White Paper: The LocalFlow Framework

Metadata-First AI: A Frugal, Sovereign, and Reproducible Approach to Data Analysis

RENAUD PAWLAK, LocalFlow, France

Executive Summary

Artificial Intelligence is transforming the way we interact with structured data. However, current *Agentic* models—where a cloud AI continuously processes raw data—are computationally expensive and pose massive risks to data privacy. LocalFlow introduces **Metadata-First AI**: a paradigm where *only metadata crosses the inference boundary*. The LLM receives only structure—column names, data types, statistical samples—never the actual data values. It generates transparent, reusable analytical code that executes locally on the full dataset.

Use cases range from natural language querying of sensitive enterprise datasets ("Talk to your Data" without data exposure) to large-scale geospatial analysis, document intelligence on confidential PDFs, and any analytical pipeline where deterministic, repeatable results matter.

This is a fundamentally new approach at the intersection of **Generative Engineering** and *Local-First* architecture. Instead of using AI to continuously process data, LocalFlow uses it as a one-time "compiler" to write a transparent, reusable analytical formula. This formula then runs securely on your local machine, guaranteeing data sovereignty, explainability, and reproducibility. It drastically reduces cloud AI costs and entirely bypasses heavy database preparation (ETL) pipelines. Furthermore, by utilizing an Analysis Repository as a Retrieval-Augmented Generation (RAG) knowledge base, the system continuously improves its logic without compounding inference costs.

1. The Problem: The Cloud Inference Bottleneck

Modern businesses rely on data intelligence, from financial modeling to renewable energy prospection. While LLMs allow users to query structured data using natural language, real-world deployment faces severe trade-offs:

- **Privacy Risks:** Sending proprietary company data to cloud LLMs violates strict data governance policies.
- **Runaway AI Costs:** Continuously querying an AI to analyze thousands of data points creates a linear explosion in API token costs.
- **The Data Pipeline Burden:** To prevent AI agents from failing during massive API calls, developers are forced to manually download and synchronize heavy static datasets locally, creating a costly ETL maintenance nightmare.

2. The Solution: Metadata-First AI

The central insight of Metadata-First AI is a strict **inference boundary**: the LLM receives only metadata, never raw data values. LocalFlow defines precisely what constitutes metadata for each data type:

- **Structured data** (CSV, Excel, CRM): column headers and statistical samples—enough for the LLM to write correct analysis code. Raw rows are never transmitted.
- **Documents** (PDF): the extracted text structure is shared so the LLM can write a reliable parser. Users can work with obfuscated or template documents to generate formulas, then run them locally on real documents—the LLM only needs the structure, not the actual values.

This positions Metadata-First AI on a distinct axis from the two approaches most commonly discussed, as summarized in Table 1.

Author's Contact Information: Renaud Pawlak, LocalFlow, Paris, France, renaud.pawlak@localflow.fr.

Table 1. Comparison of AI data analysis paradigms

	Cloud AI	Local-model AI	Metadata-First AI
Raw data stays local	×	✓	✓
Metadata stays local	×	✓	○ / ✓[a]
Computation executes locally	×	✓	✓
Uses best-in-class models	✓	Limited	✓
Self-hosted LLM support	✓	✓	✓
Results are deterministic	×	×	✓
Re-runs without AI tokens	×	×	✓
Scales to large datasets	✓	Limited	✓

[a] ○ with cloud LLM (metadata leaves the browser); ✓ with self-hosted LLM (nothing leaves the infrastructure).

It is worth noting that Metadata-First AI is a **complement to classical AI, not a replacement**. Tasks requiring free-form inference over full data content—translation, summarisation, unstructured text understanding—still benefit from classical pipelines. LocalFlow opens up use cases out of reach for classical AI (large-scale analysis without privacy exposure, deterministic and repeatable results, scalable execution at zero marginal AI cost) while leaving room for classical approaches where genuinely needed. The proxy layer can expose LLM-powered tools that operate on a carefully scoped subset of data, extending the boundary in a controlled and auditable way.

As shown in Figure 1, the user simply describes their goal; the LLM translates this intent into standard, executable JavaScript code. **The AI never sees your actual proprietary data.** Once the logic is generated, the AI steps aside. Execution takes place entirely within a secure sandbox on the user's local machine, fetching only the exact data it needs "Just-In-Time".

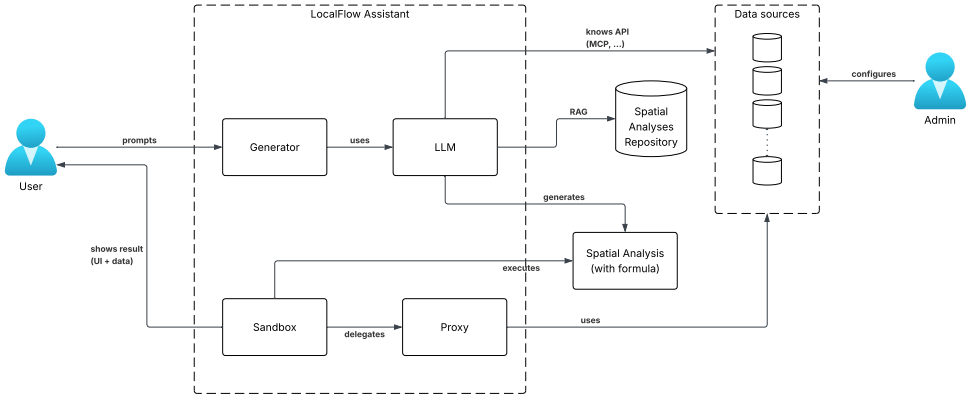

Fig. 1. The LocalFlow Architecture: Separating AI logic generation from secure, local execution.

3. Explainability, Reproducibility, and Continuous RAG Improvement

Transitioning the AI from a continuous decision-maker to a one-time code compiler unlocks several critical engineering properties:

- **Explainability (XAI):** Unlike opaque neural networks that act as "black boxes," LocalFlow uses code as a transparent artifact. By looking at the generated JavaScript, analysts can read, debug, and fully understand exactly what the AI is doing, satisfying strict engineering and regulatory compliance.
- **Reproducibility:** By freezing the AI's probabilistic output into a static script, the execution phase becomes perfectly deterministic. Running the same formula on the same data will always produce identical results, eliminating AI hallucinations from the actual analysis.
- **Continuous Improvement via RAG:** Validated formulas are stored in an **Analysis Repository**. This repository functions as a Retrieval-Augmented Generation (RAG) knowledge base. When a user requests a new analysis, the AI retrieves proven formulas as contextual "anchor examples," organically compounding the collective intelligence of the platform over time.

4. GeoAI Case Study: Photovoltaic Prospection

Consider a company looking for large commercial sites with available roof space and parking lots to install solar panels. Instead of paying an AI agent to analyze 10,000 targets one by one, the user asks LocalFlow to calculate the solar potential. The AI instantly generates the necessary code (Listing 1) to fetch building data and calculate the usable surface area locally.

```
// 1. Fetch Buildings (IGN BDTOPO via WFS)
const urlBat = `https://data.geopf.fr/wfs/ows?SERVICE=WFS&VERSION=2.0.0` +
               `&REQUEST=GetFeature&TYPENAMES=BDTOPO_V3:batiment` +
               `&BBOX=${bbox.join(',')},urn:ogc:def:crs:EPSG::4326&OUTPUTFORMAT=application/
                   json`;
const resBat = await fetch(urlBat).then(r => r.json());

// 2. Fetch Parking Lots (Overpass API)
const opQuery = `[out:json][timeout:25];(way["amenity"="parking"]` +
                `(${bbox[1]},${bbox[0]},${bbox[3]},${bbox[2]}););out body;>;out skel qt;`;
const resOp = await fetch(`https://overpass-api.de/api/interpreter?data=` +
                          encodeURIComponent(opQuery)).then(r => r.json());
// 3. Local Execution: Calculate Intersection Surface Area
let totalBuildingArea = 0;
if (resBat.features) {
    for (const f of resBat.features) {
        const inter = turf.intersect(parcelGeom, f);
// Resolved locally via turf.js
        if (inter) {
            totalBuildingArea += turf.area(inter);
            drawToMap(inter, { fillColor: '#3498db' });
        }
    }
}
```

Listing 1. Listing 1: The AI-generated formula orchestrating local calculations without sending proprietary data to the cloud.

This reusable formula can now be executed across thousands of prospects simultaneously, entirely locally (Figure 2).

Fig. 2. Local execution of the Solar Audit, dynamically calculating areas on the client side.

5. The Numbers: Proof of Frugality

As shown in Table 2, when scaling to a national level (1,000,000 prospects), a realistic Cloud AI Agent requires a heavy upfront download of static data to avoid API timeouts, all while burning massive AI tokens to orchestrate the logic dynamically. LocalFlow accepts a higher "Just-In-Time" network fetch volume in exchange for completely eliminating both the AI token costs and the grueling human data-engineering (ETL) workload.

Table 2. Simulation for Intensive National Use ($N = 1,000,000$ Prospects)

Metric	Scenario A: Cloud AI Agent	Scenario B: LocalFlow
Data Transfer	~300 GB (150 GB static DB sync + 150 GB dynamic API fetches)	~**500 GB** (Just-In-Time API fetching, Zero ETL)
AI Energy	~3,000 Wh (Continuous inference loops)	~**1.5 Wh** (One-time logic generation)
AI Token Cost	~$800.00+	< **$0.10**
Setup/Prep Time	60+ hours (Building ETL pipelines & handling agent timeouts)	~**2 hours** (Managing local batch queuing)

Conclusion

Metadata-First AI establishes a strict inference boundary: the LLM receives only schema and structural metadata—never raw data values—and produces deterministic code that executes entirely on the user's machine. The approach proposed by LocalFlow provides a realistic (frugal and cost-effective) solution for advanced analytical reasoning. By utilizing AI simply to write transparent, reusable tools, businesses can deploy data intelligence at a massive scale. Because the generated JavaScript acts as an open artifact, analysts can audit the code for complete **explainability** and run it deterministically for perfect **reproducibility**. Furthermore, integrating the **Analysis Repository** creates a continuous **RAG** feedback loop that organically improves system capabilities without skyrocketing computing costs. Ultimately, LocalFlow empowers organizations to harness state-of-the-art AI while strictly adhering to the privacy, economic, and ecological mandates of Frugal AI.