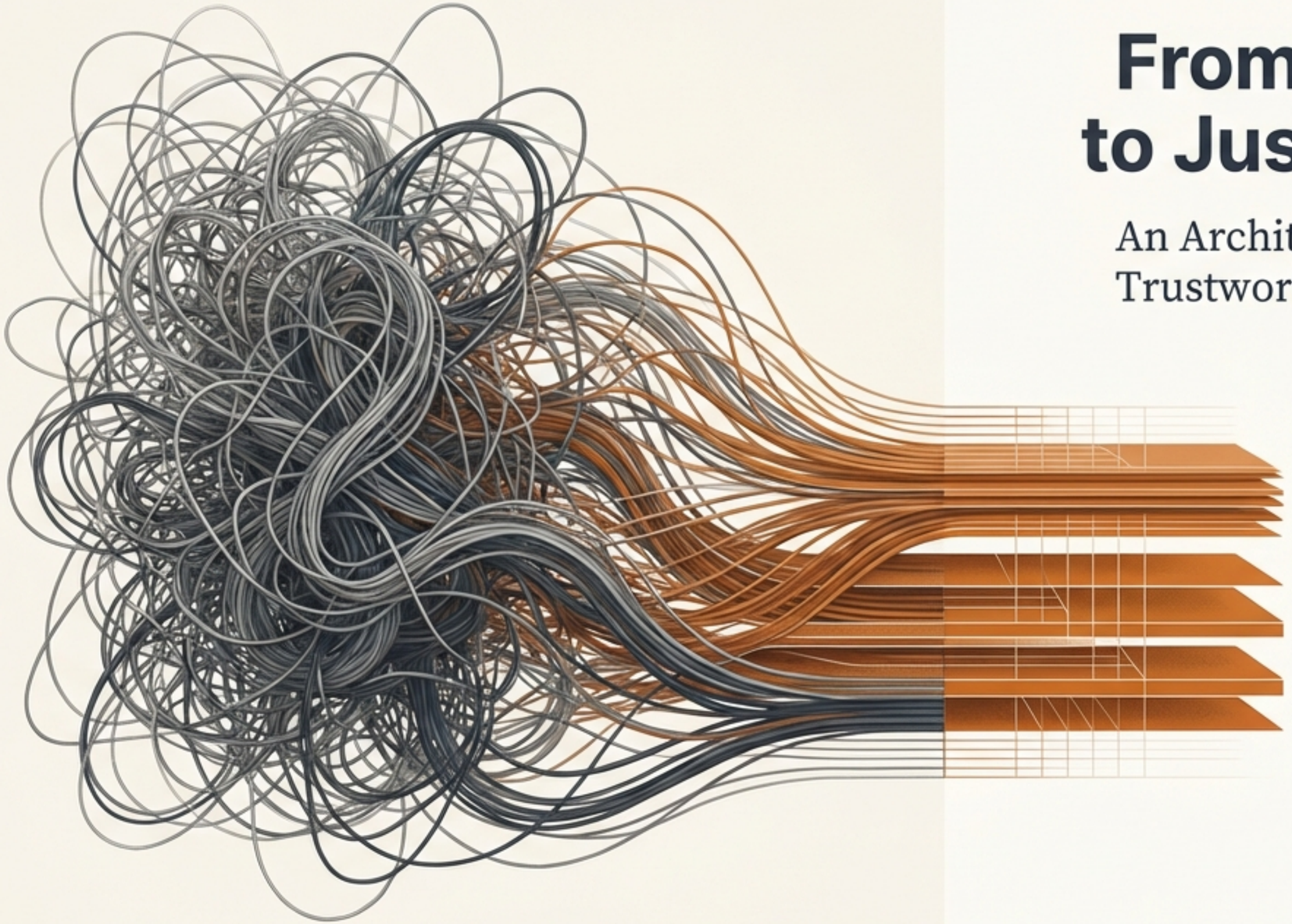


From Unbounded AI to Justified Revenue

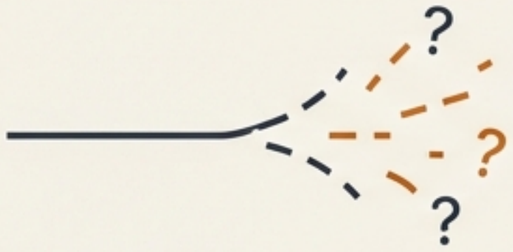
An Architectural Blueprint for Trustworthy, Data-Driven Decisions

Modern AI Fails When It Lacks Structure, Scope, and Trust

LLMs don’t fail because data is missing.
They fail because the architecture is flawed.

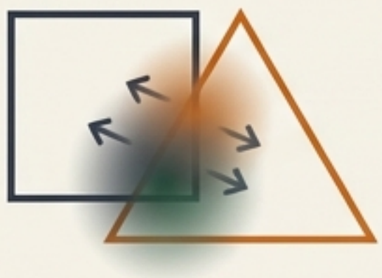
Unbounded Context

Models are given access to everything, leading to irrelevant or incorrect outputs.



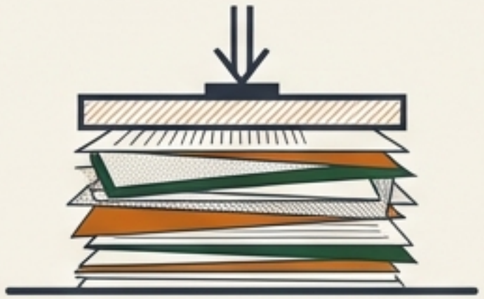
Missing Provenance

Answers are generated without a clear, auditable trail back to the source data or claims.



Implicit Meaning

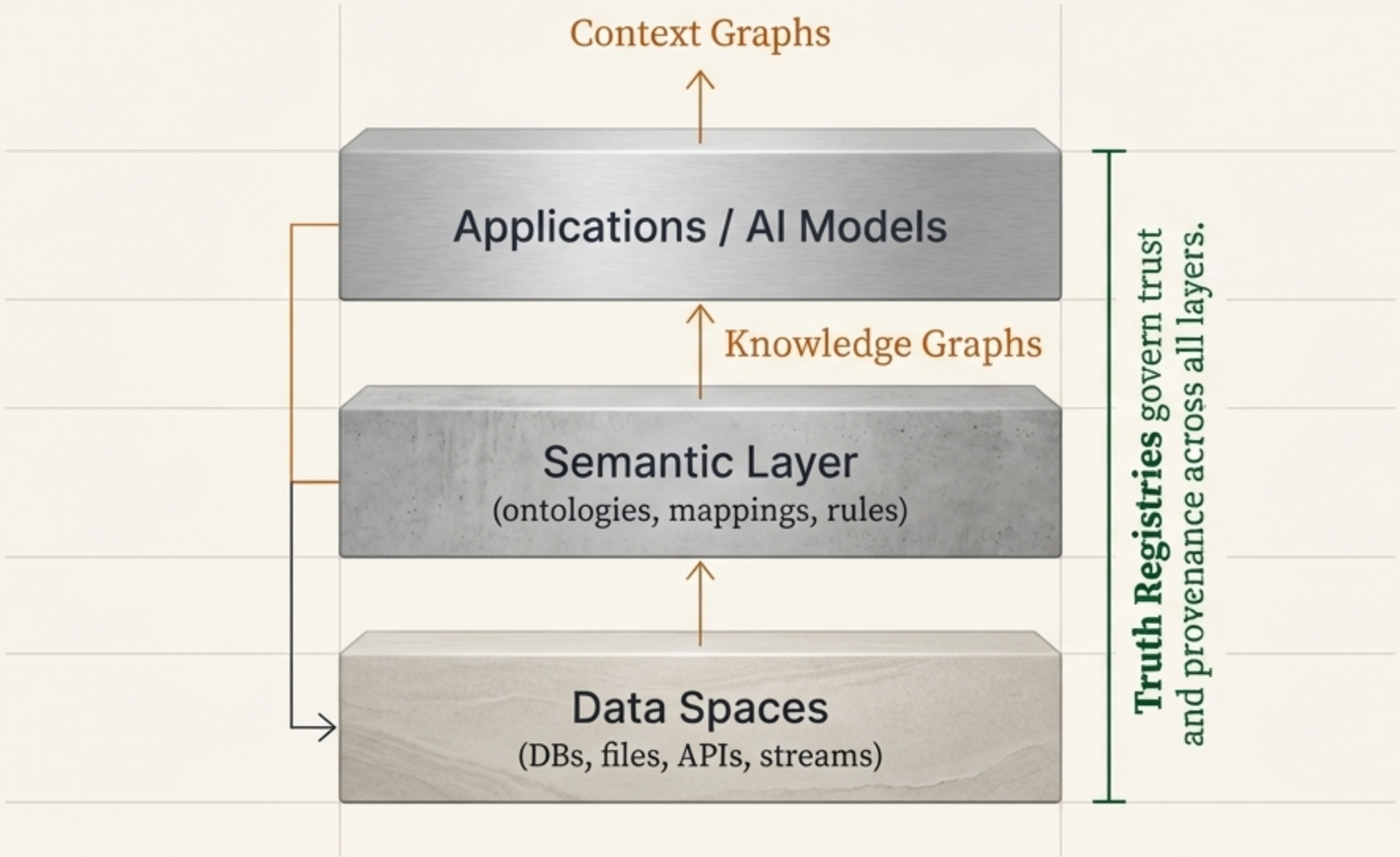
The system has to guess the relationships between data points, leading to misinterpretation.



Flattened Truth

All information is treated as equally valid text, erasing nuance, authority, and context.

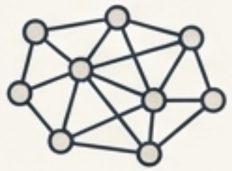
The Solution is a Layered AI-Data-Semantics Stack



Each Layer Answers a Different, Critical Question

Data Spaces answer: *“Where does it come from?”*

Semantic Layers answer: *“What does it mean?”*



Knowledge Graphs answer: *“What is asserted?”*

Context Graphs answer: *“What is relevant now?”*

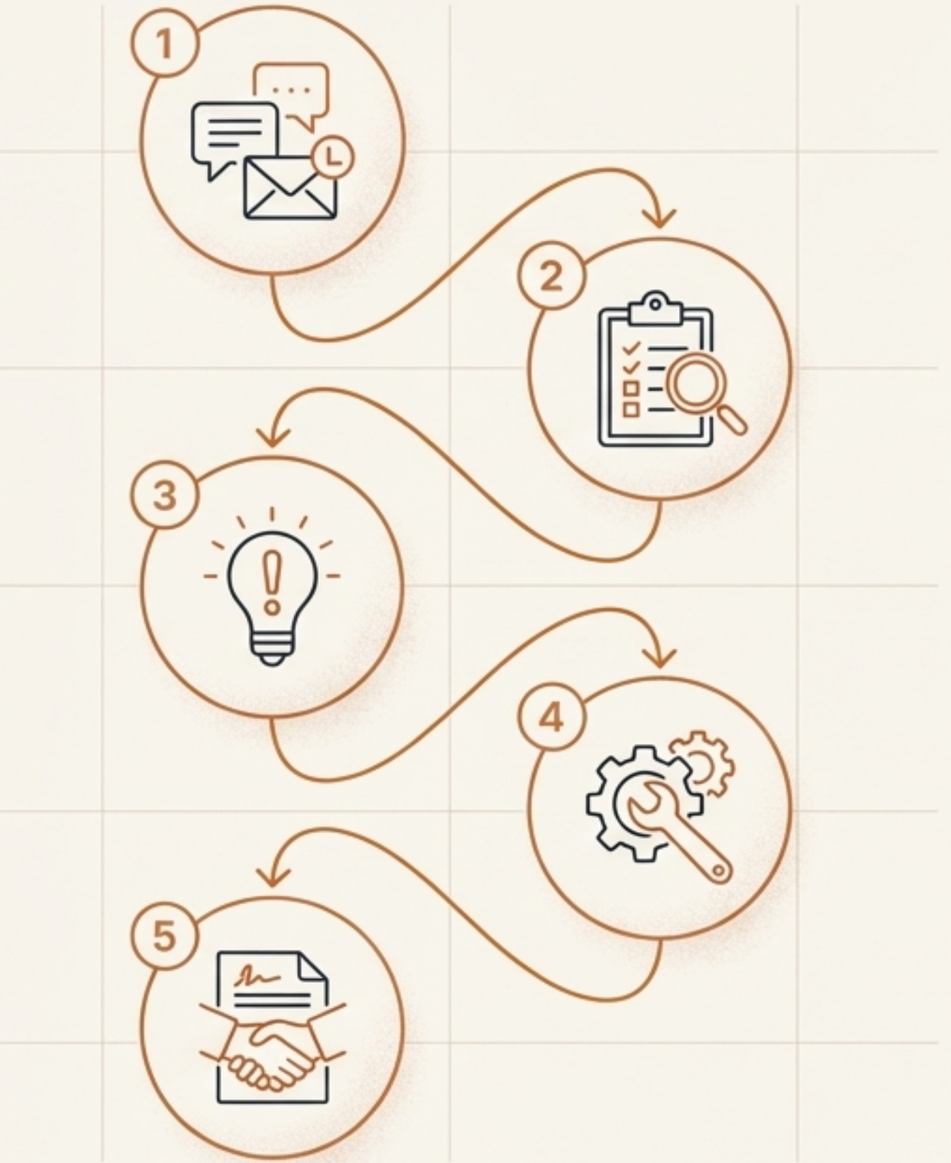
Truth Registries answer: *“Why should I believe it?”*

These are complementary, not competing concepts.

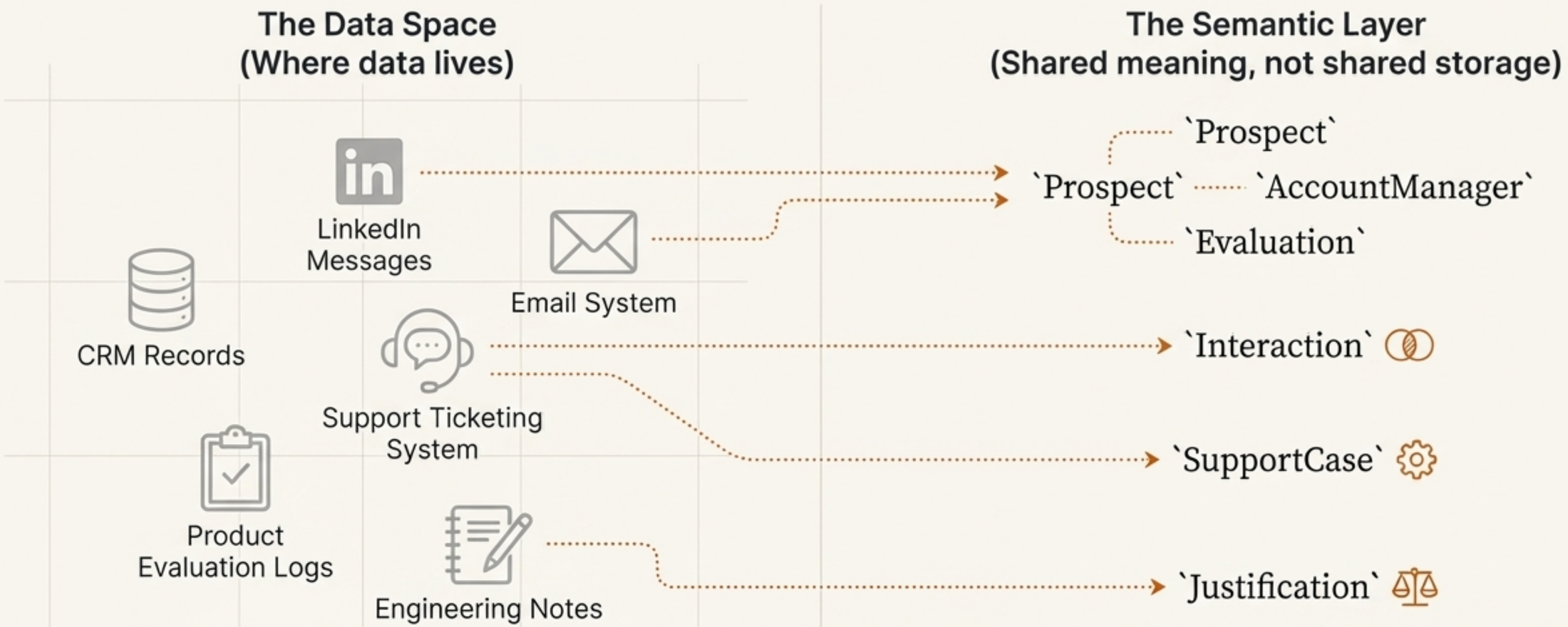
Grounding the Blueprint: A Complex Sales Funnel

An account manager navigates a business development funnel with a key prospect. The process involves:

1. Initial interactions across LinkedIn and email.
2. A formal product evaluation, requiring support from a technical consultant.
3. A support case uncovers a critical product feature gap.
4. The product team creates a new feature to address this gap.
5. A final negotiation where a discount is justified by the insight gained during the support process.



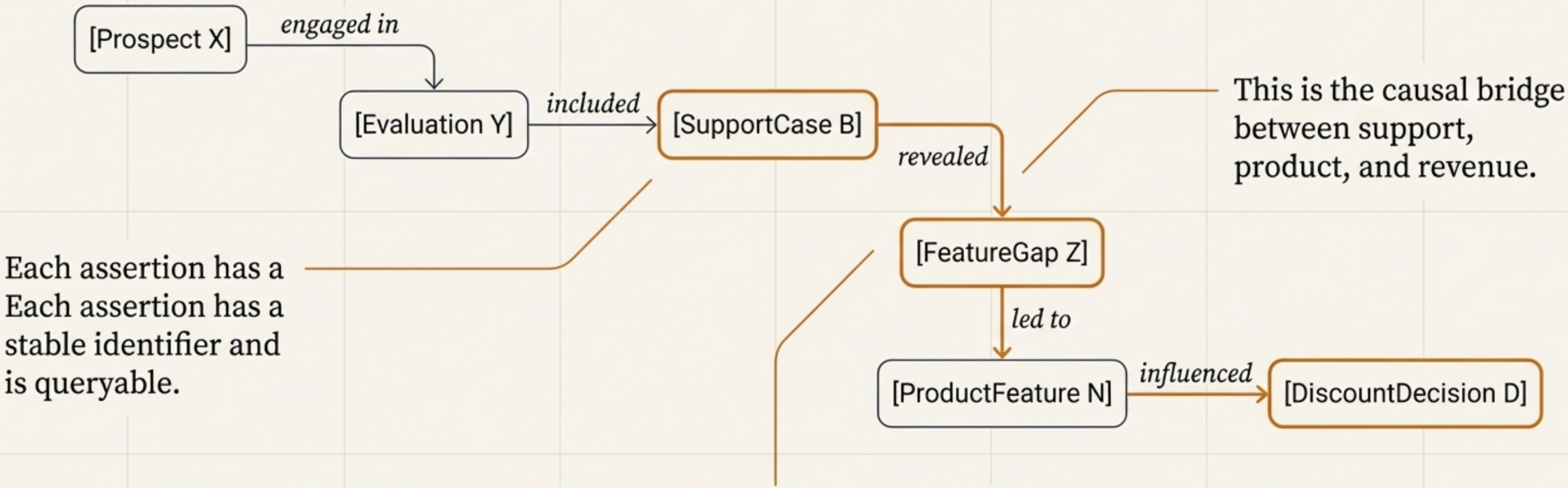
We Begin with a Data Space of Autonomous Systems



No global schema, incremental alignment, federation over centralization.

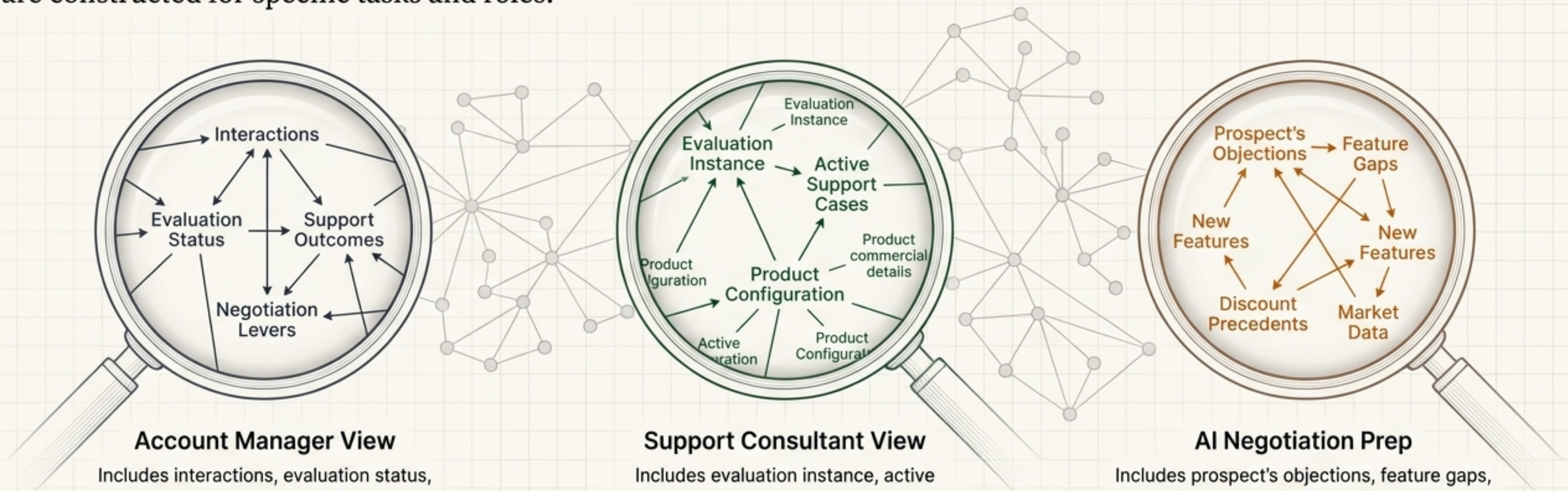
The semantic layer does not move data; it maps meaning onto it.

The Knowledge Graph Materializes Organizational Memory



No One Uses the Entire Graph; Context is Assembled On-Demand

Instead of one view, **ephemeral Context Graphs** are constructed for specific tasks and roles.



Account Manager View

Includes interactions, evaluation status, support outcomes, negotiation levers. *Excludes* internal engineering details.

Support Consultant View

Includes evaluation instance, active support cases, product configuration. *Excludes* commercial negotiation details.

AI Negotiation Prep

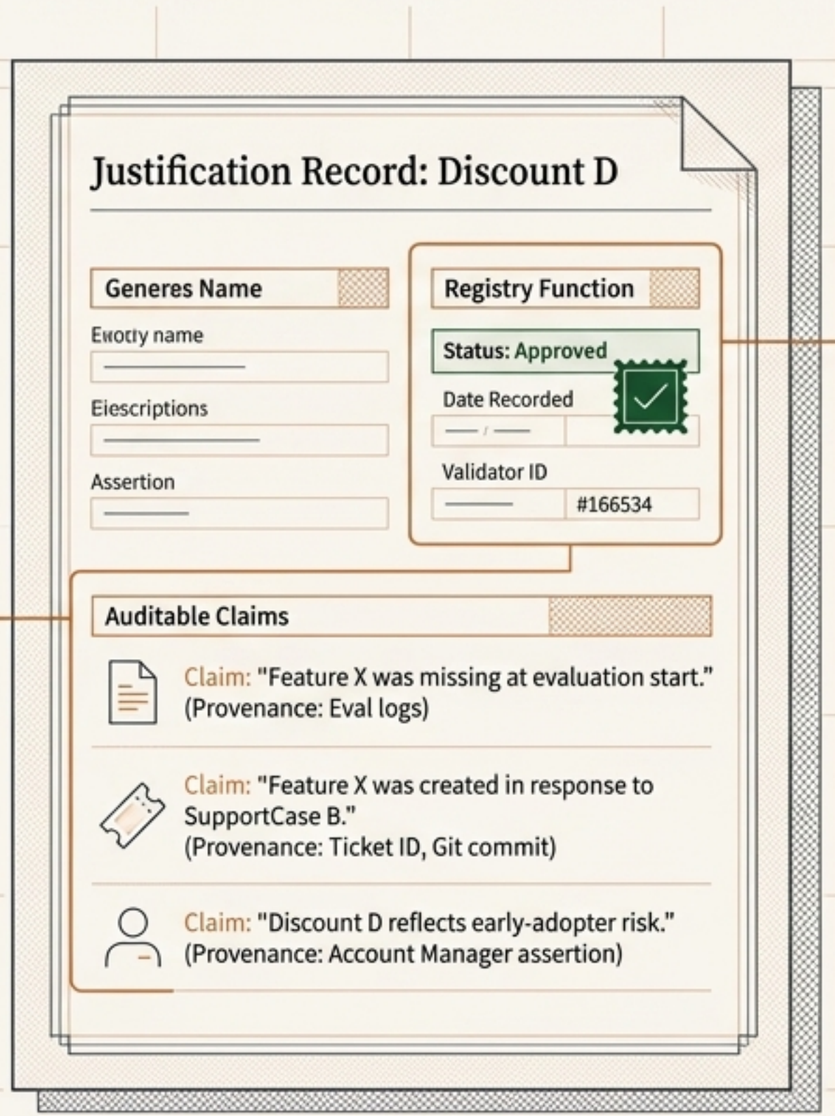
Includes prospect's objections, feature gaps, new features, and discount precedents. A purpose-built graph for a specific task.

Context graphs are projections shaped by intent. They answer, 'What matters *right now?*'

The Truth Registry Records Why a Decision Was Justified

The discount is granted.
It is not just a number in a CRM field.

The justification is backed by a series of auditable **claims:**

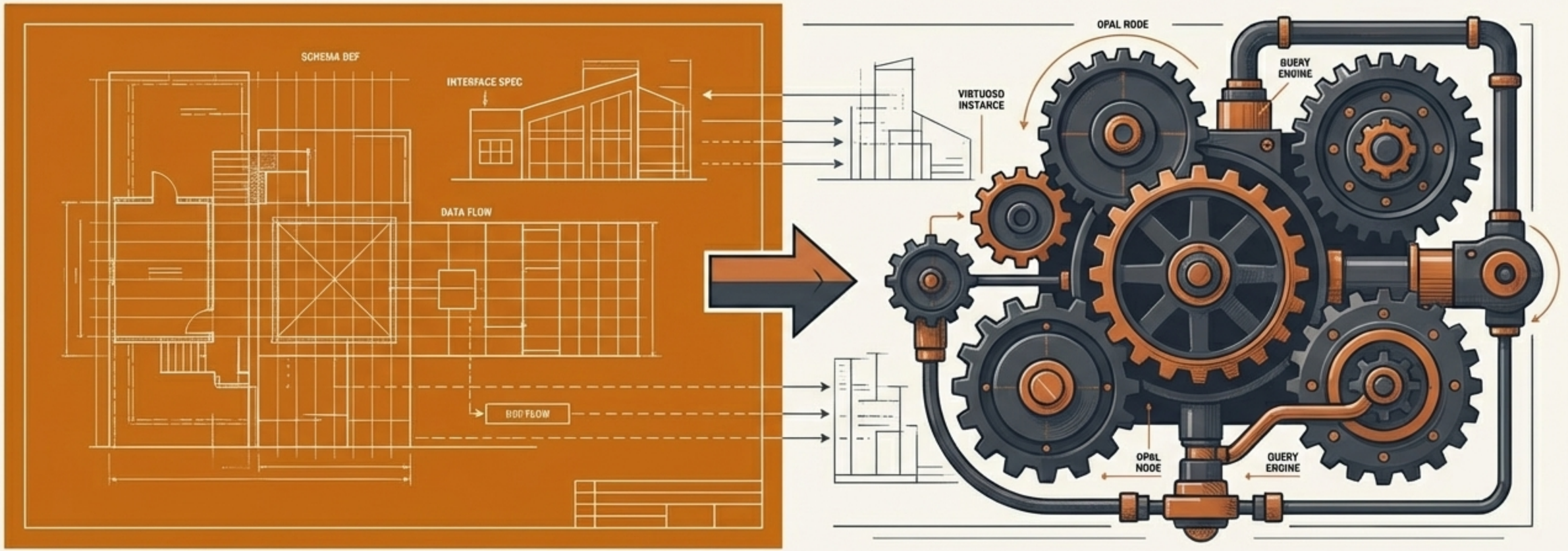


The Role of the Truth Registry:

- It records the claim, its provenance, who asserted it, and its validation status (e.g., Approved).
- It does **not decide truth**; it records the basis for a decision, making it auditable and defensible.

From Blueprint to Reality: A Concrete Implementation

How OPAL and Virtuoso realize this architecture in practice.



We will now show the actual RDF structure and system roles, not just diagrams.

The Knowledge Graph is Partitioned into Named Graphs

We deliberately separate facts, claims, and context into distinct, governable graphs.

Named Graph	Purpose
`:interactions`	LinkedIn + email exchanges
`:evaluation`	Product evaluation process
`:support`	Support cases and outcomes
`:product`	Feature gaps and new features
`:sales`	Negotiation + discount
`:truth-registry`	Claims, provenance, justification

Each graph can have different Access Control Lists (ACLs), enabling fine-grained security.

Code as Evidence: Connecting Support Insights to Revenue

`GRAPH ex:support`

```
GRAPH ex:support {
  ex:supportCase456 a ex:SupportCase ;
    ex:partOfEvaluation ex:evaluation789 ;
    ex:identifiedGap ex:featureGap77 .
}
```

Support case identifies a specific feature gap.

`GRAPH ex:product`

```
GRAPH ex:product {
  ex:newFeature88 a ex:ProductFeature ;
    ex:addressesGap ex:featureGap77 ;
    prov:wasDerivedFrom ex:supportCase456 .
}
```

New feature explicitly addresses the gap and records its provenance, linking it directly to the support case.

Code as Evidence: Modeling a Justified Business Claim

`GRAPH ex:truth-registry`

```
GRAPH ex:truth-registry {
  ex:claim001 a ex:BusinessClaim ;
    ex:claimText "Discount granted due to feature gap..." ;
    prov:wasAttributedTo ex:accountMgr45 ;
    prov:wasDerivedFrom ex:supportCase456 ;
    prov:wasDerivedFrom ex:newFeature88 ;
    ex:claimStatus ex:Approved .
}
```

Auditability

The claim is explicitly linked to the support case and the new feature that justified it.

Reuse

This justification can be surfaced in future negotiations.

AI Grounding

Provides a verifiable anchor for AI reasoning, preventing hallucination.

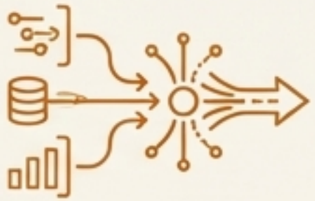
The Engine: Virtuoso Provides the Foundation, OPAL Provides the Intelligence

Virtuoso's Role

Provides:

- Industrial-strength Quad Store (Named Graphs are first-class citizens).
- Row-level and graph-level security.
- Live SPARQL federation.

Core Function:
Acts as the scalable persistence and query layer that "tolerates the reality" of enterprise data.



OPAL's Role (Critical)

Sits between:
LLMs, Tools, and Virtuoso.

Does:

- Assemble context graphs on-the-fly.
- Enforce access controls.
- Bind natural language to SPARQL queries.
- Track provenance of AI outputs.

Core Function:
Acts as the context-aware orchestration layer that makes the data useful for AI.

This is Not a CRM, a KG Demo, or RAG. It is a System for Justified Decisions.

This Architecture Is:

- ✓ A **data space** of autonomous sales, support, and product systems.
- ✓ A **semantic layer** that aligns meaning without centralization.
- ✓ A **knowledge graph** that preserves institutional memory.
- ✓ **Context graphs** assembled just-in-time for humans and AI.
- ✓ A **truth registry** that explains *why* revenue decisions were justified.

Data Spaces hold the raw interactions, the **Semantic Layer** gives them shared meaning, the **Knowledge Graph** preserves what happened, **Context Graphs** surface what matters now, and the **Truth Registry** explains why key decisions—like a discount—were justified and defensible.