‑service** that exposes pre‑aggregated analytics produced nightly (and, for some metrics, near‑real‑time).  
 It **does not** write business data; it reads star‑schema tables managed by the ETL pipelines defined in layer 5.4.

#### **2 . Technology stack**

| **Aspect** | **Decision** |
| --- | --- |
| Framework | Django 4.2 + Django REST Framework 3.15 |
| ORM / DB | Read‑only PostgreSQL 16 (analytical cluster, separate from OLTP) |
| Caching | Redis 7 (TTL 600 s default) |
| Asynchronous refresh | Airflow 2.9 DAGs (see 5.4) |
| Auth | Same JWT middleware as Core 0.2; all endpoints require IsAuthenticated |
| Rate limit | 60 req/min per user (UserRateThrottle) |

#### **3 . Data contracts (OpenAPI excerpts)**

| **Endpoint** | **Method** | **Query params** | **Response 200** | **Cache TTL** |
| --- | --- | --- | --- | --- |
| /reports/smart-vote | **GET** | range (24h,7d,30d,custom), grouping (day/week) | {labels[], votes[], avg\_score[]} | 600 s |
| /reports/usage | **GET** | range, grouping, domain\_code? | {labels[], mau[], projects[], docs[]} | 600 s |
| /reports/perf | **GET** | range, endpoint? (/auth/login, …) | {labels[], p95\_latency[], error\_rate[]} | 300 s |
| /reports/export | **GET** | report (smart-vote, usage, perf), format (csv,json), same filters as source | File stream | 60 s |

*All time ranges must be ≤ 90 days; otherwise API returns 400 INVALID\_RANGE.*

#### **4 . Database schema (read‑only star)**

* **Fact tables** (managed by ETL):  
  + smart\_vote\_fact (grain vote × question × date) – correctif 5.3
  + usage\_mau\_fact (user × month)
  + api\_perf\_fact (endpoint × hour)
* **Dim tables**: dim\_date, dim\_domain, dim\_endpoint.

Indices are column‑store (PG16 zstd compression). Views vw\_smart\_vote\_30d, vw\_usage\_90d, vw\_perf\_24h simplify queries.

#### **5 . Caching & invalidation policy**

* Key pattern reports:{endpoint}:{sha256(params)}.
* Spring‑cleaner Celery task purges keys older than 12 h.
* Export route sets Content‑Disposition: attachment; filename="report\_<type>\_<date>.csv".

#### **6 . Permissions & audit**

* Only users with role ADMIN or higher may call /reports/export.
* Each call adds entry to audit\_request\_log (user\_id, path, ip, ts).
* No PII leaves the service; results are aggregated ≥ 10 records (k‑anonymity).

#### **7 . Error codes (JSON Error format 0.2)**

| **Code** | **HTTP** | **Meaning** |
| --- | --- | --- |
| INVALID\_RANGE | 400 | Range > 90 days or start > end |
| UNSUPPORTED\_FORMAT | 400 | Format ≠ csv,json |
| ENDPOINT\_UNKNOWN | 400 | Perf endpoint filter invalid |
| EXPORT\_FORBIDDEN | 403 | Role < ADMIN |

#### **8 . Performance targets (SLO for API only)**

| **Path** | **P95 Latency** | **Error rate** |
| --- | --- | --- |
| GET /reports/smart-vote | ≤ 400 ms | ≤ 1 % |
| GET /reports/usage | ≤ 400 ms | ≤ 1 % |
| GET /reports/perf | ≤ 400 ms | ≤ 1 % |
| GET /reports/export | ≤ 800 ms | ≤ 1 % |

Latency measured post‑cache miss (worst case).

#### **9 . Unit of work sequence (Smart Vote)**

sequenceDiagram

U->>API: GET /reports/smart-vote?range=7d

API->>Redis: GET key

alt cache hit

Redis-->>API: dataset

else cache miss

Redis-->>API: null

API->>DB: SELECT \* FROM vw\_smart\_vote\_30d WHERE date >= now()-7d

DB-->>API: rows

API->>Redis: SET key TTL 600

end

API-->>U: 200 JSON

#### **10 . Tests & quality gate**

* **Unit**: serializers, range‑validation, cache decorator.
* **Integration**: query → db snapshot → compare JSON.
* **Contract**: Dredd against OpenAPI on CI.
* **Load**: k6 200 RPS sustained ; p95 < 300 ms with Redis hot.

#### **11 . Dependencies**

* Python 3.12, Django 4.2, DRF 3.15
* Redis‑py 5, psycopg 3
* django‑redis‑cache with TLS.

#### **12 . Alignment with docs 5.1 & 5.4**

* Endpoints match exactly those consumed by Frontend 5.1.
* ETL jobs (etl\_smart\_vote, etl\_usage, etl\_perf) defined in DevOps 5.4 populate the fact tables.
* Monitoring dashboards pull Prometheus scrape from /metrics exporter in this service.

All backend analytics endpoints are now specified with fixed parameters, ranges, caches and error semantics—no “TBD” left outstanding.

### **Document 5.3 – Reporting & Analytics · Database & Storage layer**

*(star‑schema schema for the “Insights” module, aligned with v14)*

#### **1 . Scope**

Defines **all relational objects** that power read‑only analytics served by reports‑api (5.2): fact tables, dimensions, materialised views, indexes, retention and ETL interfaces.  
 This schema is deployed on the dedicated **analytical PostgreSQL 16 cluster** (separate from OLTP).

#### **2 . Star‑schema overview**

┌──────────────┐

│ dim\_date │

└─────┬────────┘

│ (FK)

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

│ Fact tables (grain) │

├──────────────────────────────────────────────────────────┤

│ smart\_vote\_fact vote × question × date │

│ usage\_mau\_fact user × month │

│ api\_perf\_fact endpoint × hour │

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

│

┌─────────┴─────────┐

│ other dimensions │

└───────────────────┘

#### **3 . Core dimension tables**

| **Table** | **Key(s)** | **Columns (relevant)** | **Notes** |
| --- | --- | --- | --- |
| **dim\_date** | date\_id (INT) | calendar\_date, year, month, week, day, iso\_week | Pre‑populated 2000‑01‑01 … 2035‑12‑31 |
| **dim\_domain** | domain\_id | domain\_code ENUM(domain\_code) | Maps to OLTP domain codes |
| **dim\_endpoint** | endpoint\_id | path VARCHAR(80) | Enumerates tracked REST paths |

All dims are static, SMALLINT / surrogate keys, and compressed with ENCODING zstd.

#### **4 . Fact tables (partitioned by date)**

##### **4.1 smart\_vote\_fact**

| **Column** | **Type** | **Comment** |
| --- | --- | --- |
| id | UUID PK |  |
| date\_id | INT FK dim\_date |  |
| domain\_id | INT FK dim\_domain |  |
| question\_id | UUID |  |
| user\_id | UUID (*hash‑anonymised*) |  |
| vote\_value | NUMERIC(5,2) |  |
| score\_normalised | NUMERIC(5,2) |  |

*Partition* : monthly by date\_id.  
 *Index* : (domain\_id, date\_id) BRIN.  
 ETL task **etl\_smart\_vote** (5.4) writes nightly + delta every 10 min.

##### **4.2 usage\_mau\_fact**

| **Column** | **Type** |
| --- | --- |
| id UUID PK |  |
| month\_id INT FK dim\_date (first day of month) |  |
| domain\_id INT FK dim\_domain |  |
| mau INT |  |
| projects\_created INT |  |
| docs\_uploaded INT |  |

Partition by year (range).  
 Index (month\_id, domain\_id) B‑tree.

##### **4.3 api\_perf\_fact**

| **Column** | **Type** |
| --- | --- |
| id UUID PK |  |
| date\_id INT FK dim\_date |  |
| hour\_of\_day SMALLINT (0‑23) |  |
| endpoint\_id INT FK dim\_endpoint |  |
| p95\_latency\_ms INT |  |
| error\_rate\_pct NUMERIC(4,2) |  |
| request\_count BIGINT |  |

Partition by month via date\_id.  
 Index (endpoint\_id, date\_id, hour\_of\_day).

#### **5 . Materialised views**

| **View** | **Definition (summary)** | **Refresh** |
| --- | --- | --- |
| vw\_smart\_vote\_30d | Aggregates smart\_vote\_fact last 30 days (sum votes, avg score) | On‑demand via reports‑api; ETL triggers incremental refresh |
| vw\_usage\_90d | Rolling MAU / docs / projects, grouped daily | Nightly 02:00 UTC |
| vw\_perf\_24h | P95 & error‑rate per endpoint last 24 h | Every 15 min |

All views use WITH NO DATA in migration; first population by Airflow DAG.

#### **6 . Migration order (extract)**

1. 006\_dim\_core.sql – creates dim\_date, dim\_domain, dim\_endpoint, seeds calendar.
2. 007\_fact\_smart\_vote.sql
3. 008\_fact\_usage\_mau.sql
4. 009\_fact\_api\_perf.sql
5. 010\_mat\_views\_reports.sql – defines & indexes materialised views.

#### **7 . Retention & purge**

| **Object** | **Retention** | **Mechanism** |
| --- | --- | --- |
| smart\_vote\_fact | 5 years partitions, then drop | Yearly drop oldest partition |
| usage\_mau\_fact | 10 years | Same |
| api\_perf\_fact | 2 years | Same |
| Materialised views | kept latest; auto‑refresh overwrites | — |

#### **8 . Security & PII rules**

* user\_id in facts is hashed with SHA‑256(secret) to ensure irreversibility.
* ETL discards rows where cohort < 10 (k‑anonymity).
* DB role reports\_reader has SELECT only on materialised views and dims; fact tables restricted to ETL service account.

#### **9 . Performance baselines**

* Query SELECT \* FROM vw\_perf\_24h WHERE endpoint\_id = X returns ≤ 50 ms P95.
* Monthly partition attach/detach < 2 s (tested on 100 M rows smart\_vote).

All analytical storage objects required for the Reporting & Analytics slice are now specified without ambiguities and align with service (5.2) and UI (5.1) contracts.

### **Document 5.4 – Reporting & Analytics · DevOps / Infrastructure layer**

*(infrastructure for* ***reports‑api*** *and analytics ETL, aligned with Konnaxion Tech Spec v14)*

#### **1 . Objective**

Specify **all Kubernetes manifests, Airflow jobs, environment variables, alert rules and backup policies** required to operate the Reporting & Analytics slice (layer 5).  
 Everything common (Ingress, Vault, Prometheus, GH Actions pipeline skeleton) is defined in Document 0.4; here we only describe module‑specific additions.

#### **2 . Runtime components**

| **Resource** | **Image** | **Replicas / HPA** | **CPU m (req / lim)** | **RAM Mi (req / lim)** |
| --- | --- | --- | --- | --- |
| **Deployment reports-api** | ghcr.io/konnaxion/reports-api:<sha> | 3 (HPA 2 → 6) | 300 / 600 | 384 / 768 |
| **Deployment reports-etl-worker** | ghcr.io/konnaxion/reports-etl:<sha> | 2 (manual) | 400 / 800 | 512 / 1024 |
| **Airflow (analytics-airflow)** | apache/airflow:2.9‑python3.12 | 1 scheduler, 2 workers | 250 / 500 | 512 / 1024 |
| **Job reports-db-migrate** | same as reports-api | run‑once on deploy | 250 | 256 |

HPA for reports-api triggers on **CPU > 60 %** or **latency P95 > 400 ms** (Prometheus custom metric).

#### **3 . Airflow DAGs (mounted in /opt/airflow/dags)**

| **DAG id** | **Schedule** | **Task outline** | **Target** |
| --- | --- | --- | --- |
| etl\_smart\_vote | \*/10 \* \* \* \* | load OLTP deltas → smart\_vote\_fact | Analytics PG |
| etl\_usage | 0 \* \* \* \* | hourly MAU update | Analytics PG |
| etl\_perf | \*/15 \* \* \* \* | ingest Prometheus API perf → api\_perf\_fact | Analytics PG |
| refresh\_mat\_views | 5 \* \* \* \* | REFRESH vw\_\* materialised views | Analytics PG |
| cleanup\_cache | @hourly | purge Redis keys > 12 h | Redis |
| purge\_old\_partitions | 0 4 \* \* 0 | drop partitions past retention | Analytics PG |

Airflow connections (pg\_analytics, redis\_reports) are injected via Vault secrets backend.

#### **4 . Environment variables & secrets**

| **Variable** | **Source** | **Notes** |
| --- | --- | --- |
| REPORTS\_DB\_URL | Vault secret/analytics/pg/url | read‑only user |
| REDIS\_URL | Vault secret/common/redis/url | TTL cache |
| AIRFLOW\_\_CORE\_\_FERNET\_KEY | Vault secret/analytics/airflow/fernet |  |
| PROMETHEUS\_BASE\_URL | ConfigMap analytics-settings | for etl\_perf |
| EXPORT\_MAX\_ROWS | ConfigMap analytics-settings | 100000 |

Secrets injected by CSI Vault; ConfigMap mounted read‑only.

#### **5 . CI/CD extensions**

| **Job** | **Purpose** |
| --- | --- |
| **schema‑diff** | verifies migrations match star‑schema (deny drift) |
| **dag‑lint** | pylint + airflow dags list --safe-mode |
| **perf‑budget** | fails build if k6 latency > 350 ms @ 200 RPS |
| **export‑guard** | ensures CSV export ≤ EXPORT\_MAX\_ROWS |

Build push tags images reports-api and reports-etl; Argo CD auto‑sync to staging, then manual promote prod.

#### **6 . Prometheus metrics & alerts**

| **Alert name** | **Expression** | **Severity** |
| --- | --- | --- |
| **reports\_latency\_p95** | histogram\_quantile(0.95, rate(reports\_request\_seconds\_bucket[5m])) > 0.4 | critical |
| **reports\_error\_rate** | sum(rate(reports\_requests\_total{status=~"5.."}[5m])) / sum(rate(reports\_requests\_total[5m])) > 0.02 | warning |
| **etl\_task\_fail** | airflow\_dag\_run\_failed\_total{dag\_id=~"etl\_.\*"} > 0 | critical |
| **mat\_view\_refresh\_duration** | reports\_matview\_refresh\_seconds > 60 | warning |
| **redis\_cache\_hit\_ratio\_low** | rate(reports\_cache\_hit\_total[5m]) / rate(reports\_cache\_req\_total[5m]) < 0.5 | info |

Dashboards reports-api.json and report-etl.json show throughput, cache hit rate, ETL runtime.

#### **7 . Runbooks**

* **High API latency** → scale reports-api (check HPA), inspect slow query log on analytics PG.
* **ETL failure** → Airflow UI → retry task; if database lock, run SELECT \* FROM pg\_locks and kill blocking PID.
* **Mat view refresh over 60 s** → vacuum partition, review index bloat.
* **Redis miss flood** → increase maxmemory, verify cache key churn.

#### **8 . Backup & disaster recovery**

| **Asset** | **Strategy** |
| --- | --- |
| Analytics PG | pgBackRest full nightly, incremental 30 min, retain 14 days |
| Airflow metadata DB | included in pgBackRest |
| DAG code | in Git; Argo CD redeploy |
| Redis cache | no backup (ephemeral) |

Quarterly restore‑drill: restore analytics snapshot to staging and run sample /reports queries.

#### **9 . Security & compliance**

* All facts store **hashed user IDs** (see 5.3).
* Role reports\_reader is enforced via Postgres search\_path and RLS OFF.
* API export limited to 100 k rows and only for ADMIN role.
* Audit log (audit\_request\_log) shipped via Fluent Bit to OpenSearch; retention 6 months.

#### **10 . SLO mapping**

| **KPI** | **Objective** | **Alert** |
| --- | --- | --- |
| GET /reports/\* P95 | ≤ 400 ms | reports\_latency\_p95 |
| Error rate | ≤ 2 % 5 min | reports\_error\_rate |
| ETL task success | 100 % daily | etl\_task\_fail |
| Mat view refresh | < 60 s | mat\_view\_refresh\_duration |

All Reporting & Analytics infrastructure elements are now fully specified: deployments, Airflow DAGs, variables, monitoring and recovery procedures, completing the layer 5 documentation set.