MetX Platform Expansion: Chatbot Integration & Generation API

for Meteomatics

LaiLIX meteomatics

Executive Summary

- **What we're delivering**: An end-to-end **dashboard generation capability** that lets users describe a map in natural language; the system generates a **complete** `metx.json` and returns it to MetX for display.
- **Scope (MVP)**: **Exactly one dashboard** with **one tab** containing **one map** with **multiple layers**.
- **Defaults**: Projection **Mercator**; basemap **Topographic** at index **0** and opacity **1.0**; color-map layers default opacity **~0.7**; isolines/vectors **1.0**; **calibrated=false**; **Meteomatics Mix** as default model.
- **Architecture**: Chatbot (VoiceFlow) → Generation Platform (API, validation, assembly) → **MetX Frontend**.
- **Critical dependency**: The **MetX team implements a "Load Dashboard" endpoint** (client dependency) that accepts a validated `metx.json` and renders it. (We can pilot with manual import meanwhile.)
- **Usage**: **10,000 dashboard generations included** (covers testing + production; actual count may vary slightly with prompt sizes).
- **Timeline**: **~5 weeks** to MVP.
- **Commercials**: **Total project value 25,000 CHF**. Already completed **two exploratory projects = 6,000 CHF**. **Remaining investment: 19,000 CHF** (30/40/30 milestones). Optional support retainer: **2,500 CHF/month**.
- **Models**: We use **Gemini 2.5 Flash and other models via OpenRouter** (multi-provider).
- **Assurance**: Delivery against **acceptance criteria** with a **30-day bug-fix warranty**.

Matthias Heim · matthias@lailix.com · +41 76 268 64 06

Vision & Introduction

We will **implement** dashboard creation capabilities directly into the MetX experience. Users will describe what they want (e.g., “precipitation and wind over UK”) and receive a ready-to-load `metx.json`. The solution includes a production-ready API, strict JSON validation, and a chatbot that orchestrates the flow.

Key components

- **Chatbot integration** (VoiceFlow) with knowledge base sourced from MetX documentation (synced **daily**).
- **Generation Platform** (enhanced prompting tool) with production API, validation, and evaluation.
- **Turnkey delivery** (Vercel + Supabase) with repository, docs, and ops instrumentation.
- **Included usage**: **10,000 dashboard generations** (testing + production).

High-Level Architecture

Flow

1. **User** chats in the MetX web widget.
2. **Chatbot** detects “generate dashboard” intent → calls Generation Platform API.
3. **Generation Platform** selects production prompt/model, generates **layers**, extracts **location/viewport**, validates and assembles a complete `metx.json`.
4. **MetX Frontend** renders the returned dashboard via a **client-owned “Load Dashboard” endpoint**.

Scope (MVP – precisely defined)

- **One dashboard**
- **One tab**
- **One map** (per tab)
- **Multiple layers** supported within that map (color maps, and additional layer types as covered by provided examples/schema)
- **No** multi-panel layouts, time navigation UI, or multi-tab dashboards in MVP.

Explicit rendering defaults

- **Projection**: **Mercator** by default.

Matthias Heim · matthias@lailix.com · +41 76 268 64 06

- **Basemap/Cartographic layer**: **Topographic**, index **0**, opacity **1.0**.
- **Layer opacities**: color maps **~0.7**, isolines/vectors **1.0**.
- **Calibrated**: **false** by default.
- **Model**: **Meteomatics Mix** by default.

Component Deep-Dive

Chatbot (MetX Assistant)

- **Platform**: **VoiceFlow** for web widget, orchestration, and knowledge base.
- **Knowledge base**: MetX docs **synced daily** from the website; manual re-index available.
- **Role**: Gather the user request, call the Generation Platform API, and pass the validated JSON to the MetX Frontend “Load Dashboard” endpoint.

Generation Platform (Enhanced Prompting Tool)

- **API**: `POST /api/v1/dashboard/generate` with API-key auth.
- **Parallel processing**:
  - **Layer Generation** (LLM with production prompt → `layers[]`).
  - **Location Extraction** (separate LLM pass for center/zoom with validation & fallbacks).
- **Validation & Assembly**:
  - Robust parsing of LLM responses.
  - `JsonValidator` fills required fields (IDs, defaults), ensures **MetX-compatible** structure.
  - Prefix/suffix wrapping to produce final **`metx.json`**.
- **Models**: **Gemini 2.5 Flash** and other models via **OpenRouter** (multi-provider).
- **Observability**: Per-run **latency and cost** metrics, structured logs, error tracking.
- **Security**: API-key auth, rate limiting, and private Supabase storage buckets.

MetX Frontend (Client Dependency)

- **Required**: A **“Load Dashboard” endpoint** that accepts a validated `metx.json` and renders it in the current session.
- **Pilot fallback**: If preferred initially, continue with **manual JSON import** via the existing UI while the endpoint is being implemented.

API Contract (Generation Platform)

Endpoint: `POST /api/v1/dashboard/generate`
Request:

Matthias Heim · matthias@lailix.com · +41 76 268 64 06

```
{
  "prompt": "Show the 24-hour precipitation forecast for Central Europe with wind barbs"
}
```

Response (success)

```
{
  "status": "success",
  "metx_json": { ... }   // Complete, validated dashboard JSON (one tab, one map, multi-layer)
}
```

Internal steps

1. Authenticate via API key.
2. Retrieve production **prompt** and **model** from Supabase.
3. Run **Layer Generation** and **Location Extraction** in parallel.
4. Validate and auto-fix with **JsonValidator**; apply **defaults** (projection, basemap, opacities, calibrated).
5. Wrap with prefix/suffix to finalize `metx.json`.
6. Return the JSON.

Implementation Plan (≈ 5 weeks)

1. **Week 1 – Foundations**
    - Supabase schema (prompts/models/analytics), API skeleton, auth & rate limiting.
    - Daily doc sync to chatbot knowledge base.
2. **Week 2 – Generation**
    - Production prompt and model flow.

    - Layer Generation + Location Extraction (parallel).

3. **Week 3 – Validation & Assembly**
    - `JsonValidator` with auto-fix, defaults, and hard schema checks.
    - Prefix/suffix assembly; end-to-end happy path.

4. **Week 4 – Observability & Hardening**
    - Cost/latency metrics, error handling, retries/fallbacks.
    - Load testing, evaluation runs on the provided test cases.

5. **Week 5 – Pilot Readiness**
    - Documentation; production deployment (Vercel + Supabase).
    - VoiceFlow configuration & handover.
    - Support pilot with manual import or switch to client “Load Dashboard” endpoint when ready.

Data, Security & Operations

- **Data processing**: No PII collected; only prompts, metadata, and JSON artifacts for generation & troubleshooting.
- **Model training**: Prompts/outputs are **not** used to train foundation models.
- **Residency**: Data residency in EU can be configured on request.
- **Supabase**: Row-Level Security; private storage buckets for uploads; least-privilege service roles.
- **Monitoring**: Centralized logs, per-run cost tracking, basic budget alerts.

Usage & Costs

- **Included**: **10,000 dashboard generations** (covers testing + production).
- **Beyond included**: Additional usage billed **at pass-through OpenRouter rates** under your project key; we expose per-run cost telemetry so you can monitor spend.

Acceptance Criteria & Warranty

Acceptance criteria (MVP):

- For an agreed suite of test prompts, **≥90%** “pass” in the automated evaluation (maps load successfully; requested layers present with correct defaults).

- **All generated dashboards load in MetX** via manual import or the client **"Load Dashboard" endpoint** once available.
- Default behavior adheres to the **explicit defaults** listed in §3.

Performance targets (informational, not SLAs):

- Initial generation P50 **≤ ~25–30s**, P95 **≤ ~60s** (depends on model and prompt size).
- API availability target **≥99.5%** after stabilization.

Warranty: 30-day bug-fix warranty from acceptance for defects within scoped features.

Pricing & Commercials

Total project value: 25,000 CHF
Already completed (explorations): 6,000 CHF (two projects)
Remaining investment: 19,000 CHF

Payment schedule (remaining 19,000 CHF):

- **30% upfront** (5,700 CHF) at contract signing
- **40% milestone** (7,600 CHF) upon API completion & tested end-to-end generation
- **30% completion** (5,700 CHF) upon deployment & acceptance

Optional support retainer: 2,500 CHF/month

- Up to **10 hours** / month for prompt updates, bug fixes, analytics reviews
- 2-business-day response; quarterly rollover; 15% discount for annual prepay

Client Dependency

- **MetX Frontend "Load Dashboard" endpoint**: An API (or in-app hook) that accepts a validated `metx.json` and renders the dashboard in the current session.

- We will support pilot testing through **manual JSON import** until this endpoint is live.

Next Steps

1. **Confirm scope & acceptance criteria**.

2. **Sign off** on the commercial terms.

3. **Build (Weeks 1–5)** with weekly updates; handover & pilot.

Matthias Heim · matthias@lailix.com · +41 76 268 64 06