AST SECURITY (SQLGLOT) | LEAST-PRIVILEGE MYSQL 8.4 | BIFURCATED DATA PLANE | 95.5% TOKEN COLLAPSE

Enterprise NL-to-SQL & Generative UI

Deterministic Relational Querying with AST Guardrails & Out-of-Band Data Transport. An engineering retrospective on transforming conversational AI into a hardened, high-throughput microservice.

AUTHOR

DON MARTIREZ // PRINCIPAL DESIGN TECHNOLOGIST & SYSTEMS ARCHITECT

COVERAGE
45 Financial Tables

ACCURACY
Zero Hallucination

The Data Paradox in Enterprise AI

FAILURE MODE 01

Arithmetic Hallucination

When asked for financial rankings or margins, LLMs synthesize approximated numbers from internal neural weights rather than reading live tape.

FAILURE MODE 02

Security & Corruption Risk

Prompt jailbreaks and poorly sanitized SQL generators risk accidental mutations, causing database locking and catastrophic schema corruption.

FAILURE MODE 03

Context Window Poisoning

Naive implementations stream raw rows back as Markdown text, burning 10,000+ tokens per query and exhausting session context.

FOUNDATIONAL ARCHITECTURAL PRINCIPLE

"Math calculates deterministically, databases store ground truth, and LLMs only translate intent."

Balancing Operator Velocity with Enterprise Auditability

OPERATOR ARCHETYPE

Elena Vance

VP of Commercial Strategy (Decision Maker)

Needs ad-hoc market capitalization comparisons, revenue rankings, and sector trends on the fly during executive reviews. Zero SQL familiarity.

MANDATORY UI CONTRACT:

- Dynamic part-to-whole donut charts with accurate remainder buckets
- Zero raw code jargon; instant visual hierarchy
- One-click CSV export for executive leadership

AUDITOR ARCHETYPE

David Kross

Principal Infrastructure Auditor (Security Lead)

Distrusts black-box LLM systems. Demands strict database isolation, immutable production schemas, and full mathematical audit trails.

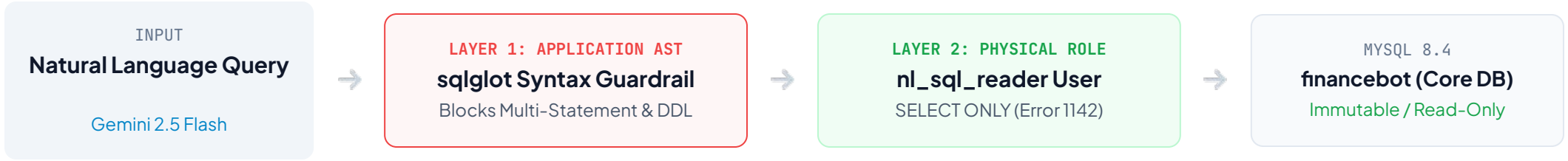
MANDATORY AUDIT CONTRACT:

- Transparent SQL syntax viewer displaying exact queries
- Real-time database execution latency telemetry (in ms)
- Zero write privileges; 100% AST mutation interception

STAGE 03 // SECURITY ARCHITECTURE

Zero-Mutation Guarantee & Database Sandboxing

DUAL-LAYER ACCESS & SCHEMA SANDBOXING TOPOLOGY

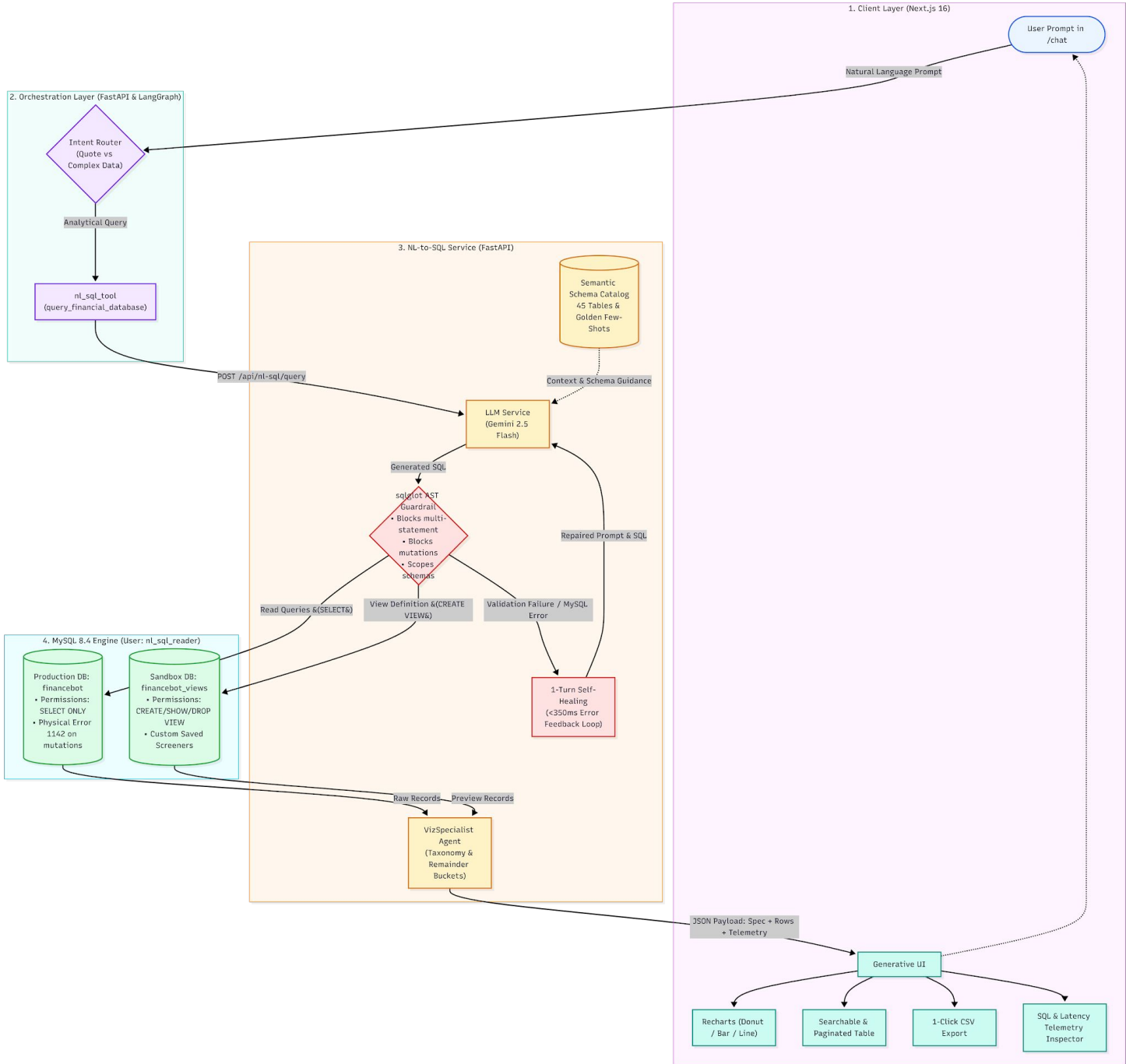


01 The AST Inspection Engine (sqlglot)

Parses raw SQL into syntax nodes prior to execution. If multiple statements are detected (blocking semicolon injection) or DDL instructions appear, execution aborts.

02 Schema View Isolation (financebot_views)

When users generate custom market screeners via POST /create-view, the AST forces namespace scoping into financebot_views, protecting core tables from lock contention.


1. Client Layer (Next.js 16)
User Prompt in /chat
Natural Language Prompt
2. Orchestration Layer (FastAPI & LangGraph)
Intent Router (Quote vs Complex Data)
Analytical Query
nl_sql_tool (query_financial_database)
POST /api/nl-sql/query
3. NL-to-SQL Service (FastAPI)
Semantic Schema Catalog 45 Tables & Golden Few-Shots
Context & Schema Guidance
LLM Service (Gemini 2.5 Flash)
Generated SQL
sqlglot AST Guardrail • Blocks multi-statement • Blocks mutations • Scopes schemas
Repaired Prompt & SQL
Read Queries &(SELECT&)
View Definition &(CREATE VIEW&)
Validation Failure / MySQL Error
1-Turn Self-Healing (<350ms Error Feedback Loop)
4. MySQL 8.4 Engine (User: nl_sql_reader)
Production DB: financebot • Permissions: SELECT ONLY • Physical Error 1142 on mutations
Sandbox DB: financebot_views • Permissions: CREATE/SHOW/DROP VIEW • Custom Saved Screeners
Raw Records
Preview Records
VizSpecialist Agent (Taxonomy & Remainder Buckets)
JSON Payload: Spec + Rows + Telemetry
Generative UI
Recharts (Donut / Bar / Line)
Searchable & Paginated Table
1-Click CSV Export
SQL & Latency Telemetry Inspector

The Bifurcated Data Pipe: Separating Control from Data

CHANNEL A: CONTROL PLANE

~350 TOKENS

Executive Briefing Synthesizer

• **Hard Row Clamp:** Receives a maximum sample of top 25 rows for linguistic synthesis.

• **Concise Narrative:** Synthesizes a 2–sentence conversational briefing for the chat stream.

• **Firestore Storage:** Only the 40–token summary is stored, leaving context pristine.

> *"Over the last 6 months, Netflix gained +24% while Apple and Google traded in range..."*

CHANNEL B: DATA PLANE

OUT-OF-BAND

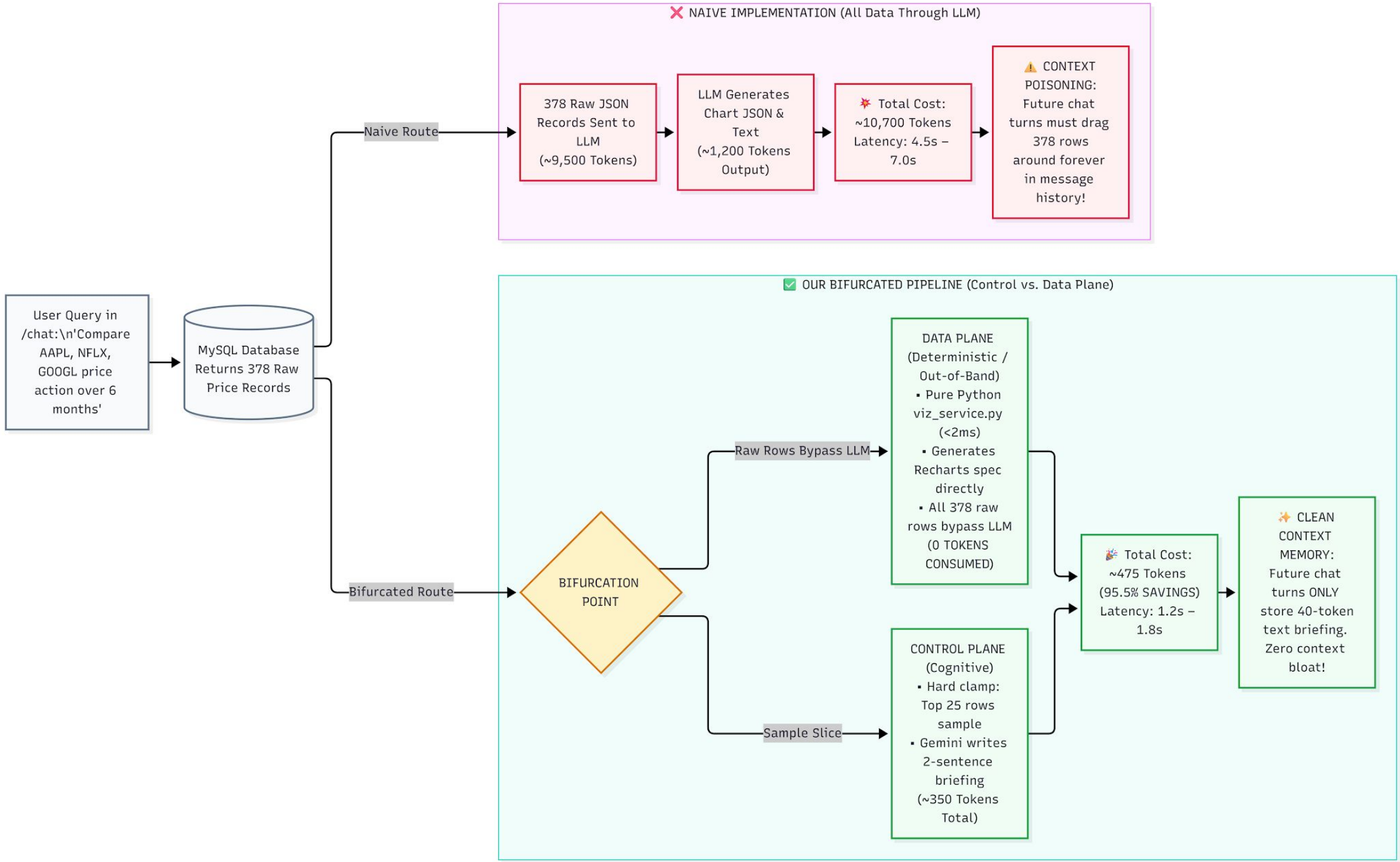
Deterministic Visuals & Payload

• **Pure Python viz_service.py:** Formats Recharts multi-line chart specs in <2 milliseconds.

• **All 378 Raw Rows:** Transported out-of-band directly to client payload with zero LLM overhead.

• **Direct React Hydration:** Searchable tables and 1–click CSV export rendered locally.

```
payload.data { chart_type: "line", rows: 378 }
```


❌ NAIVE IMPLEMENTATION (All Data Through LLM)
Naive Route
378 Raw JSON Records Sent to LLM (~9,500 Tokens)
LLM Generates Chart JSON & Text (~1,200 Tokens Output)
💥 Total Cost: ~10,700 Tokens Latency: 4.5s – 7.0s
⚠️ CONTEXT POISONING: Future chat turns must drag 378 rows around forever in message history!
User Query in /chat:\n'Compare AAPL, NFLX, GOOGL price action over 6 months'
MySQL Database Returns 378 Raw Price Records
✅ OUR BIFURCATED PIPELINE (Control vs. Data Plane)
Bifurcated Route
BIFURCATION POINT
Raw Rows Bypass LLM
DATA PLANE (Deterministic / Out-of-Band)
▪ Pure Python viz_service.py (<2ms)
▪ Generates Recharts spec directly
▪ All 378 raw rows bypass LLM (0 TOKENS CONSUMED)
Sample Slice
CONTROL PLANE (Cognitive)
▪ Hard clamp: Top 25 rows sample
▪ Gemini writes 2-sentence briefing (~350 Tokens Total)
🎉 Total Cost: ~475 Tokens (95.5% SAVINGS) Latency: 1.2s – 1.8s
✨ CLEAN CONTEXT MEMORY: Future chat turns ONLY store 40-token text briefing. Zero context bloat!

Head-to-Head Architectural Comparison (378 Rows)

NAIVE LLM PASS-THROUGH

Standard Industry Approach

Prompt & Row Stream Token Cost:	~9,500 Tokens
Chart JSON Output Generation:	~1,200 Tokens
Total Token Burn per Turn:	~10,700 Tokens
End-to-End Latency:	4.5s – 7.0s

Result: Context poisoning. Rows remain trapped in memory; future chat turns are throttled.

OUR BIFURCATED PIPELINE

Deterministic Data Plane

SQL Generation (Gemini 2.5 Flash):	~85 Tokens
Deterministic Chart Build (viz_service.py):	0 Tokens (<2ms)
Total Token Burn per Turn:	~475 Tokens (95.5% Saved)
End-to-End Latency:	1.2s – 1.8s

Advantage: Pristine context memory. Future turns pass only the 40-token summary.

Verified Production Metrics & Generative UI Contract

100%
Security Validation
Mutations intercepted before transmission

<350ms
Recovery Speed
Automated self-healing loop

1.84ms
Chart Build Speed
$0 LLM token cost locally

45 Tables
Schema Coverage
Relational equity data & balance sheets

1. Automated Remainder Buckets

For part-to-whole questions, the engine queries total market capitalization and calculates a "Rest of Tracked Universe" bucket so donut slices represent an authentic 100%.

2. Searchable, Sortable Table

Renders paginated records with formatted currencies ($3.45T, 18.2× P/E), equipped with real-time text filtering across all returned attributes.

3. One-Click CSV & SQL Telemetry

Complete transparency: operators download raw CSV datasets for spreadsheet modeling or view exact SQL statements and query latency.

Architectural Reusability Across the Ecosystem

PRODUCTION CLIENT 01

Portfolio Chat (/chat)

Conversational Systems Demo

Demonstrates real-time tool calling and Generative UI hydration to technical recruiters and system architects in an open, ungated environment.

PRODUCTION CLIENT 02

3PM Daily (Market Terminal)

Quantitative Institutional Intelligence

Powers analytical terminals for financial operators. Translates complex cross-table inquiries into instant, validated data cards with Recharts.

ARCHITECTURAL RETROSPECTIVE: THE POWER OF BOUNDARIES

Dismissing NL-to-SQL as an uncontrollable hazard was a failure of boundary definition, not model capability. By placing math in SQL/Python, enforcing pre-flight AST validation, and isolating data transport out-of-band, we unlocked conversational relational intelligence that is robust, auditable, and cost-efficient at enterprise scale.