



Virtual Economist

Joanna Nguyen, Ben Nordeman, Kaaviya Senthil, Beamlak Mebrate, Michael Moemeke, Kacy Madueke

Advisor: Jeffrey Segall

Stakeholder: Federal Reserve Bank of Philadelphia

The Problem

Public economic data — housing, income, markets, macro indicators — is abundant and freely available. But for analysts and researchers, extracting meaningful answers still requires engineering work, not economic thinking.

OPERATIONAL FRICTION

Querying data requires technical overhead

Answering "How does Philadelphia's rent burden compare to Austin?" means writing SQL, knowing which dataset to query, and stitching together various datasets

FRAGMENTATION

Housing and market data don't talk to each other

No existing tool combines real estate, demographic, macroeconomic, and stock market data in a single query surface. Cross-domain questions require multiple tools and manual reconciliation.

THE AI TRUST PROBLEM

AI tools can hallucinate data

LLMs confidently generate plausible but unverified figures. For institutional use, a fabricated home value or analyst recommendation is a liability — not a productivity tool.

Who Is It For?

For analysts and researchers who already know the question — but spend their time fighting infrastructure instead of doing economics.

WITHOUT VIRTUAL ECONOMIST

- Comparing 3 cities means 2 hours finding the right Census table, HUD codes, and Zillow series before any analysis starts
- Datasets use different geographic units. Reconciliation is manual and required before anything gets answered
- General-purpose AI gives confident figures that are wrong, uncited, and end up in reports
- Housing and market data live in separate tools. Cross-domain questions cannot be answered in one place

WITH VIRTUAL ECONOMIST

- Ask in plain English. The router picks the right agent instantly
- Live data pulled from Census, HUD, Zillow, Finnhub, and FRED automatically
- Every answer includes the source and exact data used. Fully auditable, no fabrication
- Housing and market intelligence in one session. Two agents, one interface

Our Solution - Virtual Economist

One chat interface. Two specialist agents.
Ask anything — the router decides who answers

Housing and City Agent



- Compare U.S. cities on factors like affordability, income & rent
- Analyze real estate inventory trends
- Live weather & seasonal context
- Interactive heatmap with Zillow + Census data

Unified Agent



Every question passes through an intelligent classifier first. It detects intent, routes to the right agent automatically, and returns a grounded response. Ask about housing and stocks in the same session.

Stock and Market Agent



- Live stock quotes, company profiles
- Analyst buy/hold/sell recommendations
- Sector screening & trend detection
- 90-day price chart + Sharpe ratio, drawdown, stock watchlist

Project Management & Team Process



Meetings & Communication

- Weekly scrum meetings with the Federal Reserve Bank of Philadelphia
- Weekly meetings with advisor Jeffrey Segall
- Requirements, feedback, and priorities reviewed each week



Task & Sprint Management

- JIRA used for all task tracking
- Tasks moved through
 - backlog ->
 - stories in progress ->
 - to do ->
 - in progress ->
 - waiting for review ->
 - waiting for deployment ->
 - done
- Work divided across team members by feature area

Technology Stack

Frontend

- React 18 + TypeScript (strict)
- React Router DOM 7
- Leaflet 1.9 + leaflet.heat

Backend

- Python 3.11 + FastAPI + Uvicorn
- Node.js + Express (auth server)
- psycopg2 (PostgreSQL driver)
- Pydantic v2 request/response schemas
- bcrypt + JWT (jsonwebtoken / PyJWT)

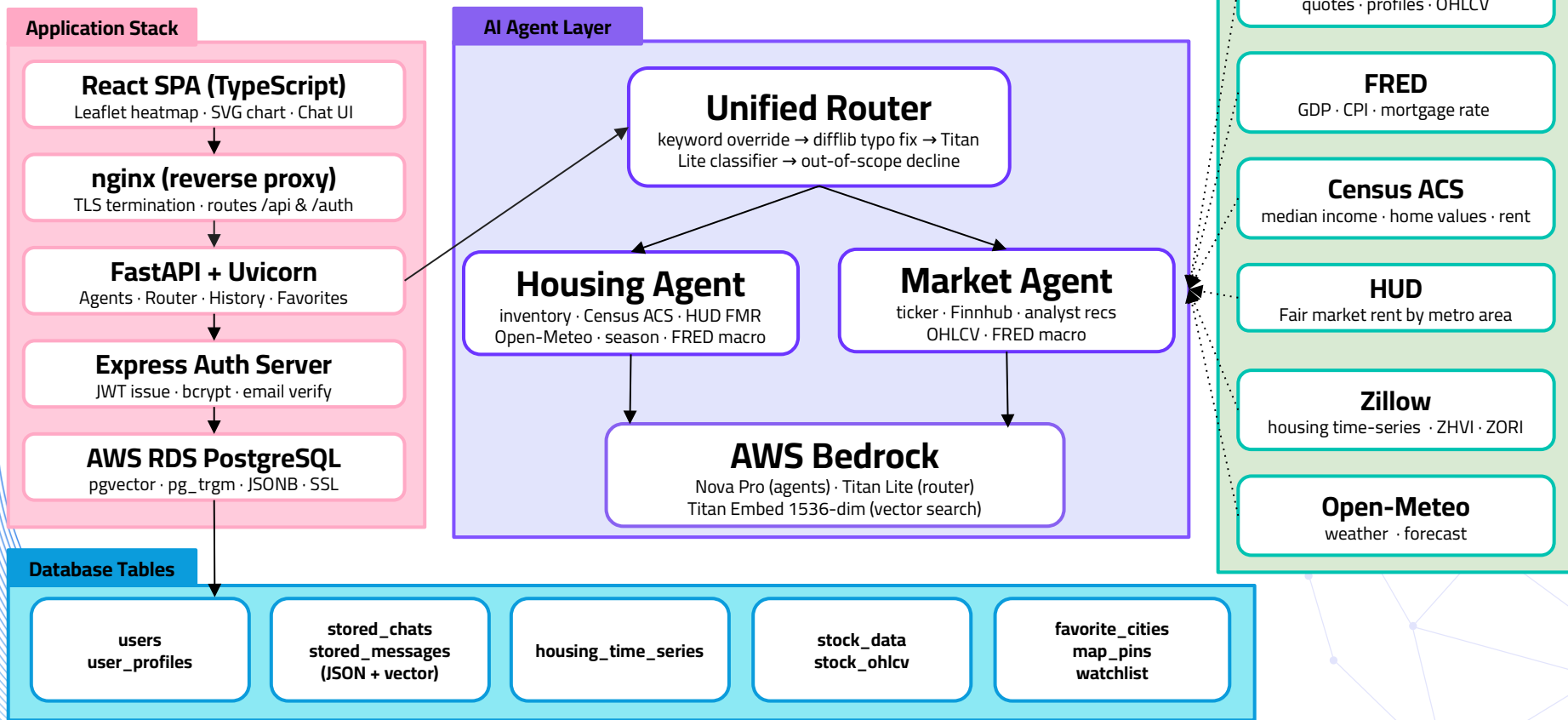
AI / LLM

- AWS Bedrock — Nova Pro (agents)
- Titan Text Lite (router classifier)
- Titan Embed 1536-dim (vector search)
- Tool-use Converse API (custom loop)
- Temperature 0.0 for determinism

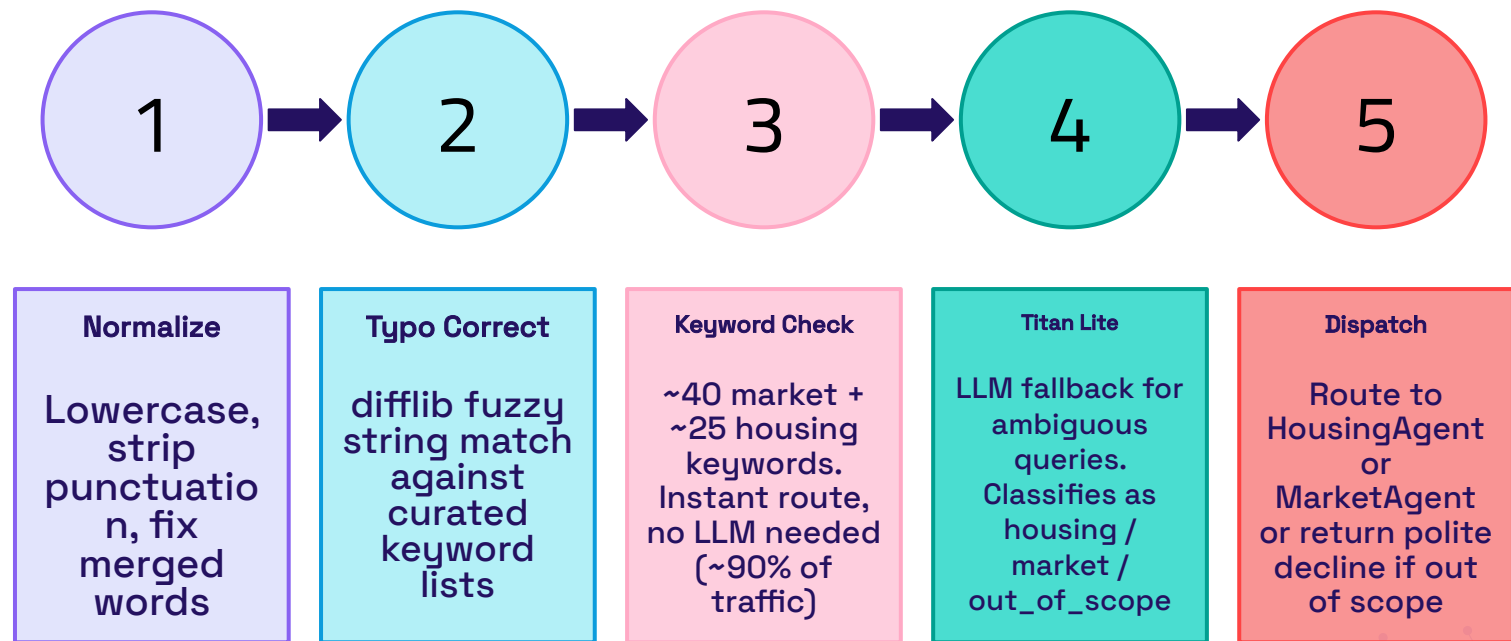
Infra & Data

- AWS EC2 + RDS PostgreSQL
- pgvector + pg_trgm + JSONB
- nginx reverse proxy + TLS
- systemd (2 units, auto restart)
- uv + Ruff + mypy strict + pre-commit

System Architecture



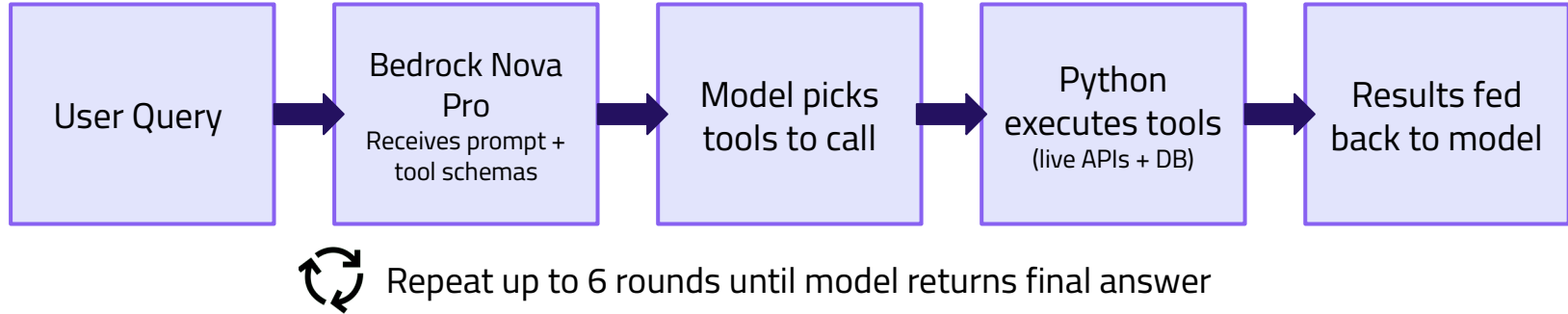
How the Router Works



Key Insight

~90% of traffic is routed by keyword rules before the LLM is ever called, reducing cost and latency significantly.

AI Agents & Tool-Use Loop



Housing Agent Tools

- search_housing_inventory
- get_city_demographics (Census ACS)
- get_fair_market_rent (HUD)

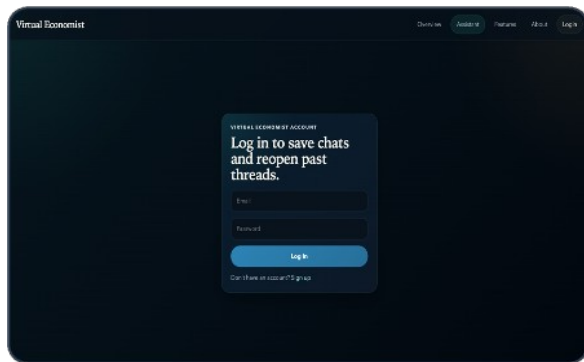
Market Agent Tools

- search_ticker
- get_stock_quote (Finnhub)
- get_company_profile
- get_analyst_recommendations

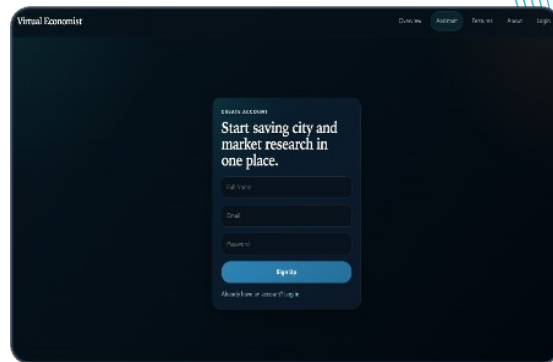
Look & Feel — UI Walkthrough



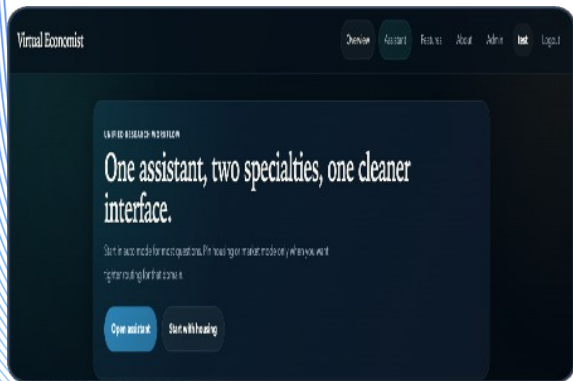
Landing Page



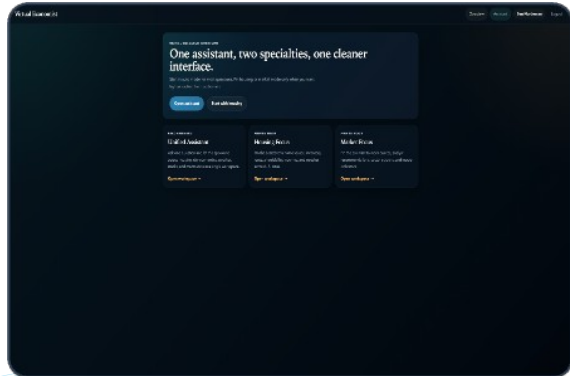
Login



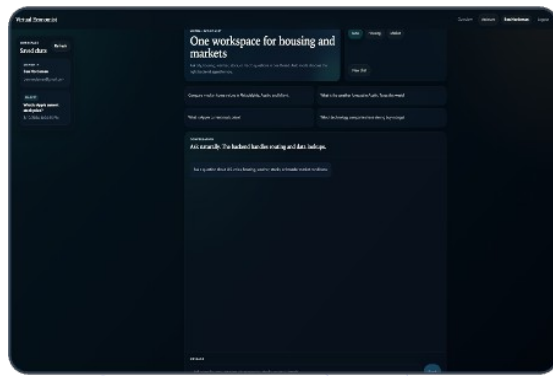
Sign up



Dashboard Hero

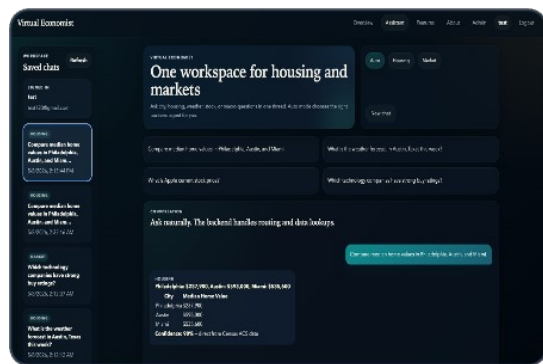


Agent Selection

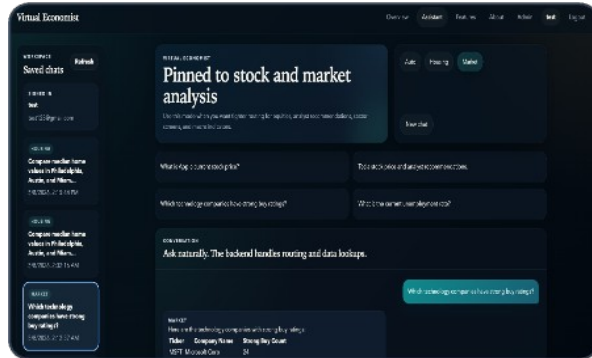


Chat

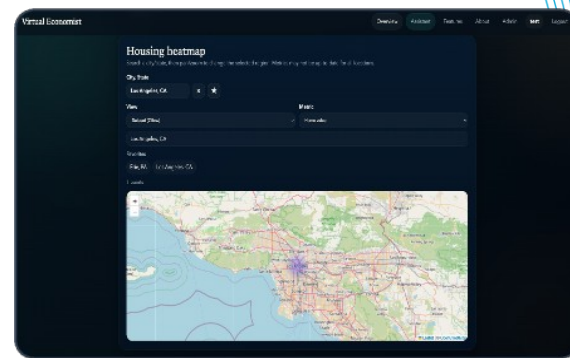
Look & Feel — UI Walkthrough



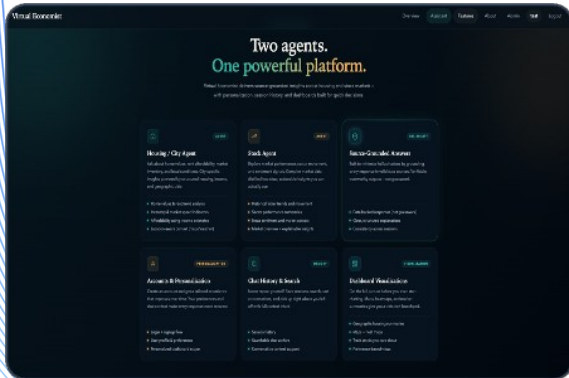
Housing Agent Chat



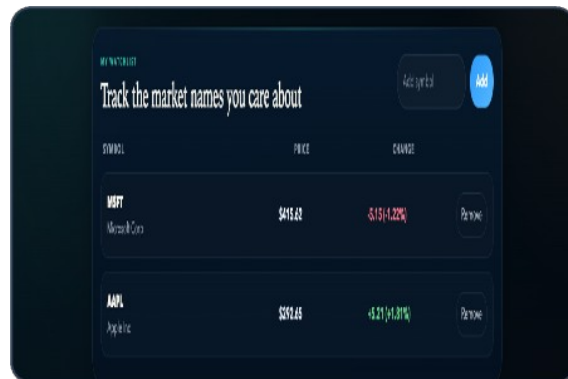
Market Agent Chat



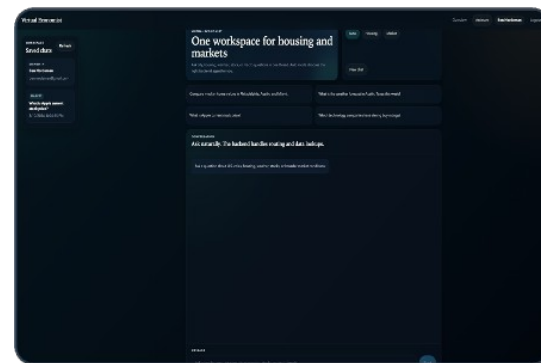
Housing Heatmap



Features Page



Watchlist



User

Technical Challenges

Custom Tool-Use Loop

Built from scratch with no LangChain. Full control over the 6-round loop, tool trace logging, and data type handling.

Quant Analytics

Sharpe ratio, max drawdown, and annualized volatility computed directly in Python, never generated by the LLM.

Typo-Tolerant Router

Handles misspelled and merged queries like "apple sotkc price" and routes them correctly without hitting the LLM.

Dual-Source Heatmap

Zillow and Census ACS overlaid simultaneously using two separate heat layers. Fetches data automatically on map pan and zoom.

Hybrid Chat History

Chat history stored in the database with automatic localStorage fallback if the server is unavailable, protecting the UI.

Data Visibility — Dev & Testing Tools

tool_trace

Every API response includes an ordered list of tools called, inputs, status, and output previews.

sql_used field

The literal SQL executed against PostgreSQL is returned to the client and stored in `stored_messages.metadata` (JSONB). Fully auditable chat history.

Swagger / ReDoc

FastAPI auto-generates live interactive API docs at `/docs` and `/redoc`. Full schema, request/response examples, and try-it-out for every endpoint.

Heatmap as Debug Tool

Pan the map, watch the debounced bbox SQL refetch in real time. Switch metric or source to inspect data layers.

Security — Built In Throughout

1. Network

nginx sole public entry · FastAPI and Express on 127.0.0.1 · HTTPS only · RDS SSL · verify-full

2. Authentication

JWT HS256 · secret via AWS Secrets Manager · 24hr expiry · 401 on invalid

3. Authorization

RBAC per request · role fetched from DB every call · 3 roles · admin self-demotion blocked

4. Password and Email

bcrypt constant-time · email verify gate · SHA-256 token · 24hr expiry

5. SQL Safety

SELECT-ONLY enforcement · psycopg2 parameterized binding · regex identifier sanitizer

Why it matters

No token, no history

Chat endpoints work without auth. But stored history, watchlists, and saved cities require a valid JWT. Public access without data exposure.

Role changes are instant

Role is re-fetched from the DB on every single request. No caching. A compromised or demoted user loses access immediately.

LLM cannot write to DB

SELECT-only enforcement. Any non-SELECT query raises an error immediately.

Secrets never in code

JWT secret and DB password pulled from AWS Secrets Manager at runtime. Never hardcoded, never in git.

Alternative Design Decisions

Decision Area	Considerations	What We Chose	Why We Chose It
Agent Framework	LangChain	Custom Base Agent	Full control, easy to unit test
Mitigating API calls and screening data	Live API call every time	Local snapshot table + periodic sync	Screening is faster, avoids rate limits and latency.
Permissions	Same endpoints for everyone	RBAC + role-gated endpoints	Prevents unauthorized access to agent features and user data
LLM Provider	OpenAI GPT-4 / Anthropic API	AWS Bedrock (Nova Pro)	One IAM trust boundary
Live vs Stored Data	Only dataset or live APIs	Both	Diversify data and accommodate data sources

Testing Strategy

What we test

Routing correctness — keyword classification, typo tolerance, city aliases, out-of-scope fallback

Agent tool loop — tool_use → end_turn cycle, trace capture, SQL, and chart payload

Live API tools — housing, market, and stock data tools across all agents

Data shaping — recommendation normalization, snapshot metadata, OHLCV risk metrics

How we test it

Fully offline — all external APIs mocked, no network calls or LLM cost

SQL assertions — fake DB cursor verifies table names, WHERE clauses, and bindings

Quality gates — Ruff, mypy strict, and pre-commit hooks on every commit

Manual testing — live UI flows with tool_trace and sql_used for real-time verification

Coverage — focused on high-risk paths: routing, tool dispatch, and API retry logic