Taming the “Verbatim” Chaos

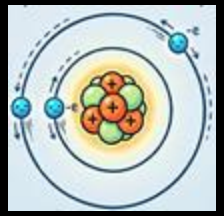
Designing Production-Ready NLP and Multi-Agent Pipelines

June 14, 2026

Sarang Brahme
Principal Architect

About Me

Sarang Brahme



Almost 2 Decades in the Tech Trenches
A Principal AI Architect, Technical Product Owner/Leader who turns big ideas into real, working AI roadmaps.

C-Suite and VC Translator
I speak fluent "Business Value," "Data Science," and "Software Engineer"—and help them all get along.

Current Obsession
Agentic AI, MLOps, building/advising startups. If it makes life easier intelligently, I'm building it.

Tech Stack of Choice
Azure at the moment, but also - Databricks, Fabric, GCP, Open Source

Speaker & Practitioner
I do talk about AI on stages; and I actually build the tech when I get off stage.

Warning
Once started, I can't stop talking about agentic AI, quantum computing, heavy metal, and philosophy.

AGENDA

01 BACKGROUND

- Who is this session for?
- What we mean by Verbatim ?
- What's wrong with standard APIs?
- What enterprises actually need

02 FOUNDATIONS

- Reference Architecture on Azure
- Data Flow
- Sequence of Operations

03 SEMANTIC INTELLIGENCE

- NLP Core
- Objective Mapping

04 AGENTIC DECISION LAYER

- Why Agents? From Analysis to Actions
- Agent Framework
- Designing the Agent Swarm
- How Agents choose - Decision Matrix
- External Signals

05 MAKIN' IT REAL - PRODUCTION

- Production Considerations
- Cost Optimization

06 CLOSING

- Recap and Resources
- Questions
- Thank You

01

Background

01: Who is this session for ?

ML/AI Engineers

Building NLP pipelines beyond off-the-shelf models

Data/Platform Engineers

Designing ingestion, lakehouse, and storage architecture

Solution Architects

Evaluating Azure Fabric, AI Foundry, and multi-agent orchestration

Product/CX Platform Owners

Managing analytics platforms that surface VoC insights to the business

02: What we mean by ‘verbatim’

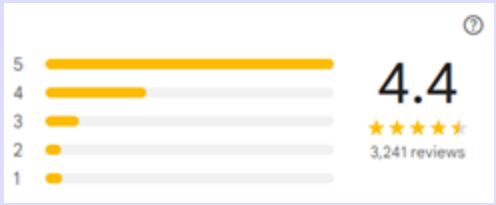
verbatim

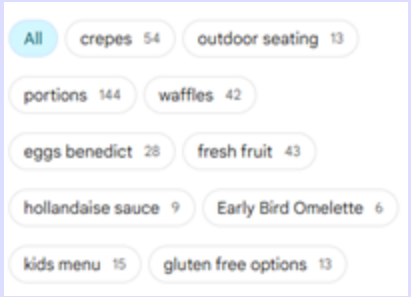
[ver-**bey**-tim] / vərˈbeɪ tɪm /

The raw, unedited free-text a customer actually wrote (app review, survey open-end, support transcript, social post)

- High value, high entropy: multilingual, misspelled, emoji-laden, sarcastic, multi-topic, variable length
- It is the single richest and least-structured asset in the customer dataset

Venue Rating





Verbatim example # 1

Came here because of the high rating. It was busy, so the delay in greeting was understandable. We ordered two omelettes, but the wait was long—about 20–25 minutes for the first, and another 4–5 minutes for the second.

Food tasted good, but the plates looked quite worn and marked. Overall, decent food but service timing and presentation could be better.

Verbatim example # 2

If you look at the menu, the theme of the 90s is there, but in decor or in the music, you can not feel it.
My experience started that I got an "outsider" table. It is a table where basically coming a border in service between 2 waitress and you hidden next to the wall. So, one person is not paying attention to you because it is not her table, and your waitress simply can not see you. I'm not much picky about it. Hence, I was okay with my book.

Food | **Service** | **Ambience** | **Overall** | **Parking** | **.....**

03: What's wrong with standard APIs ?

"The pizza was amazing, genuinely the best in town — but it showed up 40 minutes late and stone cold."

Standard API

`Sentiment: "mixed"` — not actionable

Business Need

Product, Logistics, & Support need different granular signals from the same text.

Result

Coarse sentiment creates downstream chaos — teams can't prioritize, route, or automate.

04: What Enterprises Actually Need

01. Deep Decomposition

Decompose each verbatim into (aspect, opinion, sentiment, confidence) tuples.

02. Strategic Mapping

Map each aspect to a granular business objective with a clear owner and KPI.

03. Structured Persistence

Persist in a queryable, joinable relational store — not just a data lake of JSON.

04. Contextual Enrichment

Enrich with real-world context so attribution is correct.

05. Orchestration & Governance

Coordinate the work across specialized agents, with governance and auditability.

Reframe:

Old framing: How positive is this text?

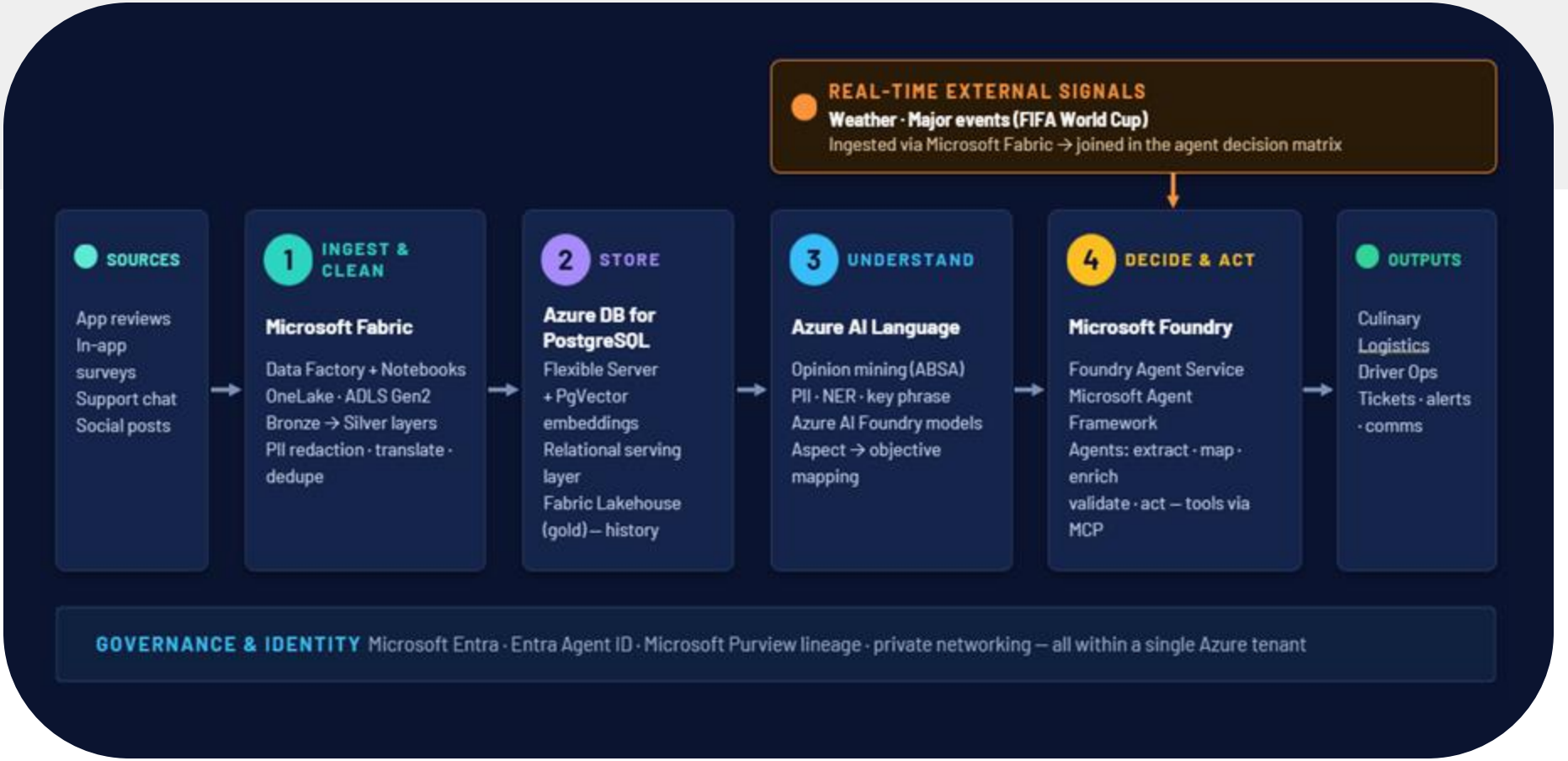
New framing: Which 'objective' is affected, in which direction, with what confidence, and what should happen next?

Output we want isn't a score — it's a routed, contextualized, auditable signal

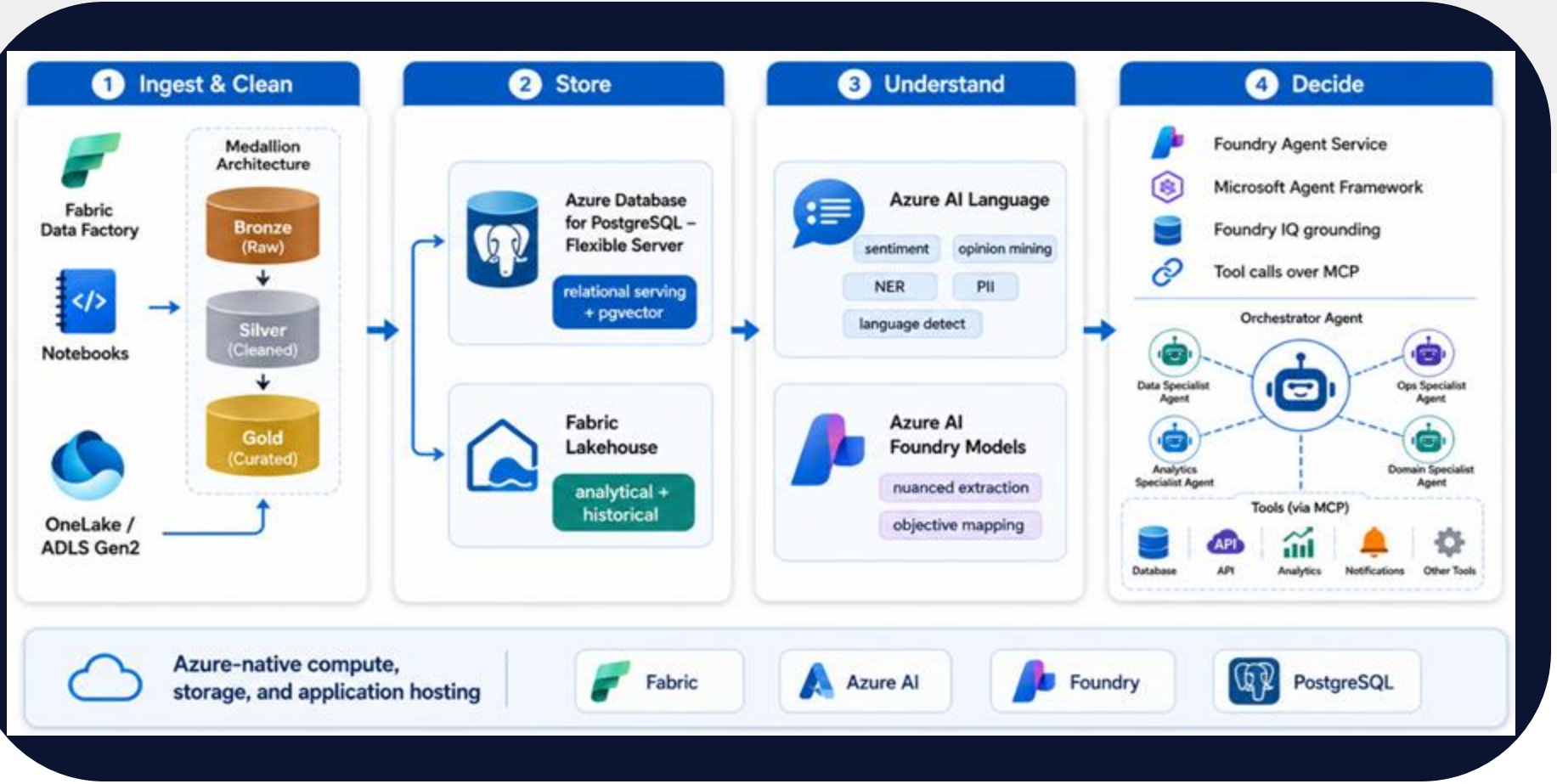
02

Foundations

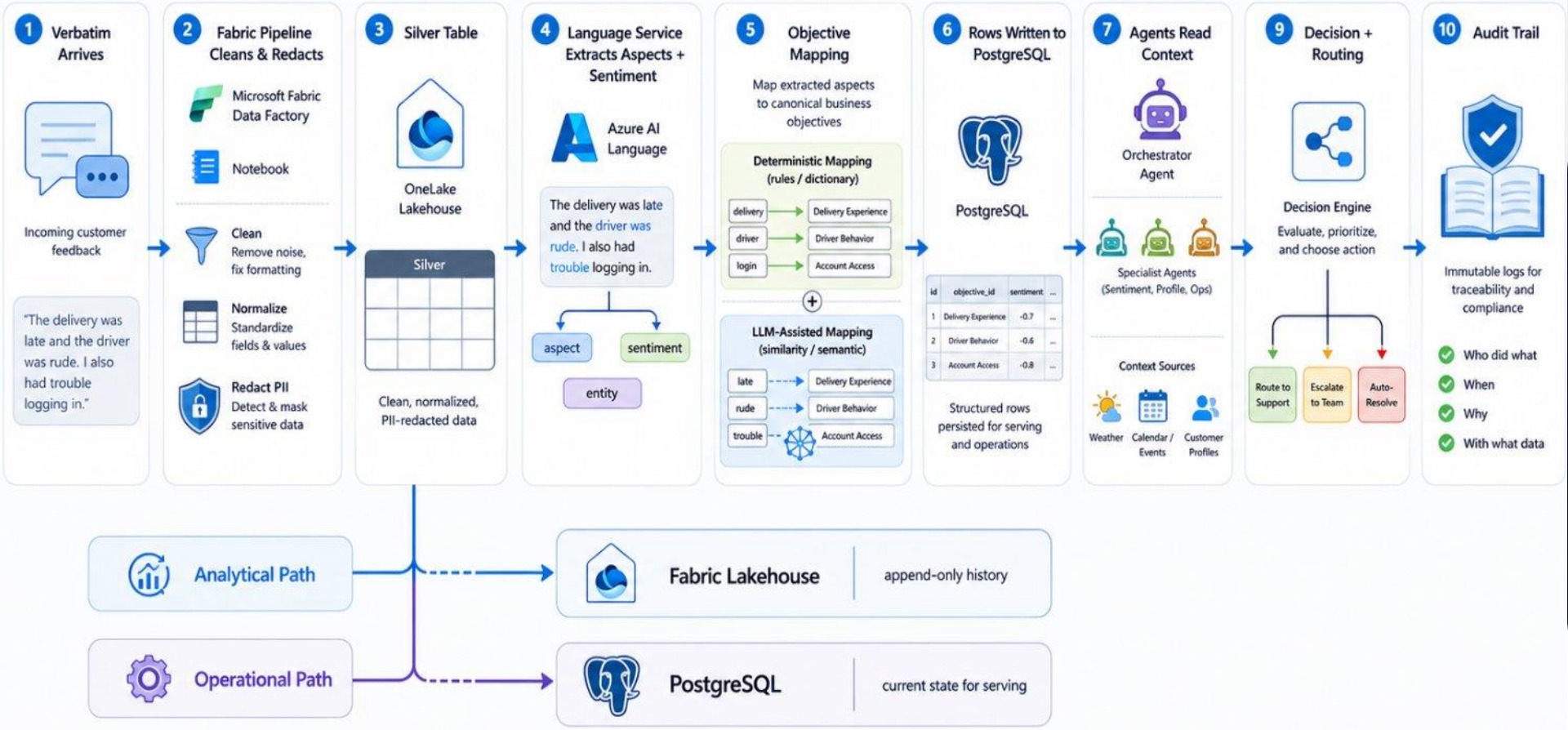
05: Reference Architecture on Azure



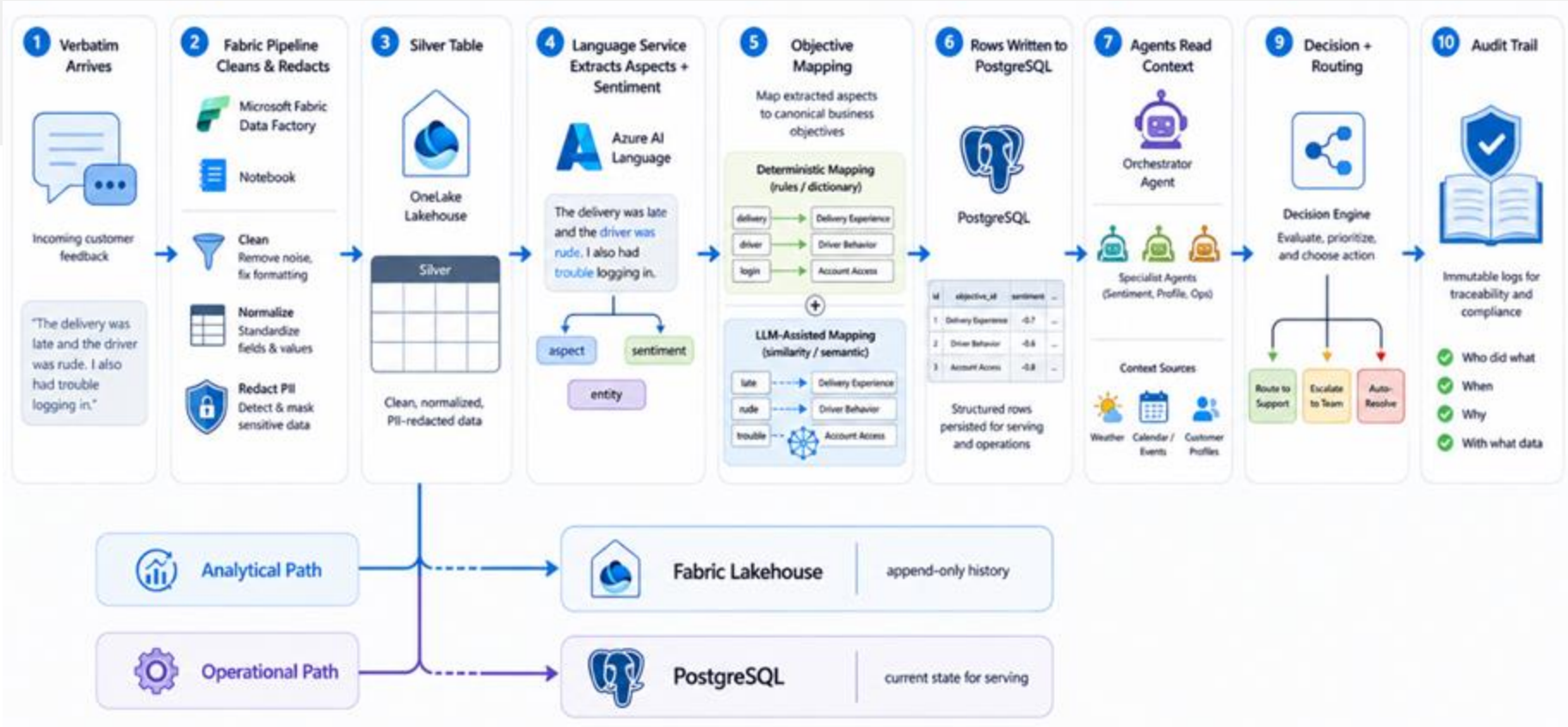
06: Engineering Diagram



07: Data Flow Sequence



07: Data Flow Sequence



03

Semantic Intelligence

08: Semantic Objective Mapping - NLP Core

Step 1: ABSA (Aspect-Based Sentiment Analysis)

- Extracts specific features or "aspects" from text.
- Determines sentiment (positive, negative, or neutral).
- Surfaces target (noun/verb) and assessment (opinion word).

"The food was great but the staff was unfriendly"

Aspect: **food** = **positive**
Aspect: **staff** = **negative**
Taxonomy is a living asset, adapting to slang and failure modes.

Step 2: Business Objective Taxonomy

A verbatim aspect is not yet a business objective. It must be mapped to corporate goals.

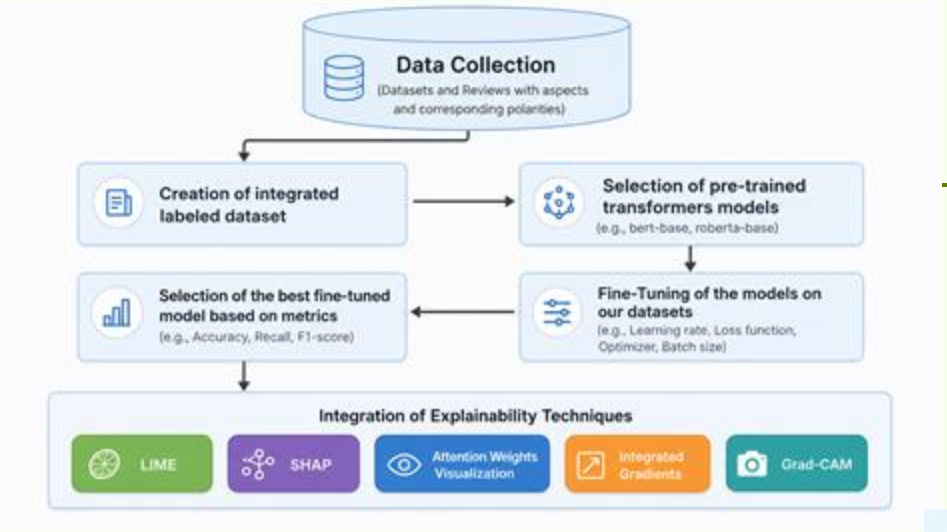
Taxonomy Hierarchy:
Objective → Parent → KPI → Owning Team

Example: Service → Attentiveness → Promptness Score → Customer Success

Technical Implementation:

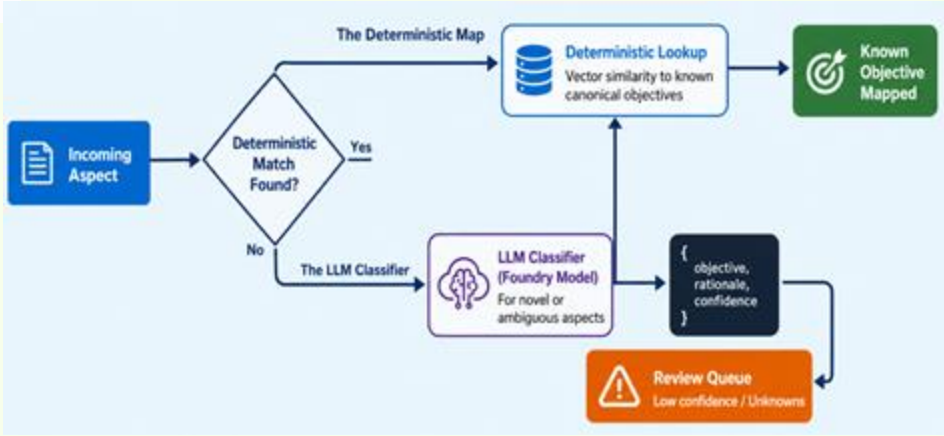
- Stored as versioned data assets.
- `objective` + `aspect_objective_map`
- Not hardcoded in LLM prompts.

09: Semantic Objective Mapping - NLP Core

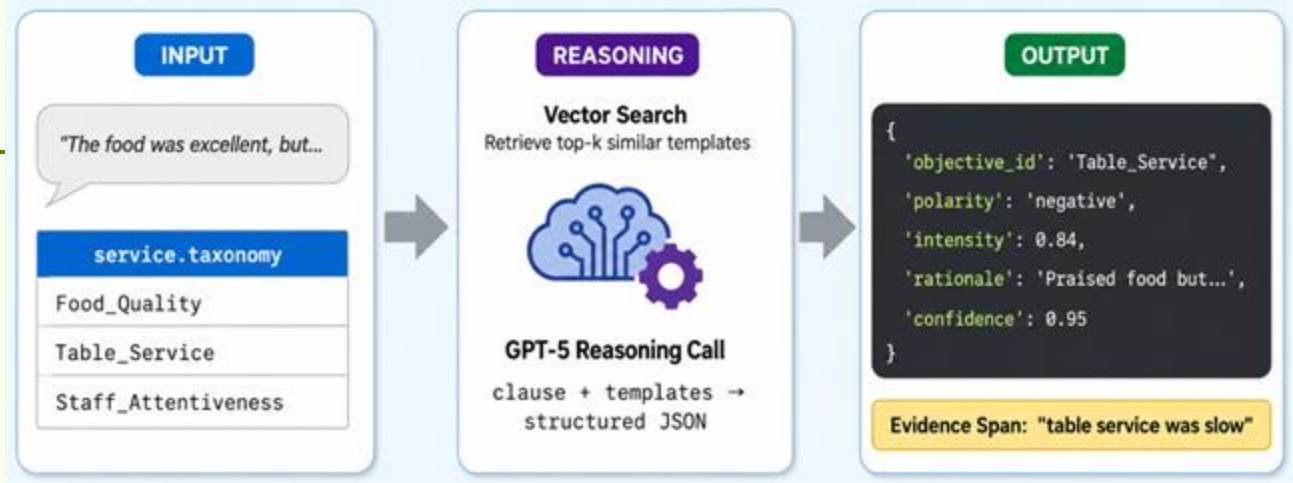


Aspect-Based Sentiment Analysis

Complementary mechanisms -
Mapping Aspects -> Objectives



10: Semantic Objective Mapping - NLP Core



- Objective taxonomy is maintained as a master table in Postgres
  - Each objective has a human-readable definition + few-shot example clauses
- Mapping logic (Azure AI Foundry-hosted):
  - Retrieve top-k similar objective templates via vector search
  - Feed clause + retrieved templates into a GPT-4o / GPT-5 reasoning call
- Structured output: JSON with objective, polarity, intensity, rationale

11: Multi-Sentence Complexity

Challenge: A single verbatim contains multiple, sometimes contradictory, signals

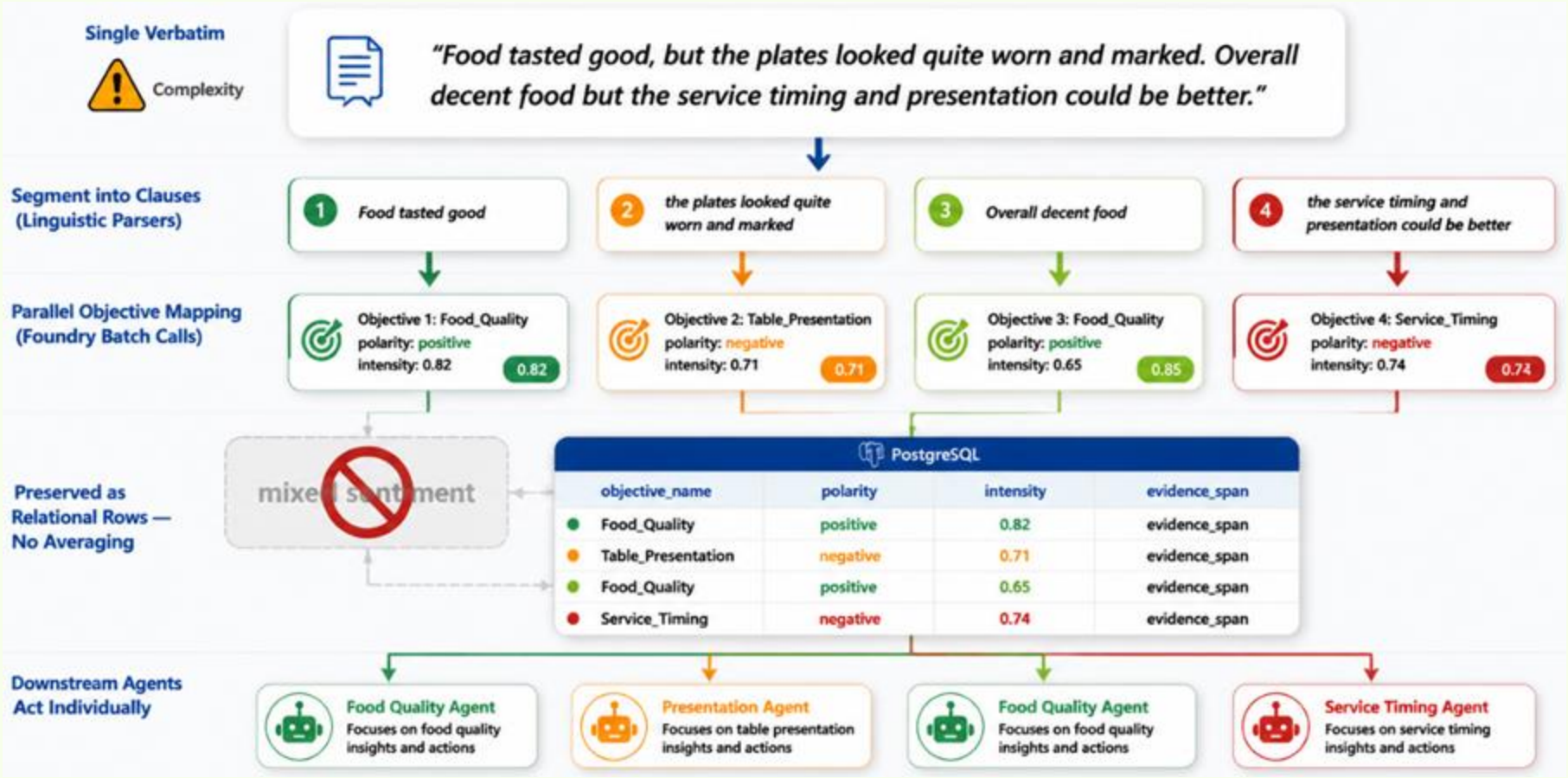
Approach:

- Segment into clauses using linguistic parsers
- Run objective mapping per clause (parallel batch calls to Foundry)
- Aggregate at the verbatim level with conflict resolution rules

Example: If Product_Quality is positive (0.85) but Support_Efficiency is negative (0.72), both records are preserved; no averaging into "mixed"

Result: Relational rows that downstream agents can act on individually

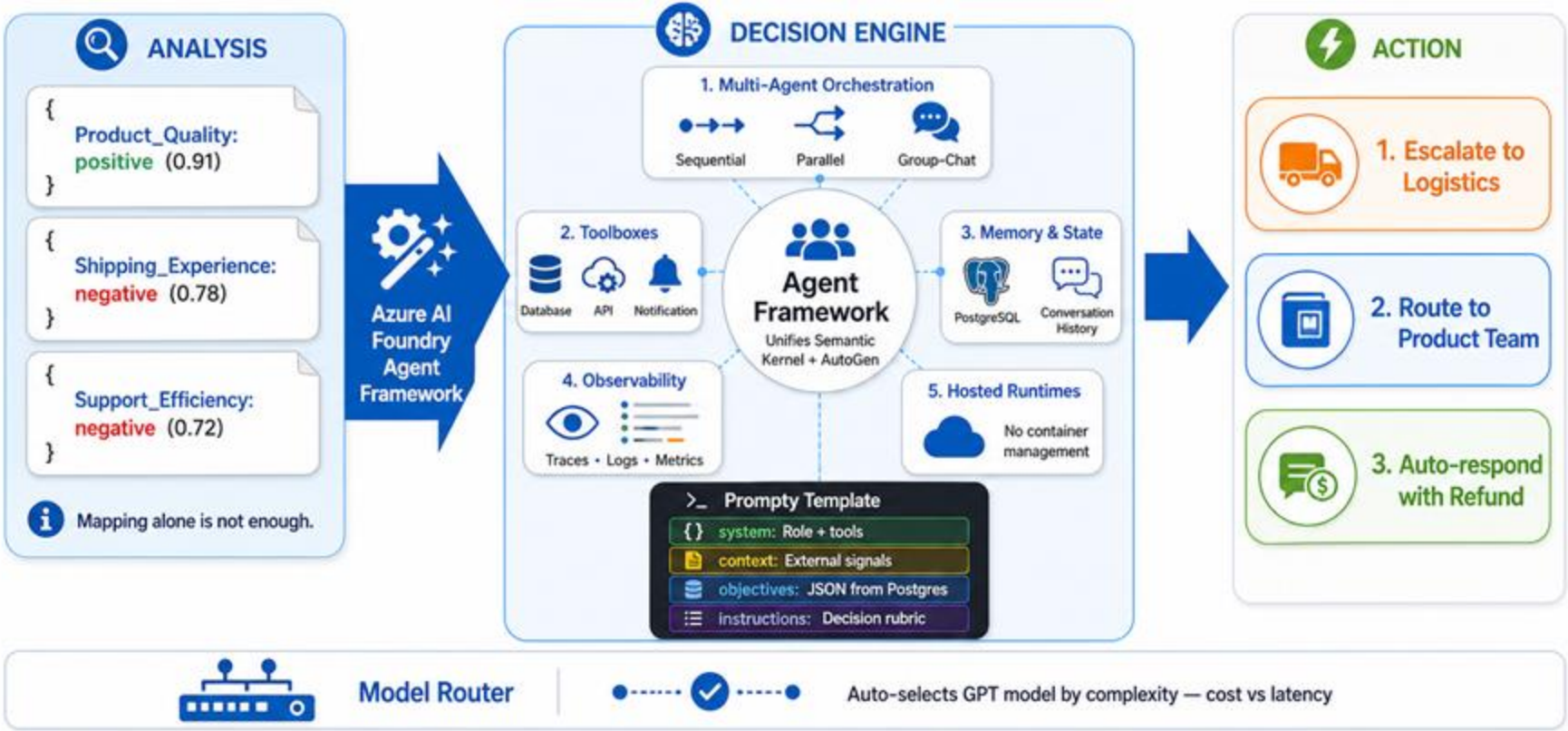
11: Multi-Sentence Complexity



04

Agentic Decision Layer

12: Why Agents? - From Analysis to Action



13: Agent Framework

The Core Idea

An agent framework is not a single LLM prompt or chatbot. It is the layer that turns mapped NLP signals into reliable, auditable business actions.

Production Runtime

A production-grade environment and SDK providing the necessary infrastructure.

Orchestration

Building, orchestrating, and governing multiple AI agents as a coordinated system.

Observability

Integrated layer for observing system behavior and ensuring auditable actions.

Dimension	Singular Agent	Agent Framework
Scope	One LLM with a long system prompt trying to do everything	Multiple specialized agents with narrow, well-defined roles
Tooling	Ad-hoc function calling; brittle prompt engineering	Registered toolboxes with typed schemas, retries, and fallback handling
State	Stateless or manual session management	Persistent memory, conversation threads, and agent state in PostgreSQL
Routing	Implicit routing via prompt instructions	Explicit orchestration: sequential, parallel, and group-chat patterns
Observability	Custom logging wrappers	Native trace replay, benchmark evaluation, and latency metrics

14: Designing the Agent Swarm

Agent taxonomy for our pipeline:

ExtractionAgent – receives mapped objectives from Postgres, decides routing

MappingAgent – deep-dives on Product_Quality signals; queries SKU databases

EnrichmentAgent – integration with external data sources and enrichment of the context

ValidationAgent – validates contextual checks for issues

RoutingAgent – notifications, routing and decision routing

Orchestration pattern: hierarchical, delegates to specialists, then synthesizes a final action plan

15: How Agents Choose - Decision Matrix

Input to Extraction Agent:

1. Mapped objectives (from Postgres)
1. Customer tier (Gold / Silver / Bronze)
1. Real-time external context (from Enrichment Agent)

Decision rules (encoded in agent instructions + Prompty files):

- If `Overall_Experience_intensity > 0.8 AND customer_tier = "Gold" AND regional_weather = "severe"` → **Immediate escalation**
- If `Product_Quality = "positive" AND Pricing_Perception = "negative"` → **Route to Promotions team with discount suggestion**

Output: A structured action record written back to Postgres for audit and downstream CRM sync

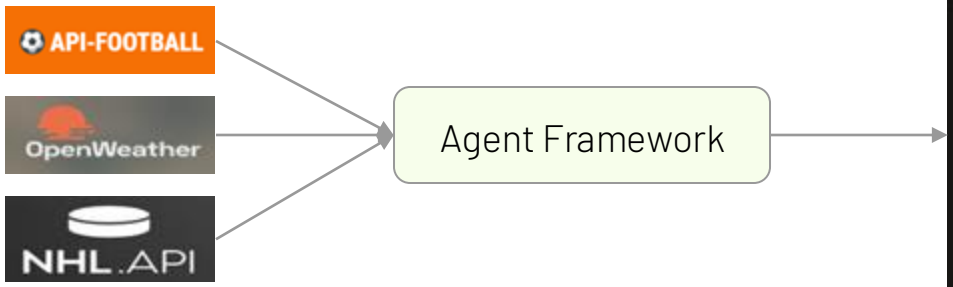
16: External Signals

Weather: Azure Functions polling OpenWeatherMap / Meteomatics → Event Hubs → Fabric streaming → Postgres signals.external

Sporting events: Scheduled data feeds (e.g., FIFA API, ESPN) → Azure Data Factory pipeline → Postgres lookup table by region + event_date

Latency requirement: *ValidationAgent* queries must return in <200ms, so signals are pre-materialized in Postgres, not fetched live per agent call

Geospatial join: Verbatims are tagged with customer_region at ingestion; *ValidationAgent* joins on region + timestamp




```json
{
  "weather_alert": "severe_storm",
  "major_event": "World Cup Final",
  "logistics_delay_expected": true,
  "event_attendance_estimate": 80000
}
```

05

Makin’ it Real - Production

17: Production Considerations

18: Evaluation and Guardrails - Trace-Based

Foundry Evaluations
Grade production traces, not synthetic datasets.

Trace-Based Eval
Point evaluators at live traces → quality scores on real behavior.

Built-In Evaluators
Coherence, fluency, groundedness, relevance, tool-call accuracy, task completion.

Agent-Specific Benchmarks
SocialReasoning-Bench (negotiation) and STATE-Bench (memory).

Custom Evaluators
Domain-specific rubrics: "Did the agent cite the correct evidence span?"

Regression Pipeline
Trace Replay → synthetic eval dataset → CI gate on quality regression.

19: Observability - Full-Stack Tracing

analytics

Foundry Tracing

Captures agent interactions, tool invocations, and LLM reasoning steps.

monitor

Monitoring Infrastructure

Utilizes Azure Monitor and Application Insights for system oversight.

timer

End-to-End Latency

Monitors the full lifecycle from ingestion to CRM webhook transmission.

speed

Performance Metrics

Tracks `mapped_per_minute` and `decision_latency_p95`.

history_edu

Compliance Auditing

PostgreSQL audit tables provide immutable records of objective mappings.

account_tree

Data Lineage

Fabric Lineage Extractor ensures column-level traceability from bronze to agent input.

20: Security, Governance & Compliance

public_off

Network Isolation

Managed VNET (GA) — private endpoints for Fabric, Foundry, Postgres, Event Hubs

fingerprint

Identity & Access

Entra ID with conditional access; workload identities for notebooks → Foundry → Postgres

gavel

Data Governance

Microsoft Purview scans OneLake + Postgres for sensitivity labels and lineage

monetization_on

Cost Attribution

Project-level cost tracking in Foundry per agent team and per model

security

Policy Guardrails

Azure Policy restricts model deployments, enforces PTU usage, and blocks public endpoints

21: CI/CD & GitOps for the Entire Pipeline

Artifact	Tooling	Promotion
Infra (Fabric, Foundry, Postgres, VNET)	Bicep/Terraform	Dev → Staging → Prod via Azure DevOps
Fabric pipelines/notebooks	Fabric Git integration (Azure Repos)	Workspace deployment pipelines
Agent code (Python SDK)	Azure DevOps Pipelines → Foundry hosted runtimes	Environment-scoped agent versions
Prompty files	Git + Foundry project deployment	Version-tagged in Foundry model registry
Postgres schema	Flyway/Liquibase migrations/ Fabric Mirroring	Applied via pipeline stage before agent deploy
Test data / eval datasets	Versioned in Git LFS / Azure ML Data assets	Promoted with code

22: Disaster Recovery and High Availability

storage **Postgres**
Zone-redundant Flexible Server (RPO=0, RTO<30s), read replicas in paired region

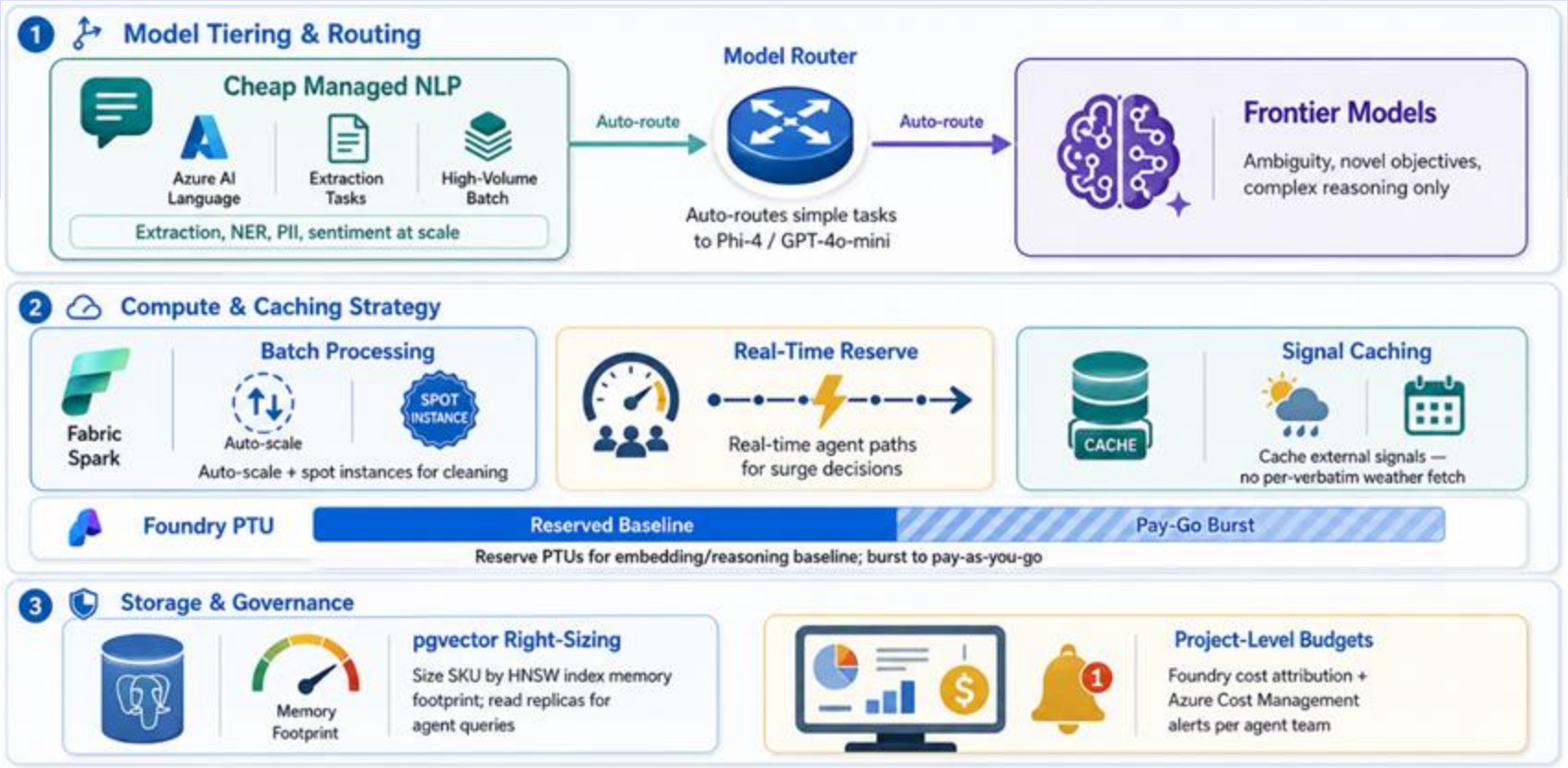
cloud_queue **Fabric**
OneLake geo-redundancy; pipeline state recovered via deployment pipelines

smart_toy **Foundry Agents**
Stateless hosted runtimes — re-deploy from Git in < 5 min; state in Postgres survives

hub **Event Hubs**
Geo-disaster recovery with namespace pairing

menu_book **Runbook**
Quarterly DR drill — failover Postgres, verify agent state consistency, validate eval scores

23: Cost Optimization - Running at Scale



Every dollar saved on inference is a dollar reinvested in model quality

06

Closeout

24: Key Takeaways and Recap

Data Interpretation
Sentiment APIs fail on real text. Map clauses to business objectives, not emotions.

Pipeline Architecture
Fabric cleans and streams. OneLake bronze → silver → gold with quality gates.

The Brain (Storage)
PostgreSQL is the brain. Vectors, objectives, agent state, and external signals in one place.

The Engine (Compute)
Foundry runs the show. Models, embeddings, agents, tracing, and evaluation.

Contextual Reality
Context changes decisions. Pre-materialize weather and events so agents act on reality.

Signal Integrity
Preserve every signal. No averaging. Relational rows let downstream agents act independently.

Framework Focus
Framework > single agent. Orchestration, memory, and tools separate prototypes from production.

Production-First
Production is not an afterthought. Evals, guardrails, and CI/CD are built into Azure.

25: Resources

Azure AI and agents

- Microsoft Foundry overview — platform overview for models, apps, and agents
- What's new in Microsoft Foundry — current platform capabilities, hosted runtimes, model router, tracing, and evaluations
- Microsoft Agent Developer Resources — Agent Framework docs, SDKs, samples, and architecture patterns
- AutoGen repository note — indicates Microsoft Agent Framework as the enterprise-ready successor for agentic patterns

NLP and language services

- Azure AI Language overview — sentiment, opinion mining, NER, PII, and language detection capabilities
- Sentiment analysis and opinion mining — Microsoft documentation for aspect-based sentiment analysis concepts

Data pipelines and storage

- Fabric Data Factory documentation — enterprise data movement and transformation patterns
- Data Factory in Microsoft Fabric tutorial — pipelines, notebooks, parallel orchestration, and end-to-end integration flow
- Azure Database for PostgreSQL + pgvector — vector search and embeddings support in PostgreSQL Flexible Server

Architecture and visuals

- Azure Architecture Icons — official Microsoft icon set for architecture and slide diagrams
- Professional Azure diagrams with PowerPoint — practical guidance for presentation-ready Azure visuals

25: Resources

https://learn.microsoft.com/en-us/azure/foundry/what-is-foundry
https://learn.microsoft.com/en-us/azure/foundry/whats-new-foundry
https://microsoft.github.io/agent-resources/develop-agents/
https://github.com/microsoft/autogen
https://learn.microsoft.com/en-us/azure/ai-services/language-service/overview
https://learn.microsoft.com/en-us/fabric/data-factory/tutorial-end-to-end-introduction
https://learn.microsoft.com/nl-nl/azure/postgresql/flexible-server/how-to-use-pgvector

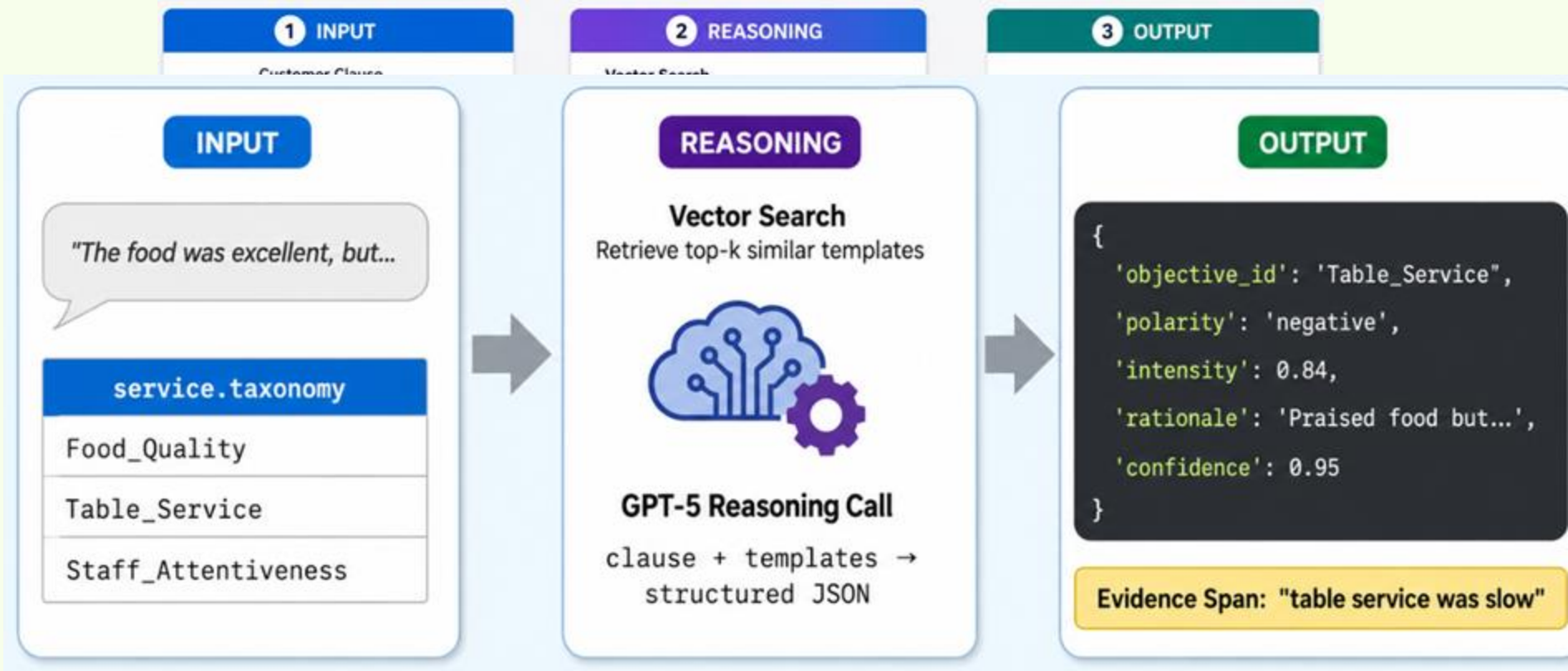
Contact

https://github.com/halfspin-qc/NLP_MultiAgent

Q/A | Consulting | Speaking | Startups

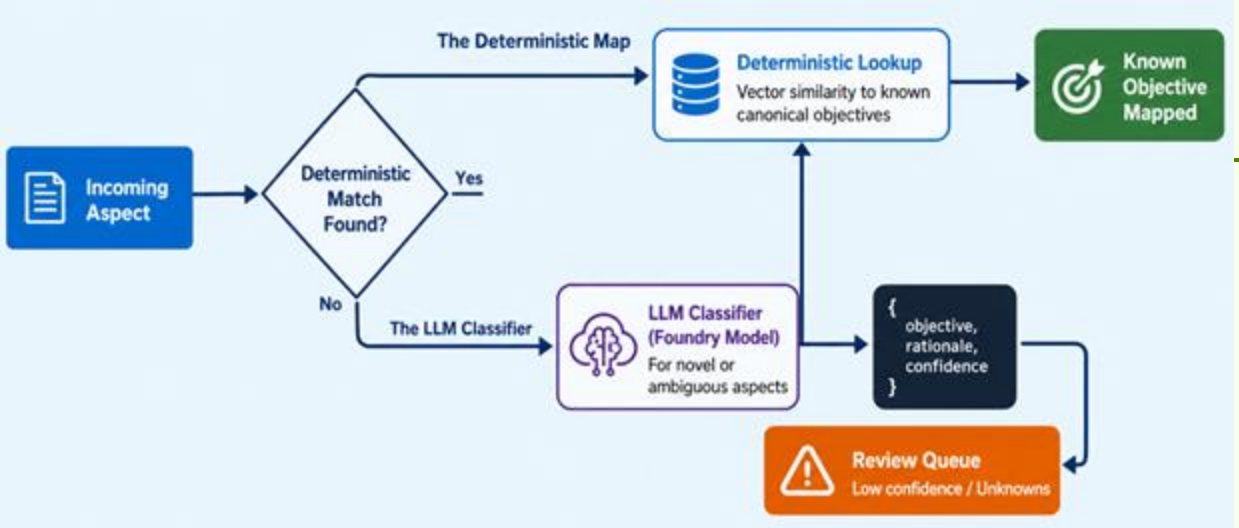
THANK YOU

07: Semantic Objective Mapping - NLP Core

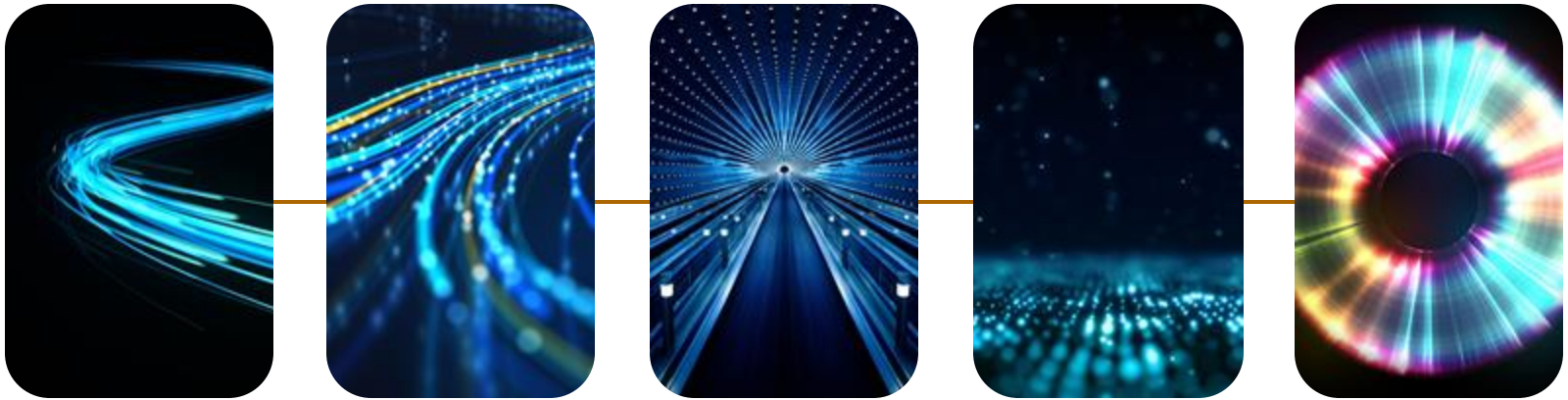


- Structured output: JSON with objective, polarity, intensity, rationale

07: Semantic Objective Mapping - NLP Core



- Use the opinion mining feature of Azure AI Language sentiment analysis = aspect-based sentiment analysis (ABSA)
- It returns sentiment at document, sentence, and aspect level, plus a document-level "mixed" label
- For each aspect it surfaces a target (noun/verb) and an assessment (opinion word) with positive/negative confidence
- Taxonomy is a living asset. New slang, products and failure modes adapt over time



https://www.scribd.com/document/996212449/Explainable-Aspect-Based-Sentiment-Analysis-Using-Transformer-Models

Strategy Canvas