

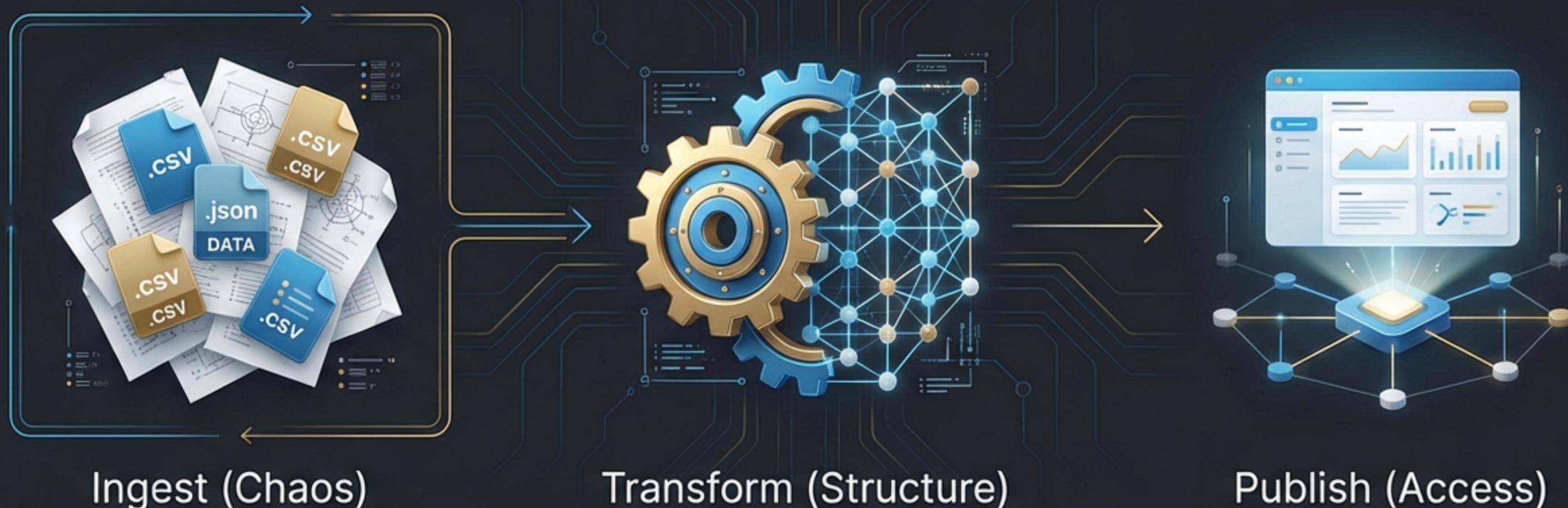
# Virtuoso: The Unified Platform for Knowledge Graphs

A Powerful Multi-Model DBMS for SQL and SPARQL



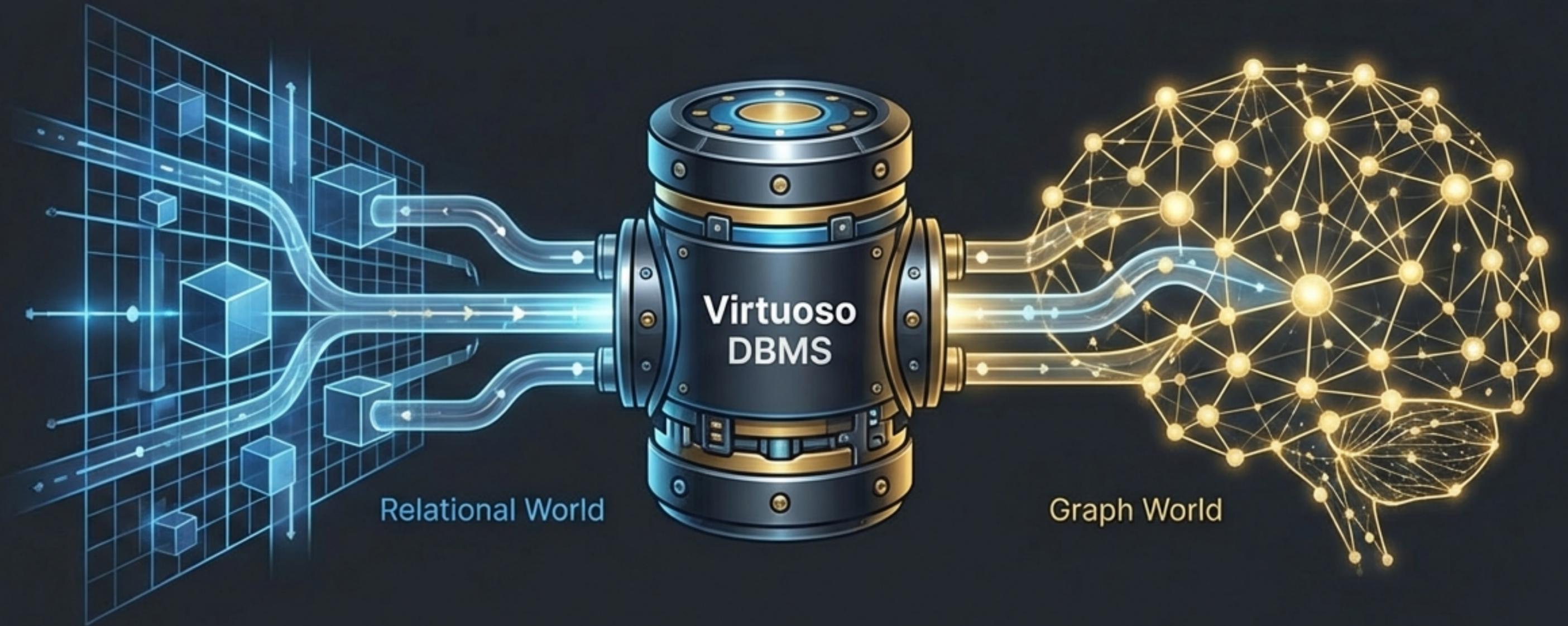
The Unified Lifecycle: Ingest -> Model -> Query -> Deploy

# What a Production Knowledge Graph Actually Looks Like



Virtuoso delivers the full lifecycle. It ingests messy source data, transforms it into structured knowledge, and publishes it with durable identifiers.

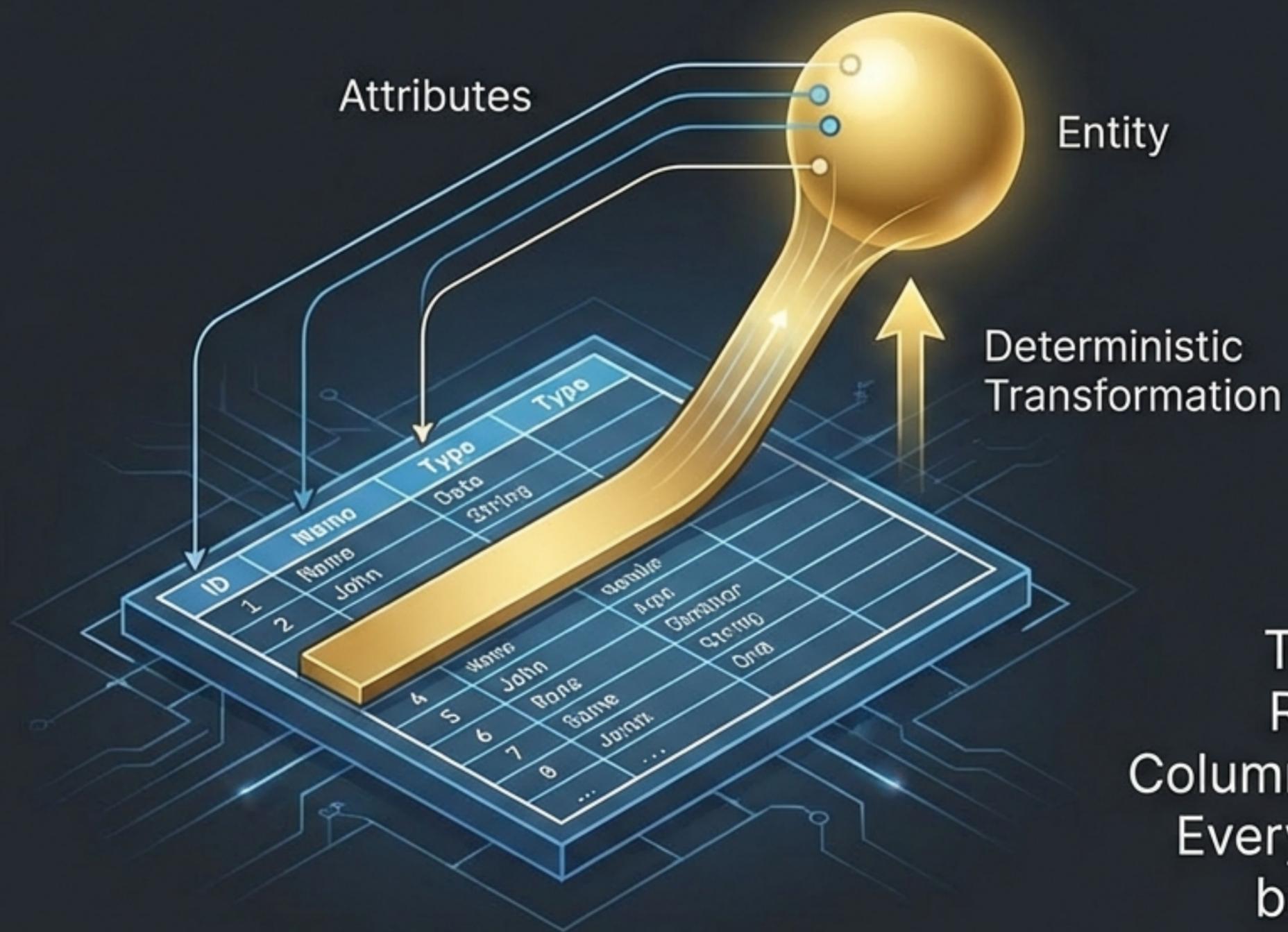
# The Multi-Model Advantage



One Engine, Two Worlds. Native support for SQL and SPARQL.

No Bolt-Ons. A unified engine handling RDF and Relational models simultaneously.

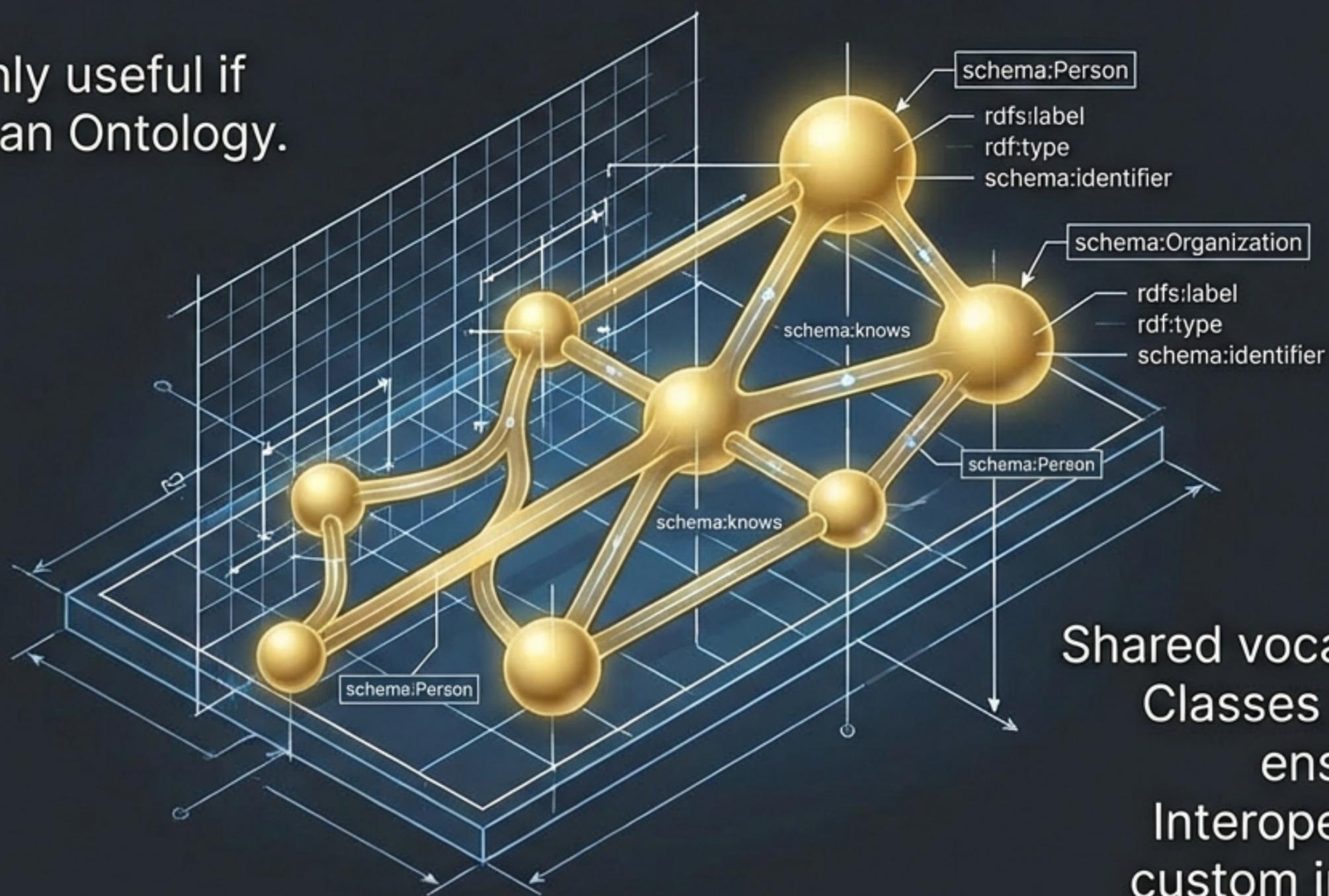
# Starting with Reality: The CSV Pipeline



Traceable Provenance:  
Rows become Entities.  
Columns become Attributes.  
Every node retains lineage  
back to **the source file.**

# Consistency Through Ontology

A graph is only useful if informed by an Ontology.



Shared vocabularies define Classes and Properties, ensuring Semantic Interoperability without custom integration glue.

# No Compromise Querying

The image illustrates a dual-view querying interface. On the left, a code editor window titled "SQL for Analysts" displays the following SQL query:

```
SELECT name, email
FROM Users
WHERE id = 101;
```

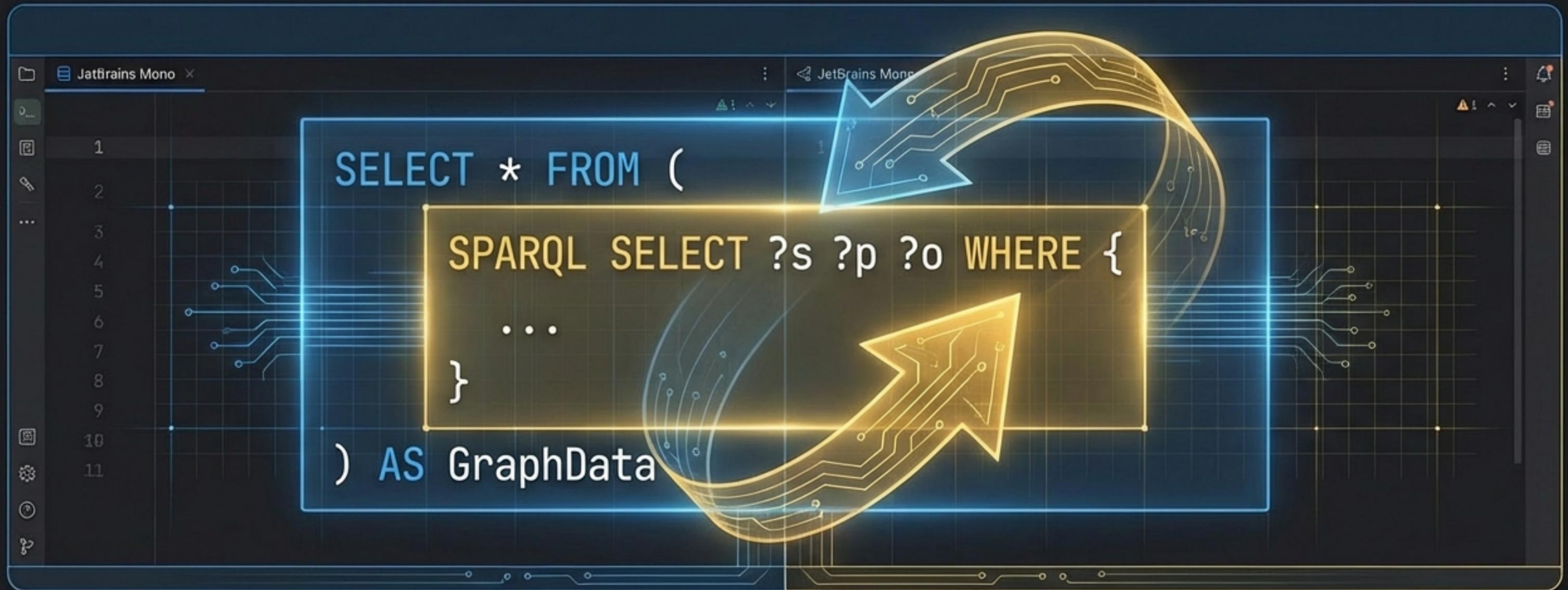
On the right, a code editor window titled "SPARQL for Engineers" displays the following SPARQL query:

```
SELECT ?name ?email
WHERE {
  ?s foaf:name ?name ;
  foaf:mbox ?email .
}
```

A central graphic features a glowing blue cube with a globe inside, connected to circuit-like lines that bridge the two code editors, representing a unified data source accessible through different query languages.

Analysts keep their SQL views. Graph engineers get native SPARQL patterns. Both access the exact same live data.

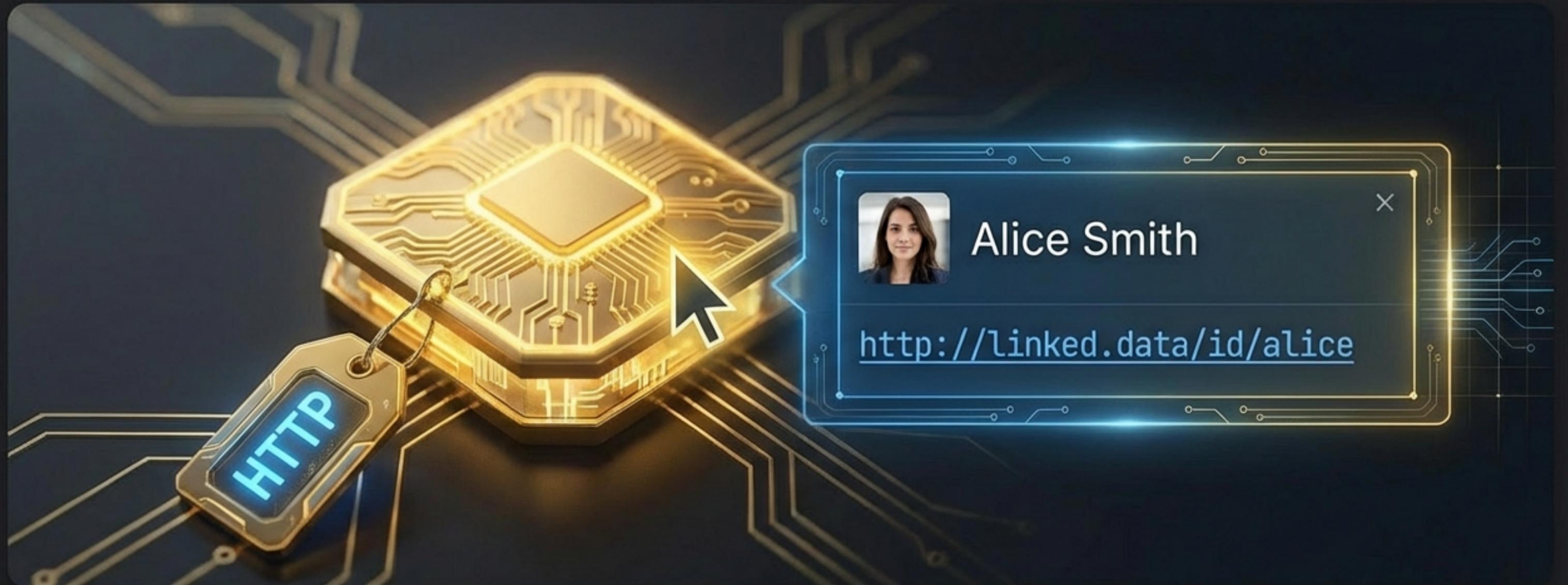
# SPASQL: Bridging the Divide



```
1 SELECT * FROM (  
2  
3     SPARQL SELECT ?s ?p ?o WHERE {  
4         ...  
5     }  
6  
7 ) AS GraphData
```

The Killer Feature: Embed SPARQL directly inside SQL. Leverage relational tables and graph reasoning in a single query execution.

# From Database to Linked Data



Entities need durable, dereferenceable Identifiers. Identifiers resolve to rich descriptions consumable by both humans and AI agents.

# Exploration Without Querying

The screenshot shows a 'Faceted Search' window with a search bar and a grid of results. The filters on the left are:

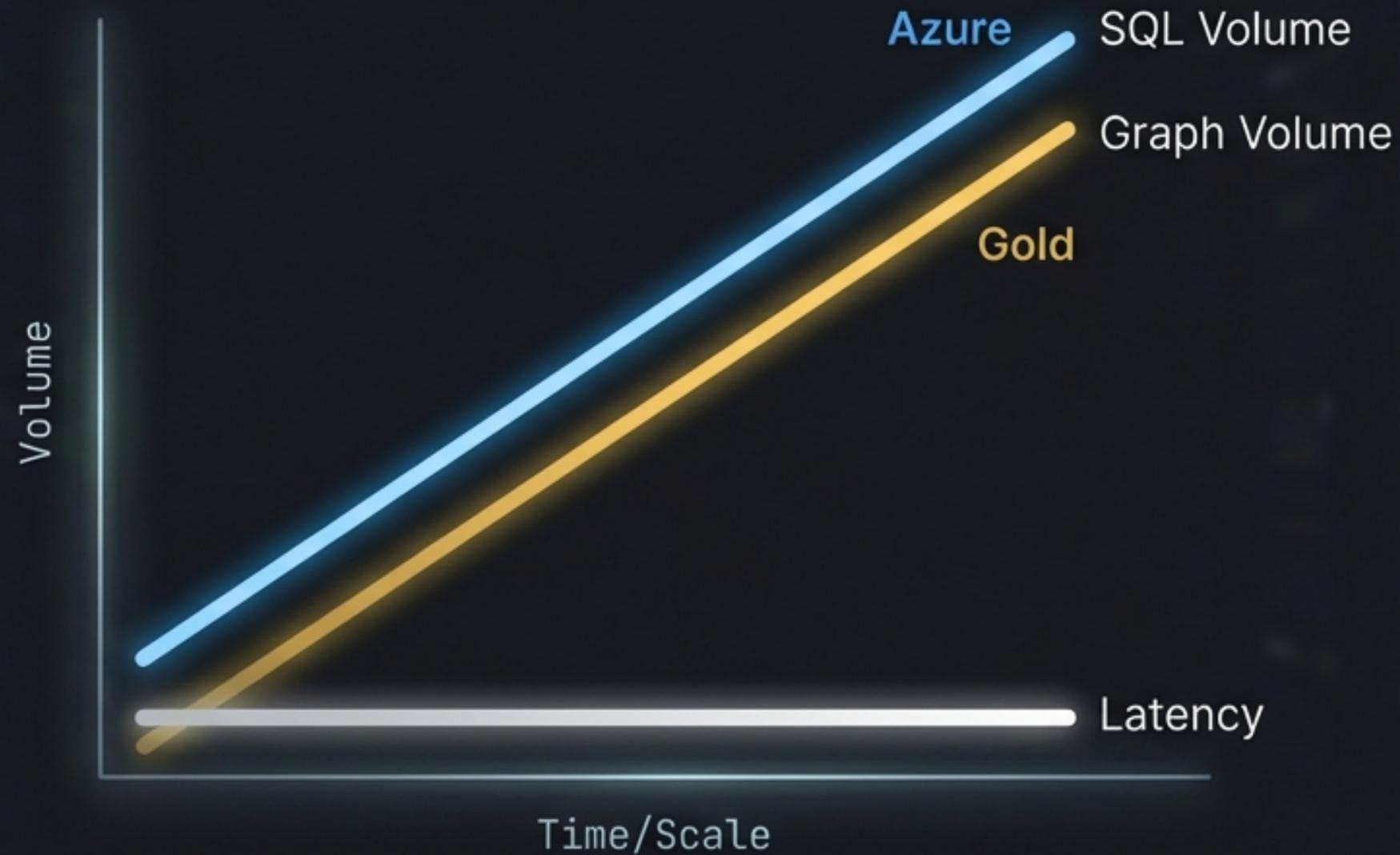
- FILTERS**
- Type**
  - Person (245)
  - Company (150)
  - Project (80)
- Date**
  - 2024 (100)
  - 2023 (180)
  - 2022 (95)
- Location**
  - New York (110)
  - San Francisco (90)
  - London (75)
- Sector**
  - Technology (120)
  - Finance (100)
  - Healthcare (60)

The search bar contains the text 'Search results for applied filters...'. The results grid shows:

- Alice Smith** (Person, CTO, San Francisco, 2024)
- TechCorp** (Company, AI Solutions, New York, 2023)
- Project Chimera** (Project, Research, London, 2024)
- Project Chimera** (Project, Research, London, 2024)
- Bob Johnson** (Person, Lead Engineer, London, 2023)
- InnovateNow** (Company, Biotech, Boston, 2024)
- Project Chimera** (Project, Research, London, 2024)
- InnovateNow** (Company, Biotech, Boston, 2024)
- Bob Johnson** (Person, Lead Engineer, London, 2023)

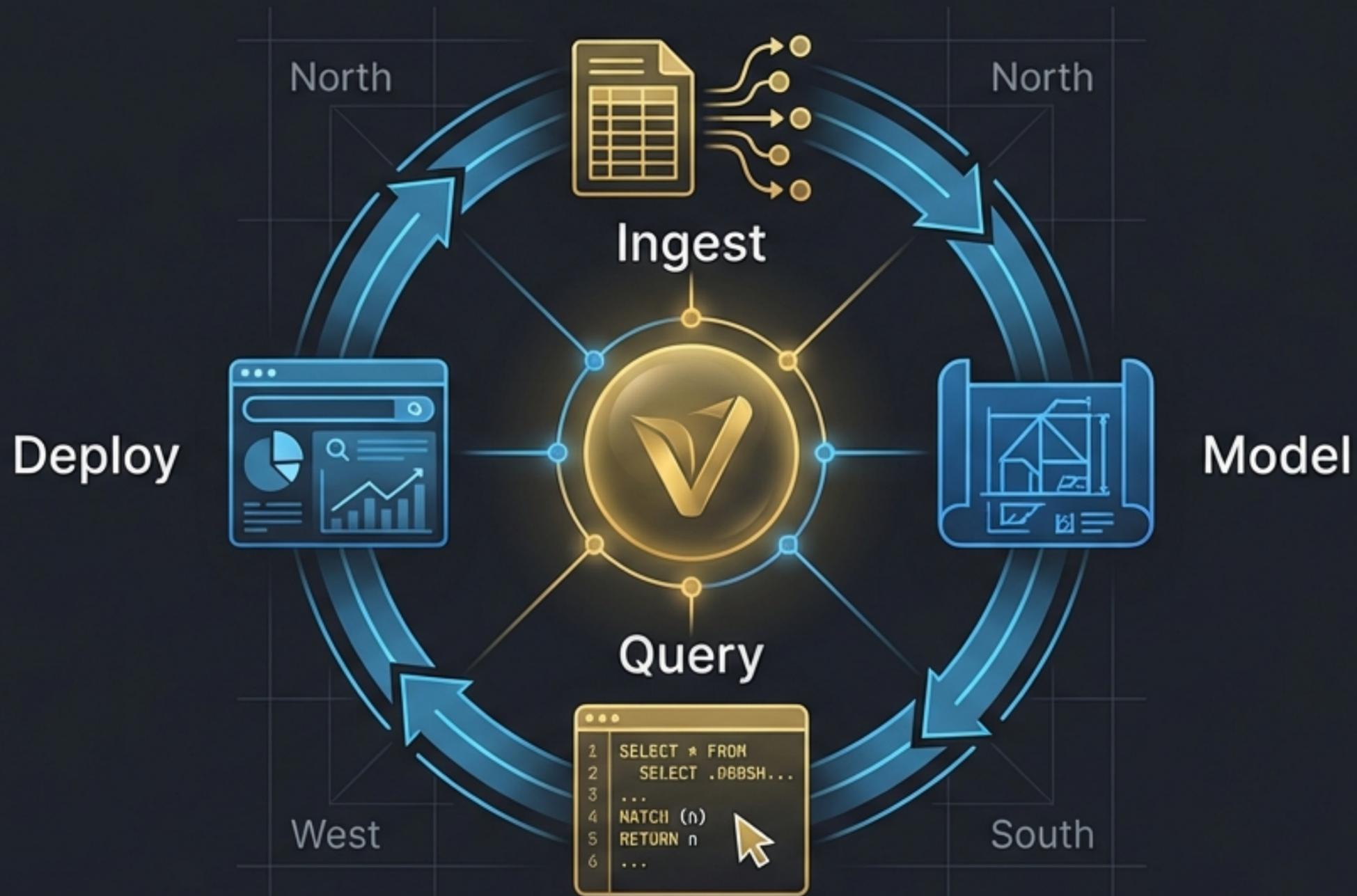
Native faceted navigation allows users to browse the graph via attributes without writing a single line of code.

# Prototype to Production



Built for Scale. High concurrency, optimized query execution, and multi-model storage without the need for migration.

# The End-to-End Solution



Virtuoso treats the Knowledge Graph as a complete system, not just a data format.

# Bridge the World of Data



Stop migrating between systems.  
Start building a rigorous, explorable **Knowledge Graph** with Virtuoso.

# Questions & Discussion



Virtuoso: The Unified Platform