JobTrends Scraping-Monitoring – Technische Zusammenfassung

Stand: Februar 2026
Ziel: Stabiles, skalierbares Monitoring der nächtlichen Scrape-Runs mit einem morgens (08:00 Uhr) aktuellen Looker-Report.
Leitprinzip: *Supabase = Source of Truth / Business-Logik, BigQuery = Analytics- & BI-Layer, Looker = Visualisierung.*

---

1. Zielbild

- Scraper laufen nachts (unterschiedliche Unternehmen, unterschiedliche Laufzeiten)
- Alle aggregierten Monitoring-Informationen liegen **bereits in Supabase** (Views)
- BigQuery dient **nur als Spiegel / Analytics-Layer** für Looker Studio
- Looker Studio enthält **keine Business-Logik**
- Um **08:00 Uhr morgens** ist der Monitoring-Report garantiert aktuell

---

2. Finale Architektur

```
Supabase (Postgres)  ← Source of Truth
  └─ daily_summary_v
  └─ daily_summary_looker_v
        ↓ (Batch-Export)
Cloud Run Job (Python)
        ↓
BigQuery (Analytics Layer)
  └─ dataset: monitoring
      └─ table: daily_summary
        ↓
Looker Studio (Monitoring Dashboard)
```

---

3. Supabase – Status

Rolle

- **Operative Datenhaltung**
- Vollständige Business-Logik (Status, Flags, Baselines, Drop-Detection)

Relevante Views

`daily_summary_v`

- interne Aggregation
- enthält komplette Bewertungslogik

`daily_summary_looker_v`

- BI-Wrapper
- exakt 1 Zeile pro `company_key × run_date`
- keine Joins oder Berechnungen in Looker nötig

Enthaltene Logik (SQL)

- 14-Tage-Baselines
- Drop-Detection
- HTTP-Fail- & Block-Rates
- finale Statusklassifikation:
- `success`
- `partial`
- `failed`
- `no_data`

---

4. BigQuery – Rolle & Design

Rolle

- **Read-only Analytics-Layer** für BI
- Keine operative Logik
- Keine eigenen Berechnungen

Dataset

- `monitoring`
- Region: EU

Tabelle

- `monitoring.daily_summary`

Schema (v1)

- `company_key` (STRING)
- `run_date` (DATE)
- `final_status` (STRING)
- `is_success` (INTEGER)
- `is_problem` (INTEGER)
- `jobs_processed_best` (INTEGER)
- `http_fail_rate_sum` (FLOAT)

- `http_block_rate_sum` (FLOAT)
- `baseline_jobs_processed_14d` (FLOAT)
- `execution_time_sec_best` (FLOAT)

Design-Entscheidungen

- **keine Partitionierung** (bewusst, MVP-freundlich)
- **MERGE-basierter Upsert** auf `(company_key, run_date)`
- Re-Runs und Korrekturen explizit erlaubt

---

5. Export-Mechanismus (Supabase → BigQuery)

Technologie

- **Cloud Run Job** (kein HTTP-Endpoint)
- Python-Container

Authentifizierung

- Supabase: Postgres Read-Only User (`looker_export_ro`)
- Zugang über **Secret Manager** (`supabase-dsn`)
- BigQuery: Service Account Identity (kein JSON-Key)

Service Account

- `jobtrends-exporter@work-for-elon.iam.gserviceaccount.com`

IAM-Rollen

- BigQuery Job User (Projekt)
- BigQuery Data Editor (Dataset `monitoring`)
- Secret Manager Secret Accessor (Secret `supabase-dsn`)
- Cloud Run Job Runner

Laufverhalten

- Export der letzten `N` Tage (aktuell: 3)
- Daten werden:
- aus `daily_summary_looker_v` gelesen
- JSON-normalisiert (Decimal → float)
- in eine Staging-Tabelle geladen
- per `MERGE` in `monitoring.daily_summary` übernommen

---

6. Scheduling (Automatisierung)

Ziel

- Looker-Report um **08:00 Uhr morgens aktuell**

Umsetzung

- **Cloud Scheduler**
- Trigger: **HTTP → Cloud Run Jobs API**
- Auth: **OAuth-Token** (nicht OIDC)

Zeitplan

- Cron: `45 7 * * *`
- Zeitzone: `Europe/Berlin`

Ziel-URL

```
https://run.googleapis.com/v2/projects/work-for-elon/locations/europe-west3/jobs/export-supabase-to-bq:run
```

---

7. Looker Studio – aktueller Stand

Datenquelle

- BigQuery
- Tabelle: `monitoring.daily_summary`

Konfiguration

- `run_date` als Date Range Field
- `is_success`, `is_problem` als numerische Aggregationen
- Keine SQL-Logik in Looker

Erwartetes Verhalten

- Report zeigt **morgens konsistente Monitoring-Daten**
- Kein Direktzugriff auf Supabase notwendig

---

8. Zentrale Lessons Learned

- Looker Studio ist **kein stabiler Postgres-Client**
- BigQuery ist **kein OLTP-System**, sondern Analytics-Layer
- Business-Logik gehört **in SQL, nicht ins Dashboard**
- Entkopplung spart massiv Debug-Zeit
- Cloud Run Jobs sind ideal für Batch-Exports
- Scheduler → Jobs API benötigt **OAuth**, nicht OIDC

---

9. Aktueller Status (Ergebnis)

- Supabase → BigQuery Export läuft stabil

- IAM & Secrets korrekt eingerichtet
- Scheduler triggert Job erfolgreich
- BigQuery enthält tagesaktuelle Monitoring-Daten
- Basis für Looker Studio ist fertig

---

10. Nächster logischer Schritt

Aufbau des Looker Studio Monitoring-Dashboards: - Health-Header (Status heute) - Company-Übersicht (success / partial / failed) - Trends (Jobs, Fail-Rates, Laufzeiten)

Dieses Dokument dient als Referenz für weitere Chats, Erweiterungen und neue Projekte.