CivicSense

AI on Data in Motion

Turning live city data streams into human-friendly guidance, as events happen.

City Alerts are Fragmented, Delayed, and Confusing

CHAOS

ANXIETY

The Parent: Is it safe to send my children to school today?

The Commuter: My train is stopped. How do I get to work on time?

The Senior: What does a ‘Code Red’ alert mean for me?

The problem isn’t a lack of data; it’s a lack of real-time interpretation.

CivicSense Delivers Clear, Actionable Intelligence in Real Time

From fragmented alerts to huma-understandable guidance.

An Architecture Modelled on a Central Nervous System

Confluent Cloud (The Nervous System)

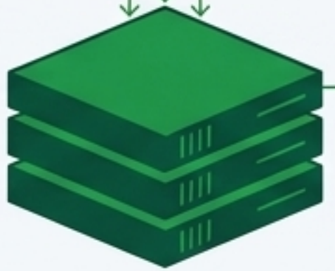
Ingests high-velocity data streams from all city sensors and services. The central channel for all information.

Apache Flink (The Reflexes)

Runs continuous SQL queries on data streams to detect patterns and compute impact signals in real time, *before* the data lands in a database.

Google Gemini AI Agents (The Brain)

A multi-agent system that performs contextual reasoning, assesses impact, and generates clear, persona-specific communication.



MongoDB Atlas Vector Search (The Memory)

Provides long-term knowledge through Retrieval-Augmented Generation (RAG), grounding AI responses in official city protocols and safety guidelines.

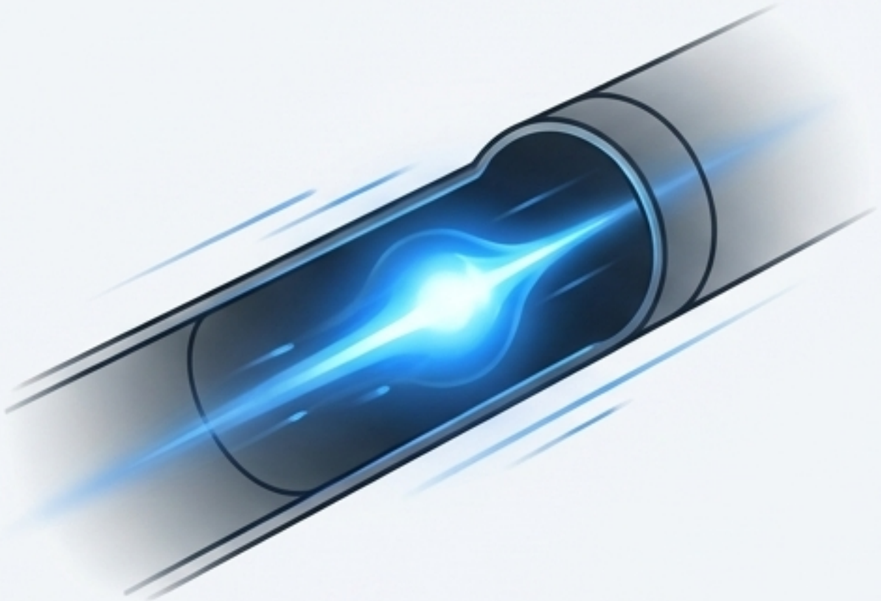
The "WOW" Moment, Part 1: A Question in a Calm City

All Systems Normal

Is it safe to go outside?

Based on current conditions, it is perfectly safe to go outside. There are no active alerts in your area at this time.

The 'WOW' Moment, Part 2: A Critical Event Streams In

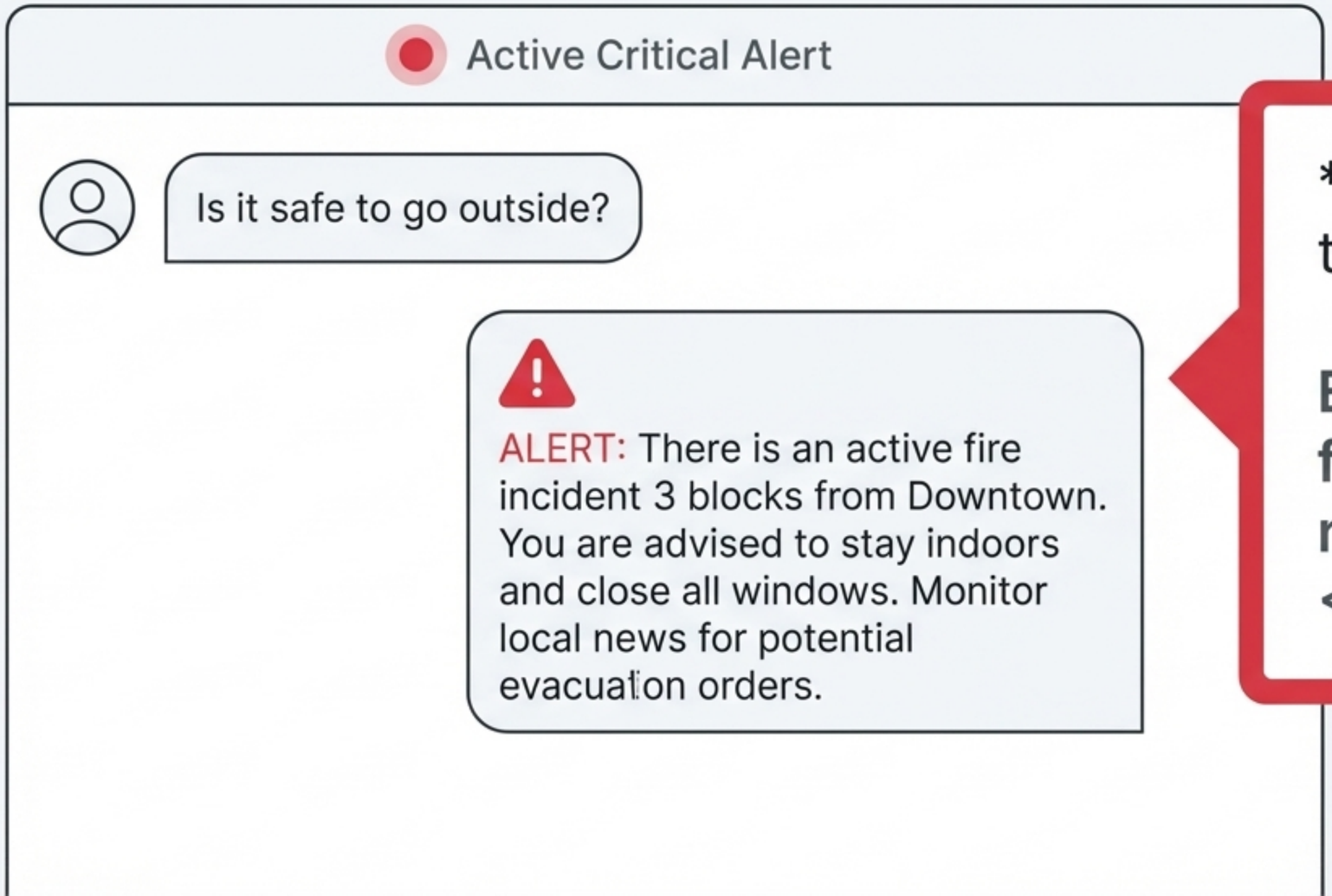


emergency_events

```
{
  "event_id": "F789-2024",
  "event_type": "fire",
  "location": "Downtown",
  "severity": "critical",
  "source": "911_dispatch"
}
```

A new event hits the Kafka topic. Flink processes it. The AI is now aware.

The “WOW” Moment, Part 3: The Same Question, A New Reality



**This is AI reacting to data in motion.

End-to-end latency from event to notification: < 5 seconds.

Inside the Brain: A Multi-Agent AI Processing Pipeline

1. Query Received

User asks, "Why is my train delayed?"

2. Triage Agent

Classifies the query.

`category: transit`,
`urgency: medium`

3. Vector Search (RAG)

Retrieves relevant documents (e.g., transit delay protocols) from MongoDB Atlas.

4. Impact Agent

Assesses severity and scope, enriched by live data from Kafka streams.

`severity: moderate`,
`affected_groups: commuters`

5. Guidance Agent

Synthesises all inputs to generate a clear, persona-adapted response.

6. Response Sent

Delivers the answer via WebSocket.

Monitoring Agent

An asynchronous agent logs all interactions for analytics and auditing without blocking the response path.

The Reflexes: Flink SQL Detects Patterns in the Stream

```sql
SELECT
    TUMBLE_START(event_ts, INTERVAL '5' MINUTE) AS window_start,
    area,
    COUNT(*) AS event_count,
    MAX(severity) AS max_severity
FROM civic_events
GROUP BY
    TUMBLE(event_ts, INTERVAL '5' MINUTE),
    area;
```

True Stream Processing

We run continuous queries on live Kafka streams, not periodic batch jobs.

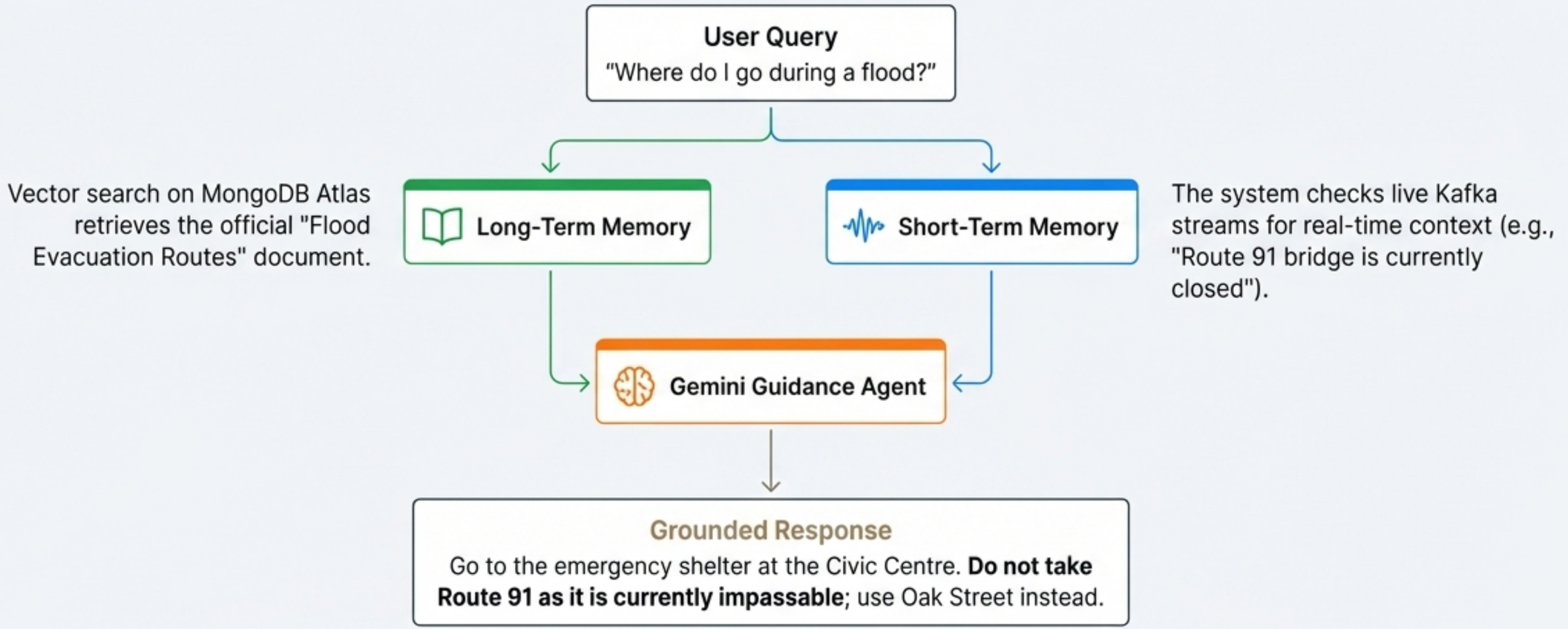
Real-Time Aggregation

5-minute tumbling windows aggregate thousands of raw events into meaningful 'Severity Signals'.

Proactive Detection

This allows us to identify event clusters and severity escalations instantly, powering proactive alerts.

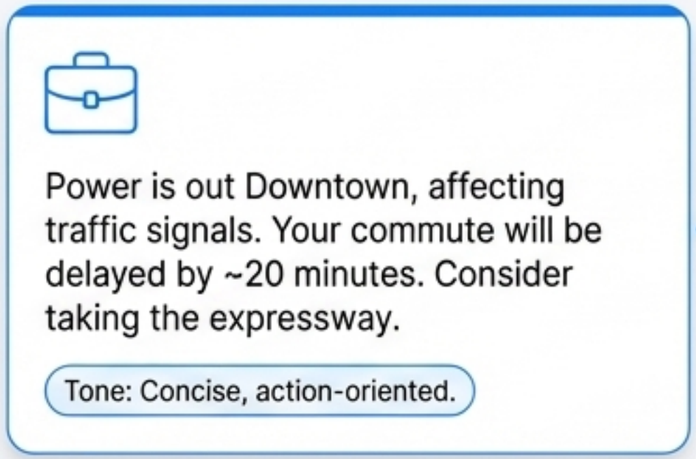
The Memory: Grounding AI in Truth with RAG on Streaming Data

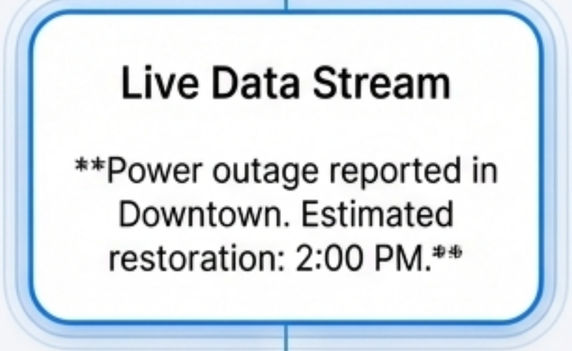


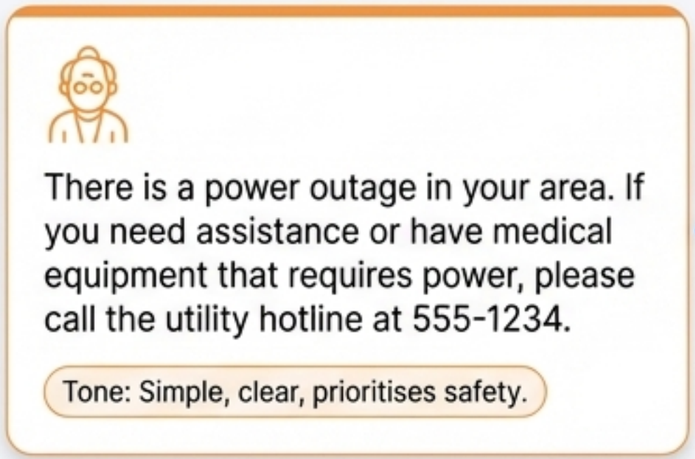
Key Insight

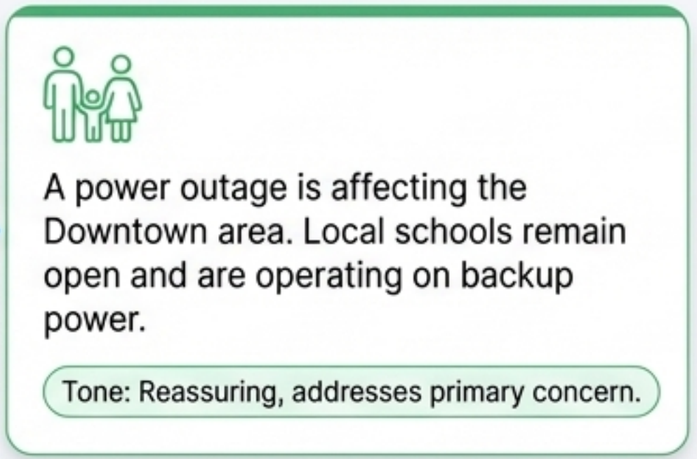
Traditional RAG uses static knowledge. We innovate by augmenting it with real-time context from data in motion, ensuring guidance is always relevant.

Designed for Everyone: Guidance Tailored to Your Needs









Engineered for Production, Not Just a Prototype

Scalability

Built on a serverless, cloud-native stack (GCP Cloud Run, Confluent Cloud, MongoDB Atlas) that auto-scales from zero to millions of users.

Performance

Key performance metrics:

- **< 5s** Event-to-Notification Latency
- **~2.5s** Average Query Response Time
- **1000+** Events/Second Processed

Reliability

Comprehensive error handling, graceful degradation, and automatic WebSocket/Kafka reconnection logic ensure system uptime and availability.

Code Quality

2,000+ lines of production-quality Python and TypeScript, fully containerised with Docker, and supported by 40+ pages of documentation.

Why CivicSense Wins: A Synthesis of Innovation and Impact

AI on Data in Motion

Our entire architecture—from Kafka ingestion and Flink processing to the AI agents and WebSocket delivery—is a live demonstration of this core theme. We don’t simulate; we stream.

Novel Technical Innovation

We introduce a **Multi-Agent Streaming Architecture** and pioneer **RAG on Streaming Data**, a significant step beyond standard implementations.

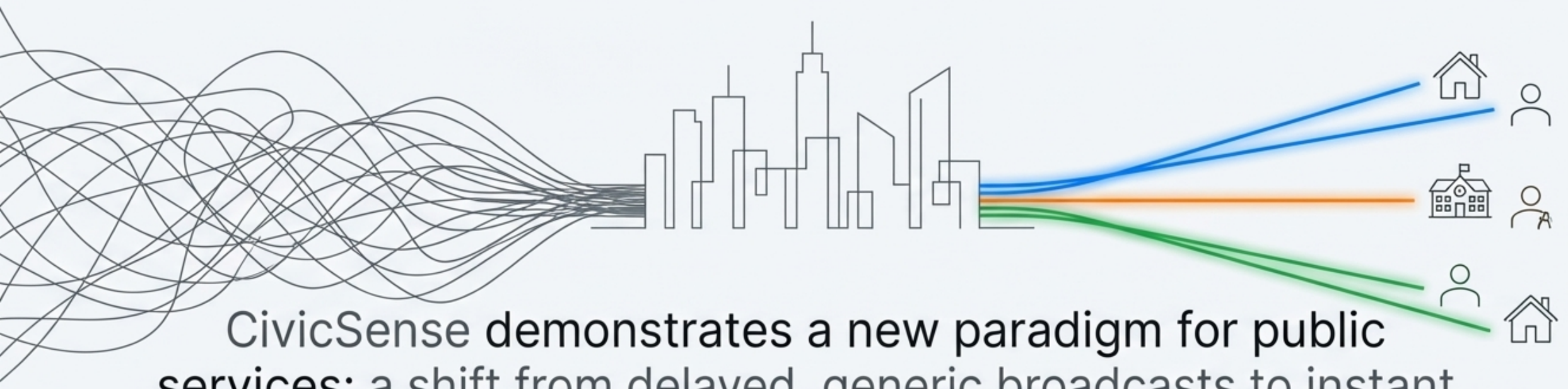
Broad Real-World Impact

This is a scalable ‘public good’ platform. It solves a universal problem for entire populations, with a focus on helping the most vulnerable community members.

Production-Ready Implementation

This is not a concept. It is a fully documented, tested, and deployable system built with enterprise-grade practices, ready to create value immediately.

The Future of Civic Intelligence is Streaming



CivicSense demonstrates a new paradigm for public services: a shift from delayed, generic broadcasts to instant, intelligent, and personalised guidance.

By placing AI directly on data in motion, we can build safer, smarter, and more resilient cities for everyone.