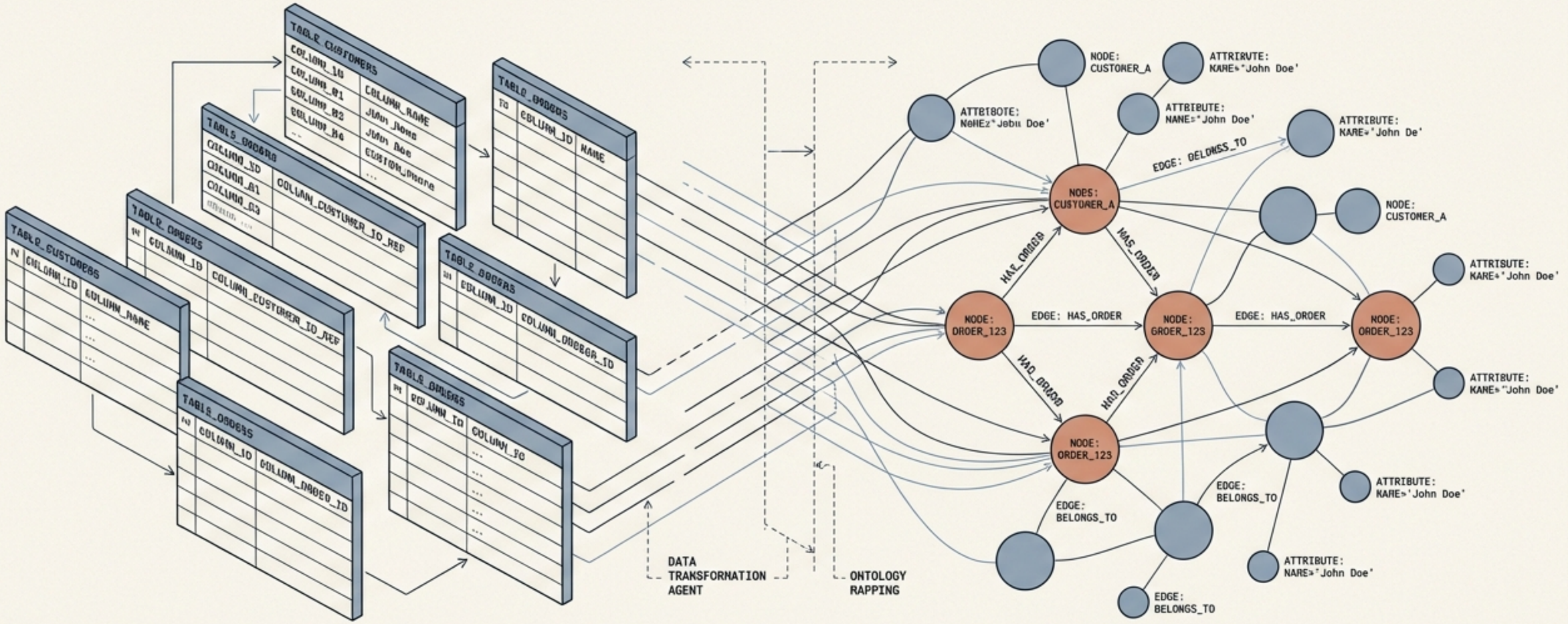


PostgreSQL to Knowledge Graph: Automating Linked Data Deployment with AI.

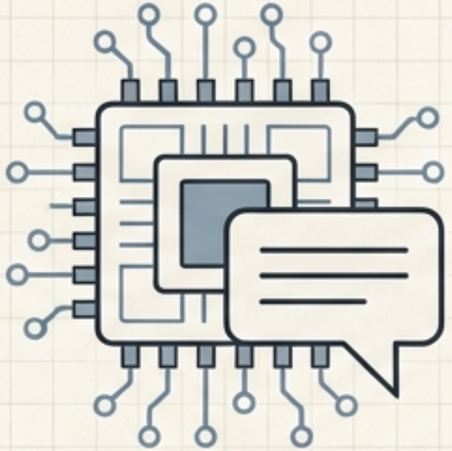
A technical teardown of zero-copy data virtualization and ontology generation using an AI agent pipeline.

The Challenge

Siloed RDBMS Data

Standard PostgreSQL schemas remain rigid and disconnected, isolated from broader enterprise interoperability and advanced semantic querying.

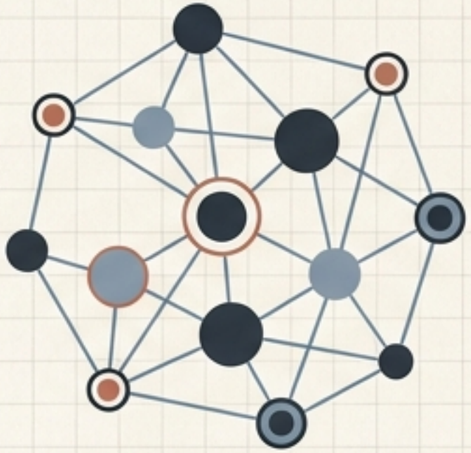
The Catalyst



AI-Driven Linked Data Skill

A conversational AI agent intelligently orchestrates Virtuoso database tools to autonomously model, map, and transform the underlying schemas.

The Outcome



Virtualized Knowledge Graph

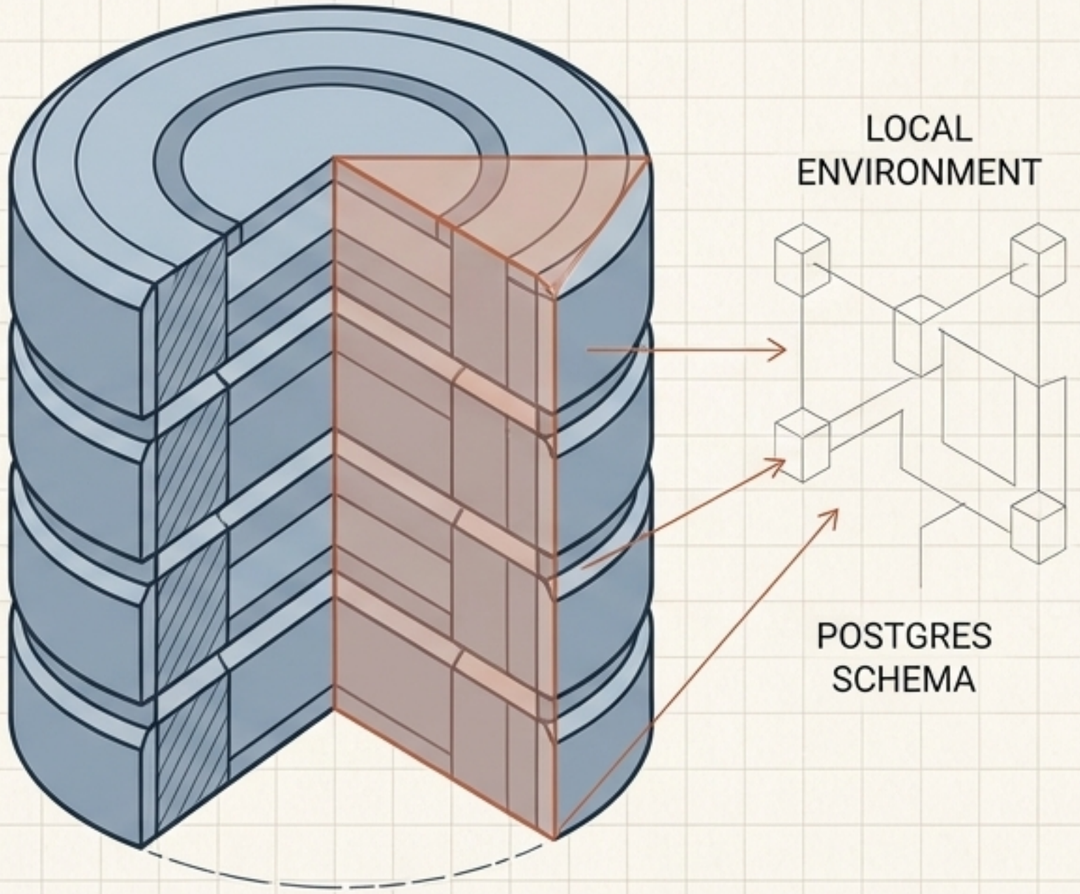
Deployment of a fully addressable, Linked Data-compliant SPARQL endpoint over the legacy database without moving physical data.

AI-Driven Schema Discovery

The AI agent utilizes built-in environmental functions to autonomously probe the database structure.

```
database_schema_objects(qualifier='postgres')
```

> The scan successfully surfaces local schemas without requiring hardcoded Data Source Names, establishing the foundational target: the postgres schema.



Targeting the Star Schema

Option A

Clinical Trials (`postgres.nih_clinical_trials.*`)

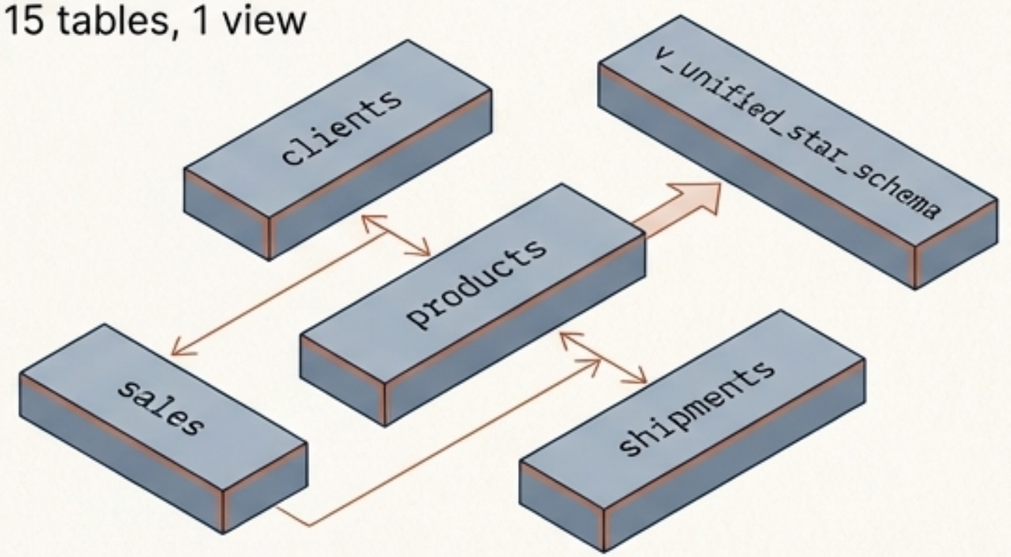
40+ tables

- `study_protocol`
- `outcomes`
- `interventions`
- `conditions`
- `facilities`
- `investigators`
- `sponsors`
- `results_data`
- `etc.`

Option B

Demo Star Schema (`postgres.postgres_jdbc_mt.*`)

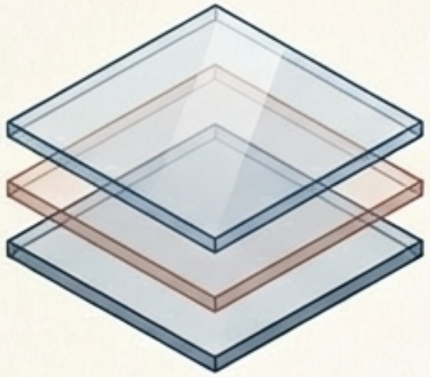
15 tables, 1 view



The pipeline cleanly isolates the target `postgres_jdbc_mt` schema, setting up the semantic transformation of core business entities.

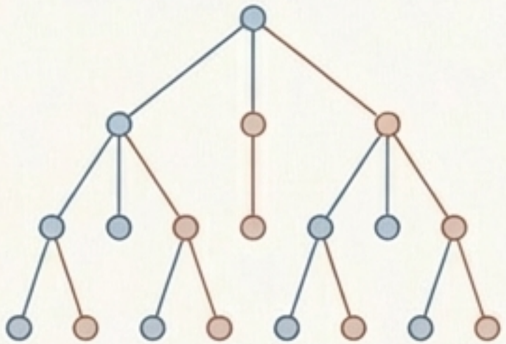
The Transformation Engine

Virtual Knowledge Graph



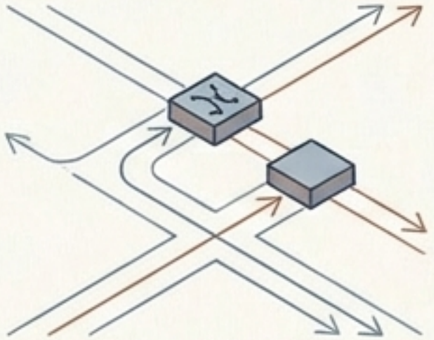
Creation of RDF Views using dynamic SQL/SPARQL quad maps.

OWL Ontology



Automated generation of semantic relationships and namespaces.

Linked Data Rules

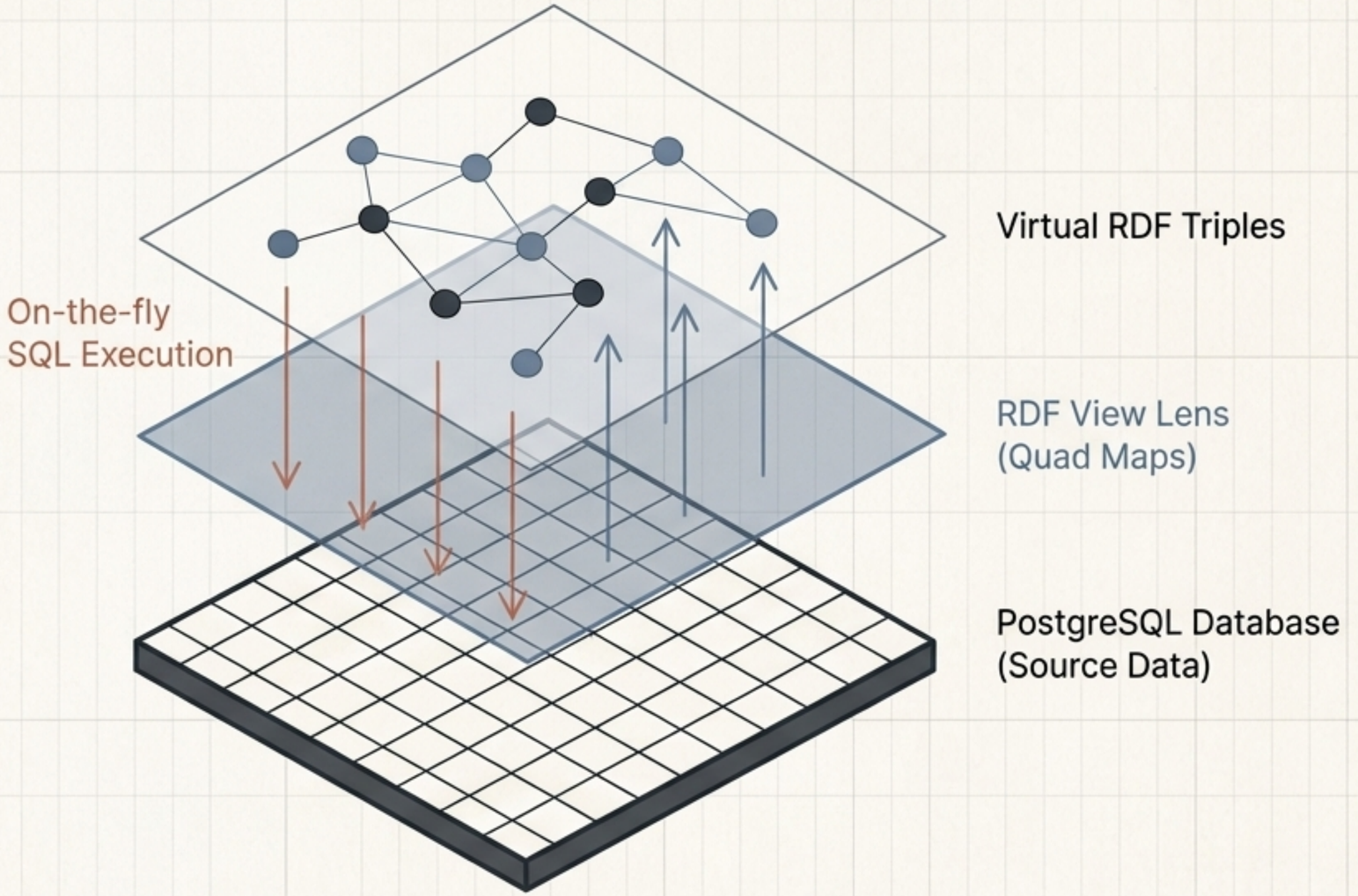


Configuration of HTTP virtual paths and VHOST URL rewrite scripts.

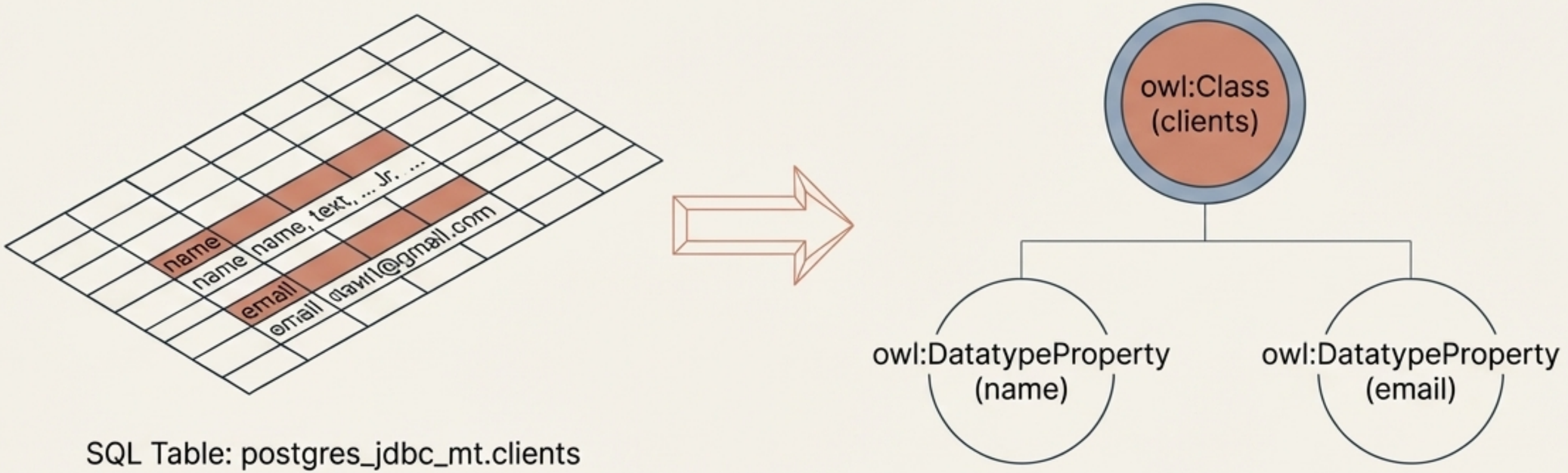
PostgreSQL

Zero-Copy Architecture via Virtual RDF Views

The source data remains entirely in Postgres. The virtualization lens interprets SPARQL queries, dynamically translates them into SQL, and serves virtual triples without moving a single byte of physical data.



Automated OWL Ontology Generation



SQL Table: postgres_jdbc_mt.clients

Baseline Schema Prefix:
`http://demo.openlinksw.com/schemas/postgres_postgres_jdbc_mt/`

The agent autonomously bridges the semantic gap, assigning formal vocabulary definitions to legacy column headers, making the data machine-readable on the semantic web.

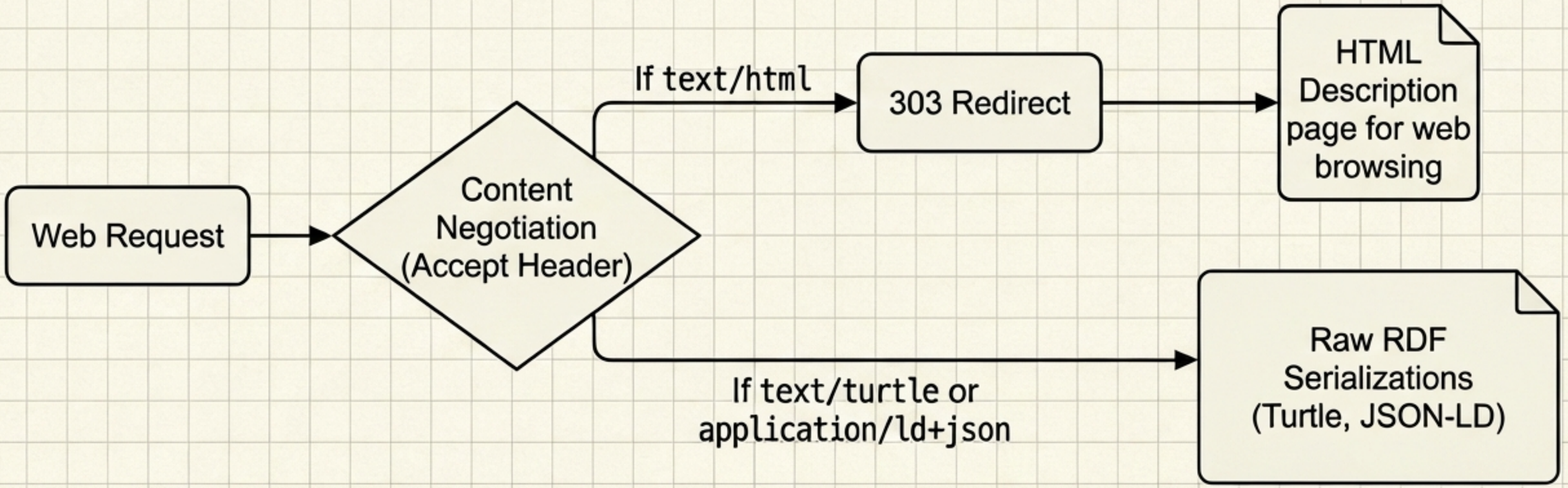
IRI Minting & Host Deployment

The Problem: Initial VHOST rewrite script failed due to blank arguments in vector calls.

~~`vector(..., , '(text/rdf...)', ...)`~~

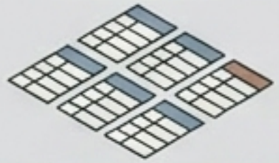
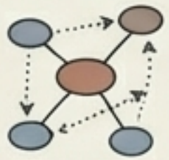
The Resolution: AI identifies the gap and requests the public host. User provides `demo.openlinksw.com`. The AI patches the blank arguments and successfully deploys the rewrite rules.

Output Variable: `^{URIQADefaultHost}^` resolves globally to demo.openlinksw.com.



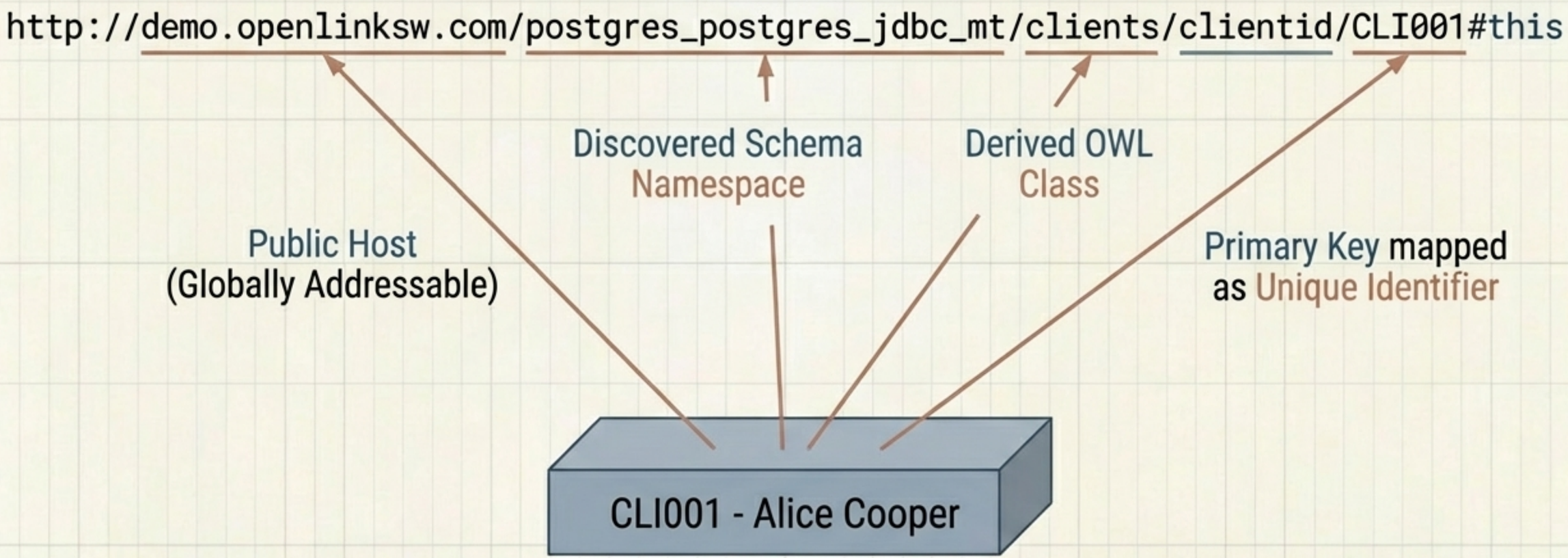
The deployed endpoint simultaneously serves human analysts browsing via web browsers and automated machine agents querying via SPARQL protocols.

Paradigm Shift: Legacy RDBMS vs. Linked Data Graph

Dimension	PostgreSQL (Start)	Linked Graph (End)
Data Structure	Rigid Tables & Rows 	Fluid Nodes & Edges 
Identifiers	 Local Primary Keys (Isolated) `ID: 123`	 Global URIs (Web-addressable) `<http://example.org/entity/123>`
Query Language	 SQL (Relational joins) `SELECT * FROM A JOIN B ON A.id = B.a_id`	 SPARQL (Pattern matching across domains) `SELECT ?s WHERE {?s ?p ?o}`
Schema	 Fixed, predefined schema	 Flexible, inferable OWL Ontology

The transformation liberates the data from internal application logic, making it universally discoverable and interoperable.

Anatomy of a Linked Entity



A previously siloed string of text in a private database is now a universally unique, queryable node on the Semantic Web.

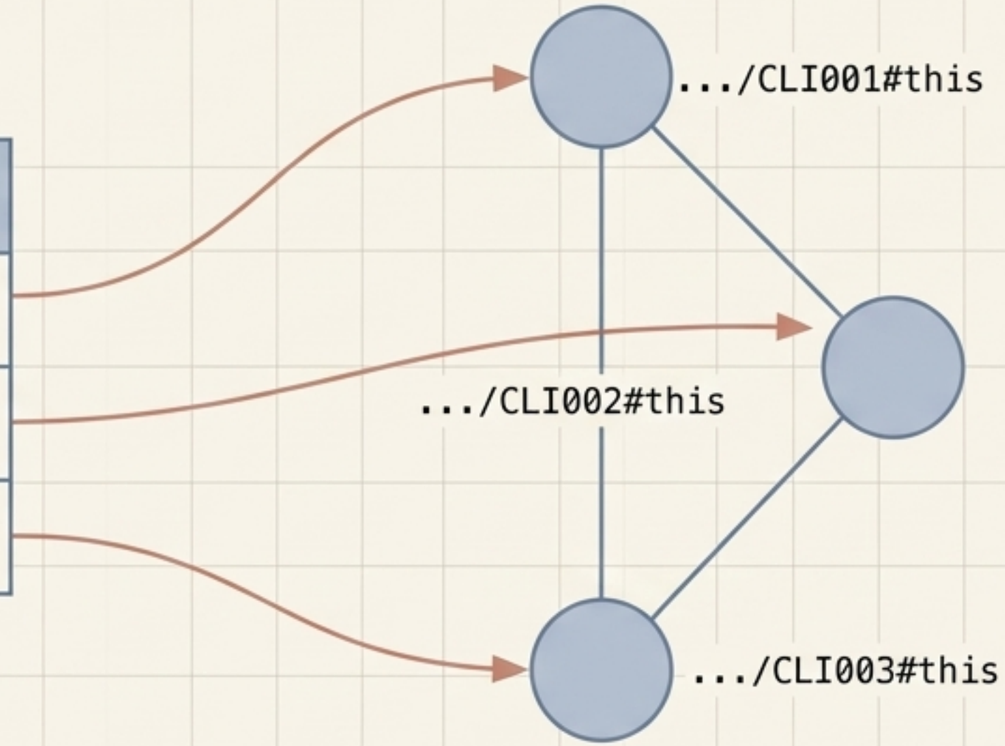
Querying the Virtual Graph

```
SELECT ?client ?name ?email ?countryName
FROM <http://demo.openlinksw.com/postgres_postgres_jdbc_mt#>
WHERE {
  ?client a <http://demo.openlinksw.com/schemas/postgres_postgres_jdbc_mt/clients> ;
  <...name> ?name ;
  <...email> ?email .
}
```

The specific FROM clause points the SPARQL engine directly to the dynamically generated virtual graph endpoint, seamlessly bridging the query to the underlying relational tables.

Live Output: The Graph in Action

	Name	Email	Country
1	Alice Cooper	alice.cooper@email.com	US
2	Bob Johnson	bob.johnson@email.com	CA
3	Carol Smith	carol.smith@email.com	UK

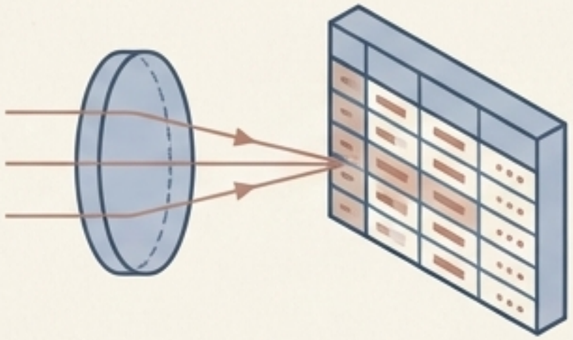


Real-time extraction of tabular data formatted flawlessly as global Linked Data entities, all executed via AI automation.

Redefining Data Engineering Velocity

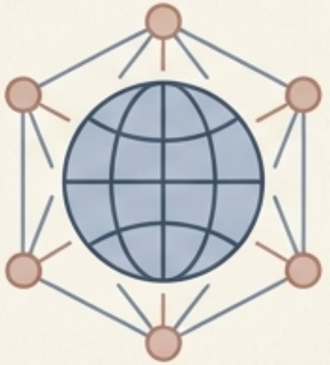
Speed to Deployment

Translates traditional multi-month, manual ontological modeling projects into an automated pipeline executed in minutes.



Zero Data Duplication

Establishes a unified semantic layer without the cost, risk, or latency of ETL pipelines and physical data movement.



Enterprise Interoperability

Instantly upgrades legacy RDBMS assets to adhere to W3C global standards including OWL, SPARQL, and Linked Data.

Semantic data integration is no longer gated by complexity; it is now an automated, repeatable capability driven by AI.