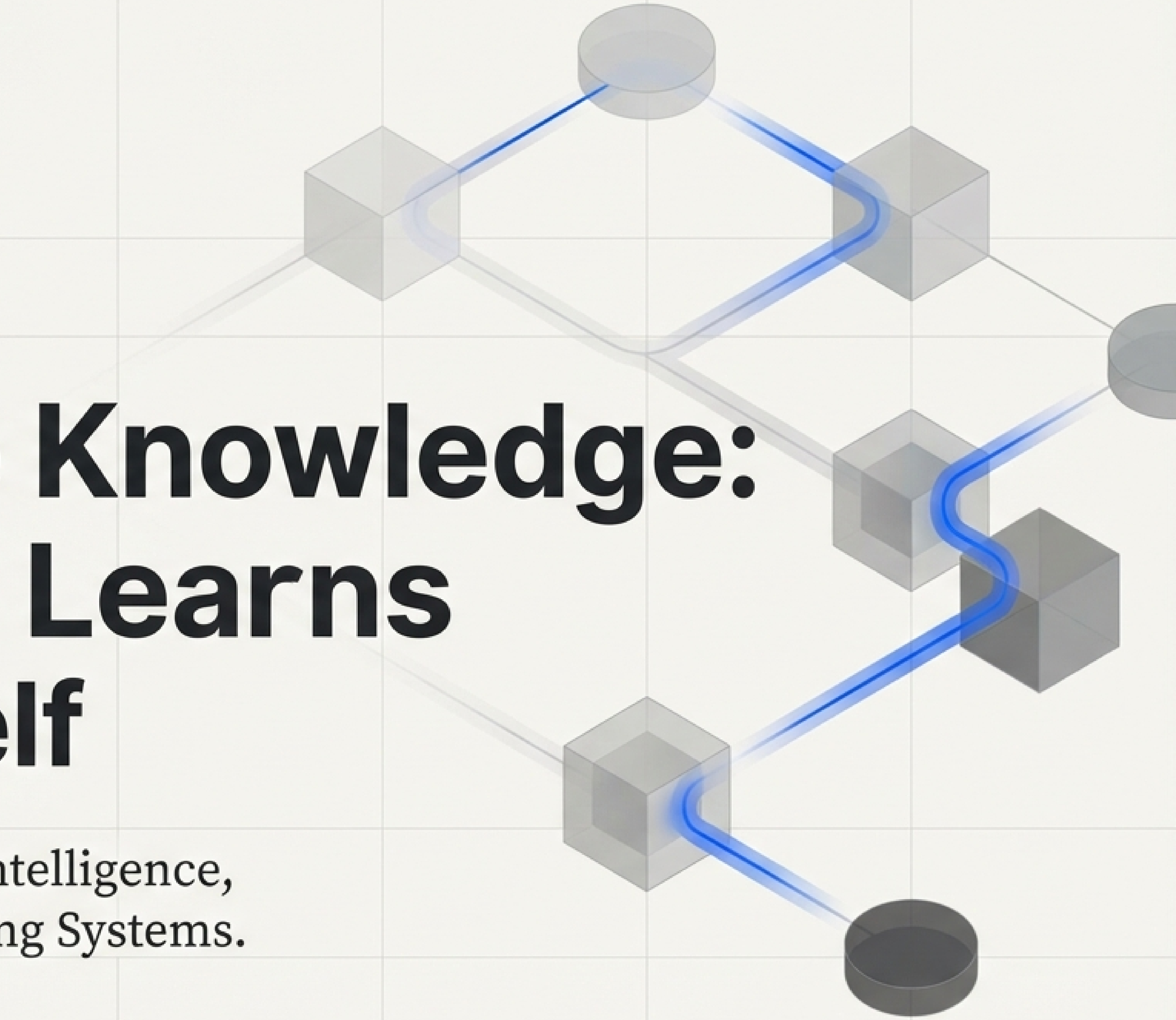
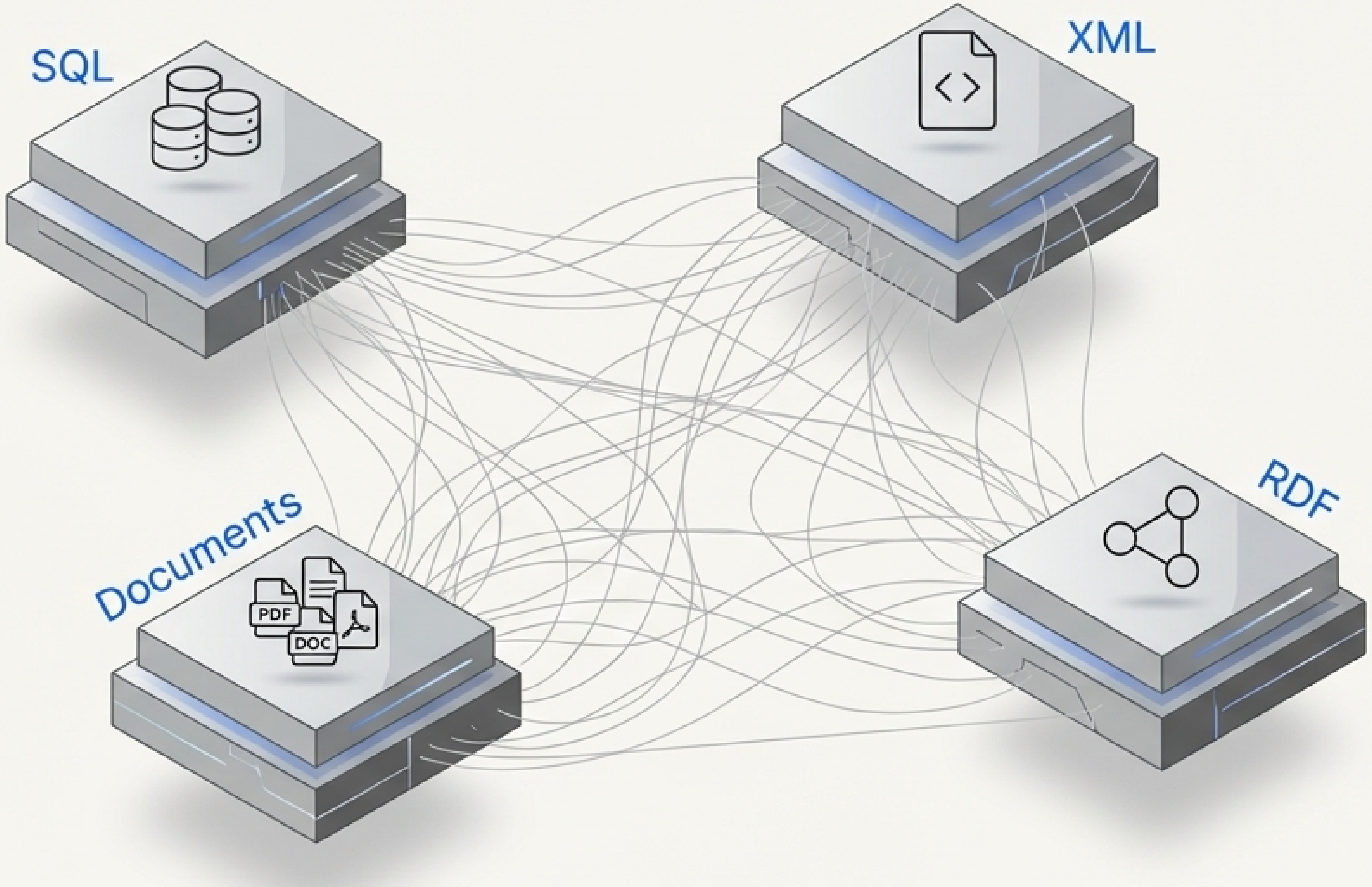
Executable Knowledge: When Data Learns to Fetch Itself

A new paradigm for Business Intelligence, Mixed Reality, and Self-Managing Systems.



Our digital world is built on disconnected “data islands.”



We live with datasets created independently, in heterogeneous formats like SQL, XML, RDF, and documents. They are globally distributed and autonomously managed.
Integrating this data, or "mashing," remains a brute-force, manual effort.

Semantic Technology’s role has been to build "bridges" between these islands, creating Linked Data.

Our windows into this data—BI and AR—provide a static view of a dynamic reality.

Business Intelligence

BI systems typically rely on pre-integrated data warehouses. The analytics answer “what happened?” based on stale data, not “what is happening *now*?”

Augmented/Mixed Reality

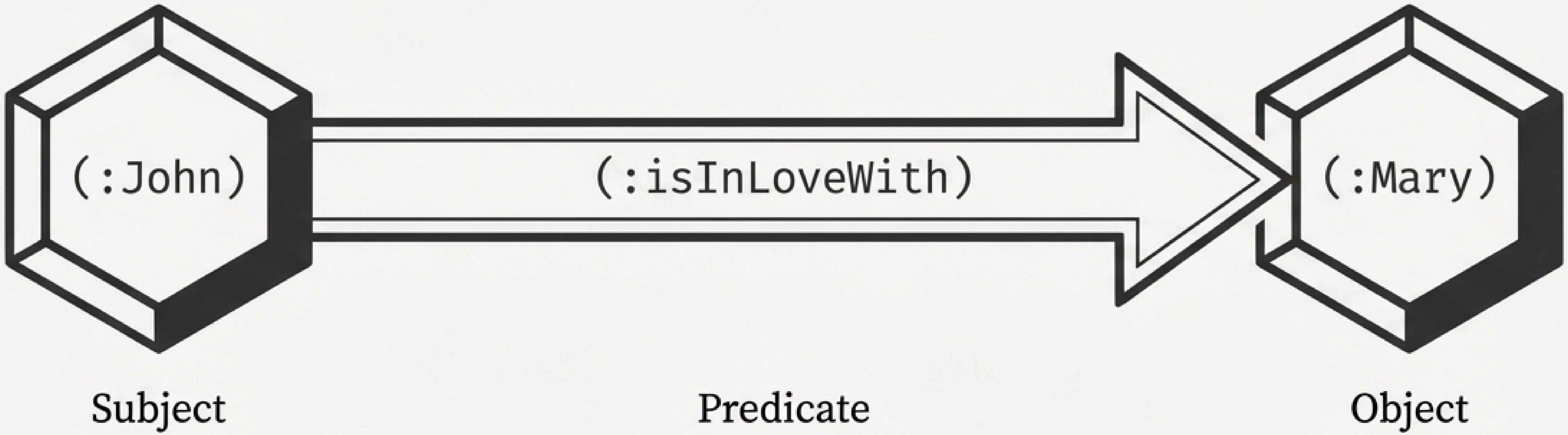
AR applications often display pre-loaded, static points of interest. This provides a layer of old information, not a live, interactive view of the world. It enriches the real world, but doesn’t interact with it in real time.

Data from Last Quarter.

Information last updated: 2 weeks ago.

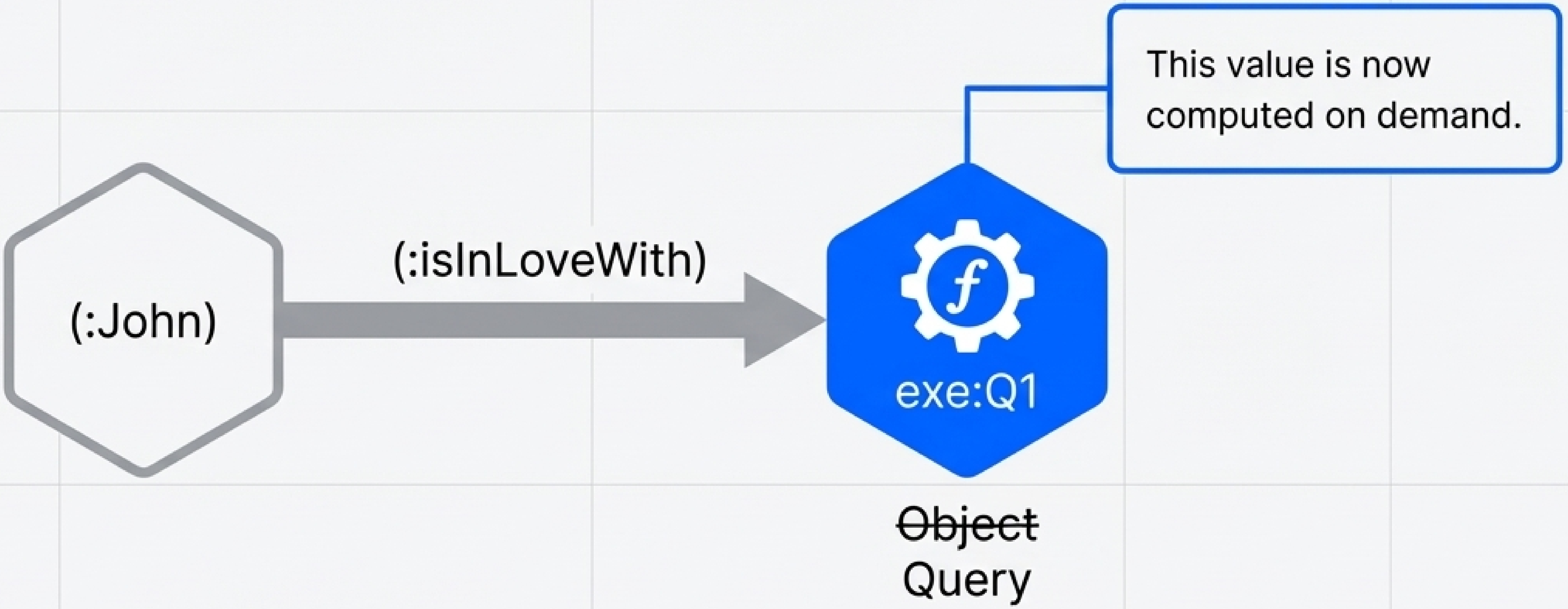
The foundational flaw lies in the nature of the link itself.

The standard semantic triple, `Subject - Predicate - Object`, is a statement of declarative knowledge. The `Object` is a static, literal value or a link to another resource. It represents a fact that was true at a specific point in time. It describes *what is*.



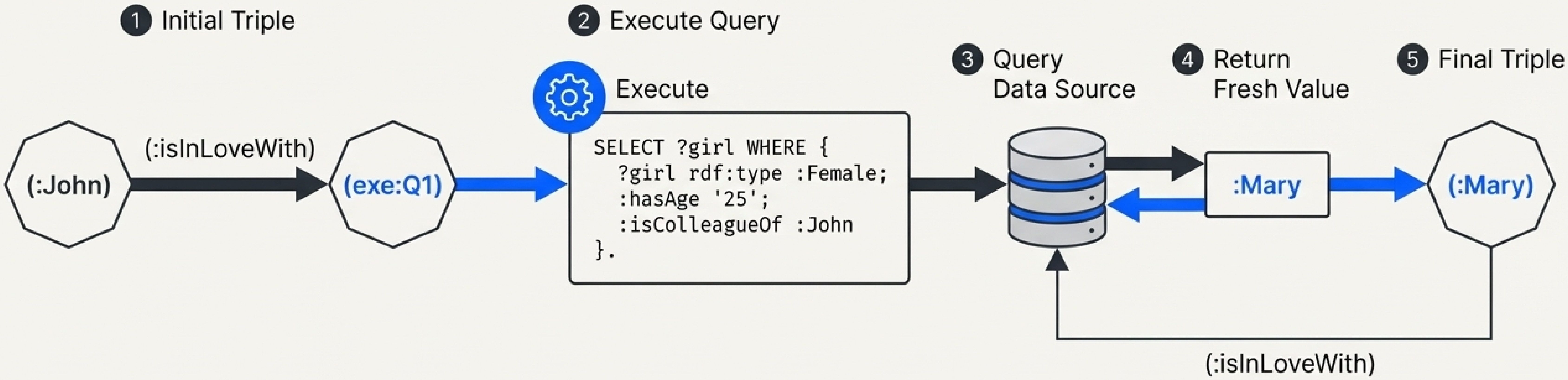
What if a link was not a fact, but a recipe for finding the fact?

We propose a new property type: the "executable property." This transforms the semantic triple into a new structure: `Subject - Predicate - Query`. The value is no longer stored; it is computed 'on-the-fly" when requested. This hybridizes declarative knowledge with procedural knowledge—it describes *how to find out*.



This value is now computed on demand.

The knowledge graph becomes a dynamic, computational engine.



❶ Initial Triple

❷ Execute Query

❸ Query Data Source

❹ Return Fresh Value

❺ Final Triple

(:John) (:isInLoveWith) (exe:Q1)

Execute

```
SELECT ?girl WHERE {
  ?girl rdf:type :Female;
  :hasAge '25';
  :isColleagueOf :John
}.
```

:Mary

(:Mary)

(:isInLoveWith)

Key Insight: If the underlying data changes autonomously (e.g., Anna turns 25), the *same* executable statement will yield a *new* result without any modification to the knowledge graph itself.

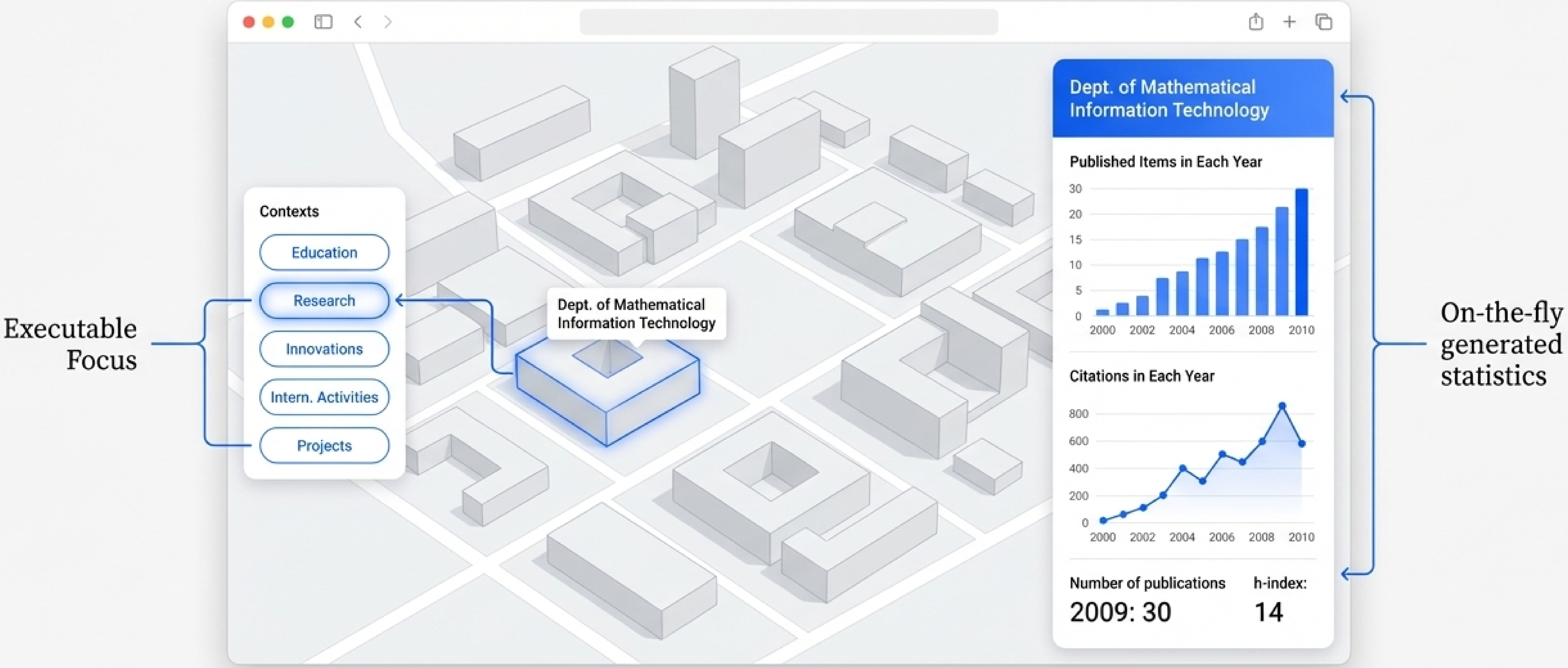
Application 1: “Executable Reality” bridges the physical world with live, intelligent data.

"Executable Reality" is an enhancement of Mixed Reality. Instead of just retrieving pre-existing data about an object, pointing a device at it triggers an "Executable Focus." This invokes on-demand BI computation, delivering real-time analytics tailored to the user’s context.

Live Performance Metrics
400 kV
450 kV
420 kV
400 kV
350 kV
300 kV
400 kV
6:00
8:00
10:00
12:00
14:00
16:00
18:00
● Voltage
● Current
● Frequency
Pig view
Current Load vs. Capacity
0
20%
40%
60%
85%
85%
106%
85%
Predictive Maintenance Forecast
Insulator Check
Connection Tightening
Task nut estimatte needs
60k
40k
20k
0
Next 7 Days
Next 30 Days
Next 7 Days

Delivering context-aware analytics, instantly.

A user views a university campus map. They click on a building (the "Executable Focus") and select a context, such as "Research." This action triggers a pre-defined executable property that formulates and runs a query to generate fresh statistics on the researrch performance of that specific department (e.g., publications per year, citations, h-index).



Executable Focus

Contexts

Education

Research

Innovations

Intern. Activities

Projects

Dept. of Mathematical Information Technology

Dept. of Mathematical Information Technology

Published Items in Each Year

Citations in Each Year

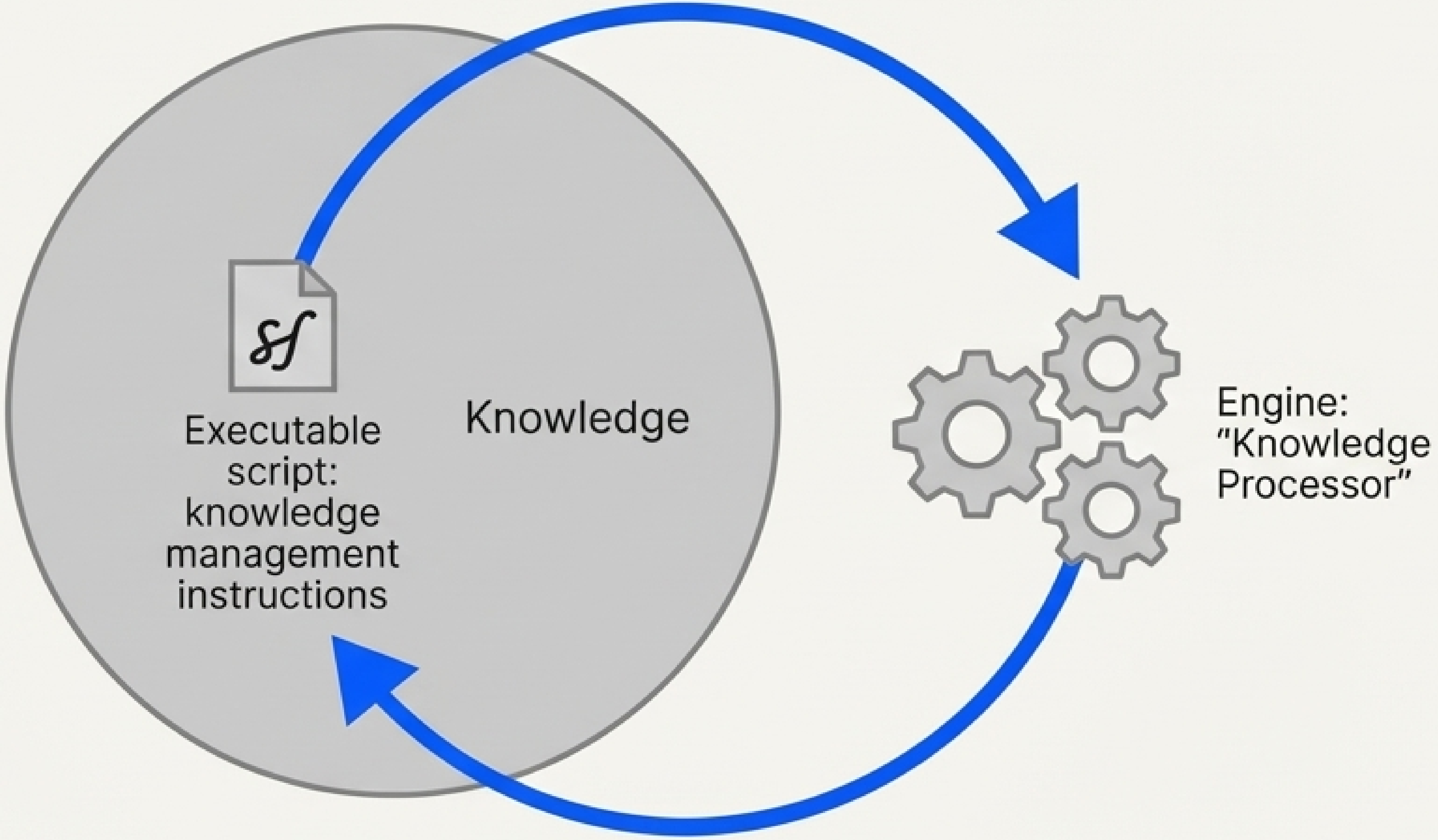
Number of publications 2009: 30

h-index: 14

On-the-fly generated statistics

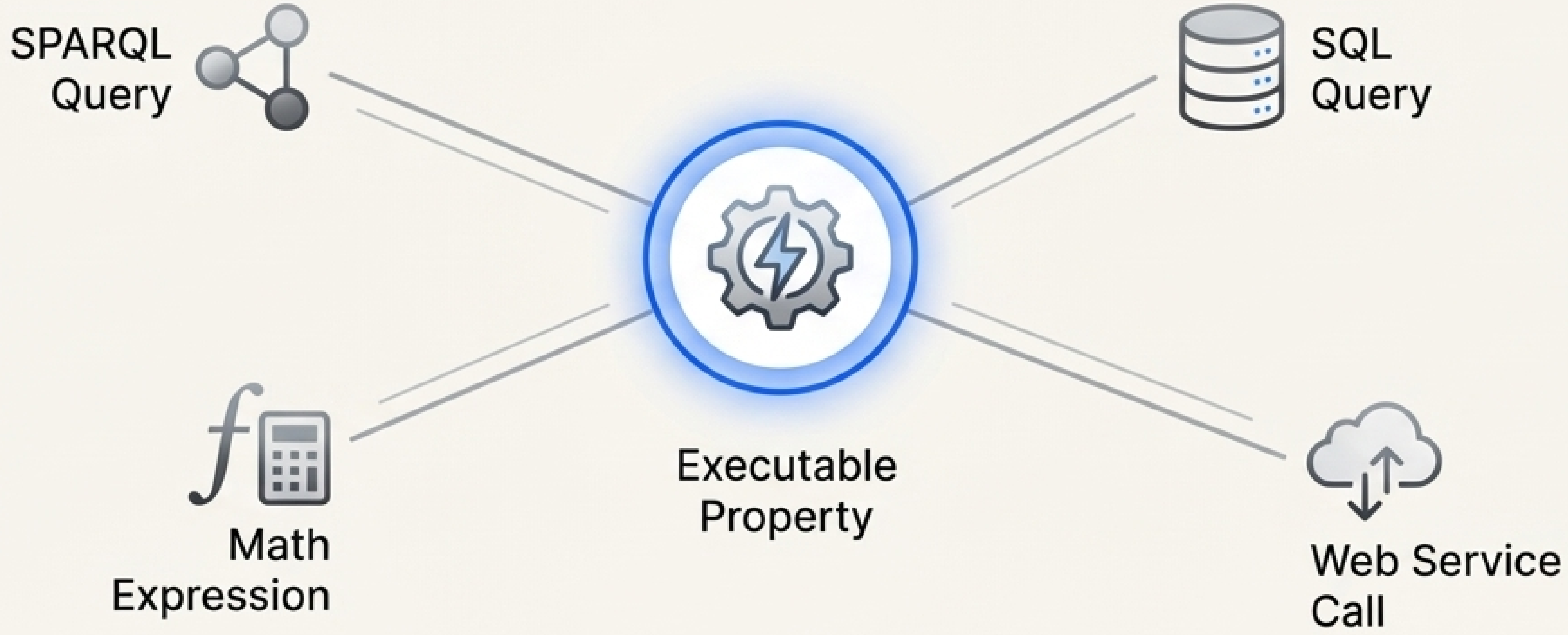
Application 2: “Knowledge Computing” enables self-managing knowledge ecosystems

Executable Knowledge is the foundation for “Knowledge Computing.” The knowledge base itself contains the explicit, executable instructions for its own management. This enables autonomic characteristics like self-configuration, self-healing, and self-optimization. The knowledge becomes a proactive, evolving entity capable of managing its own complexity and integrity.


Executable script: knowledge management instructions
Knowledge
Engine: "Knowledge Processor"

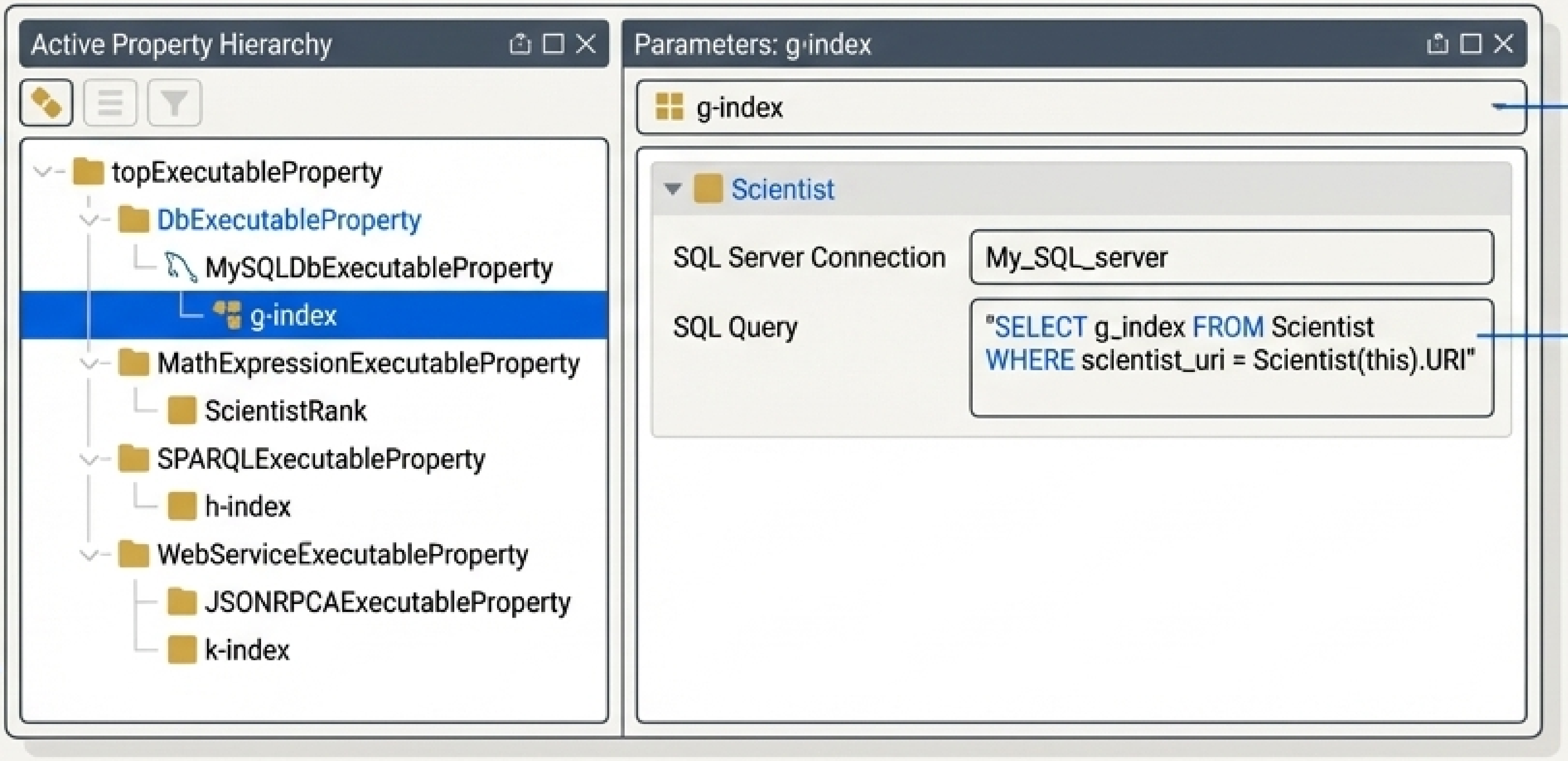
One model to query any source.

The `query` within an executable property is not limited to one language or protocol. It can be a SPARQL query to an RDF store, an SQL query to a relational database, a call to a Web Service (REST, SOAP, etc.), or even a mathematical expression. This abstracts the heterogeneity of backend sources into a clean, semantic link.



Proof of Concept: An implementation in Protégé

The Executable Knowledge model was implemented as a plug-in for the Protégé ontology development environment. This allows for the definition of new executable properties directly within the ontology. For a `Scientist` class, properties like `h-index`, `g-index`, and `ScientistRank` can be defined as executable.



```
topExecutableProperty
  DbExecutableProperty
    MySQLDbExecutableProperty
      g-index
  MathExpressionExecutableProperty
    ScientistRank
  SPARQLExecutableProperty
    h-index
  WebServiceExecutableProperty
    JSONRPCAExecutableProperty
    k-index
```

g-index

Scientist

SQL Server Connection: My_SQL_server

SQL Query: "SELECT g_index FROM Scientist WHERE sclentist_uri = Scientist(this).URI"

Executable Property Hierarchy

Query Type Definition

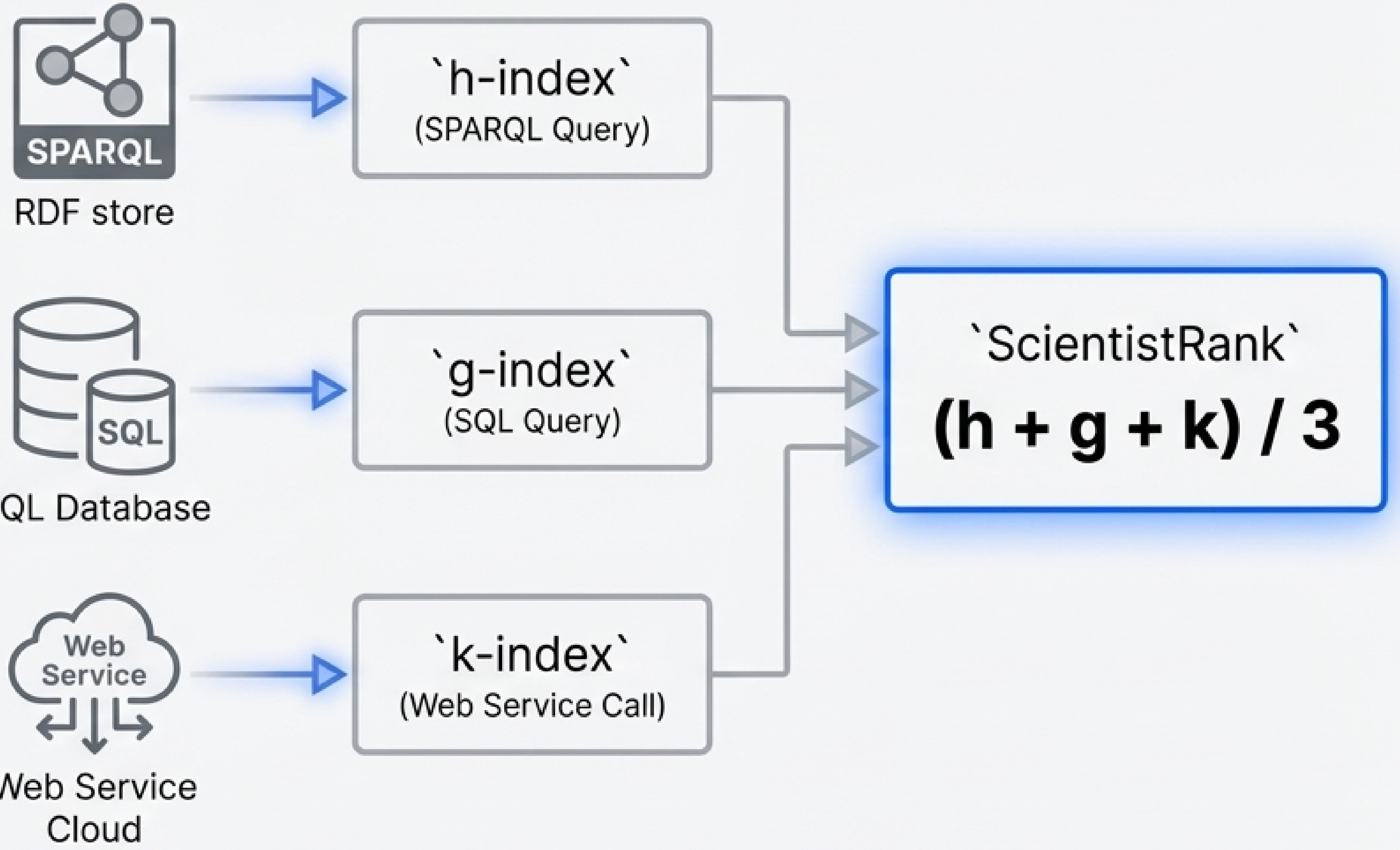
Query Parameters

NotebookLM

A dynamic, multi-source calculation in action.

A composite rank for a scientist is calculated dynamically:

- **h-index** is computed via a **SPARQL query** against a semantic repository.
- **g-index** is retrieved from a relational database via an **SQL query**.
- **k-index** is obtained by calling a remote **Web Service (XML-RPC)**.
- **ScientistRank** is then calculated as a **Mathematical Expression** that averages the other three indexes: **(h-index + g-index + k-index) / 3**.

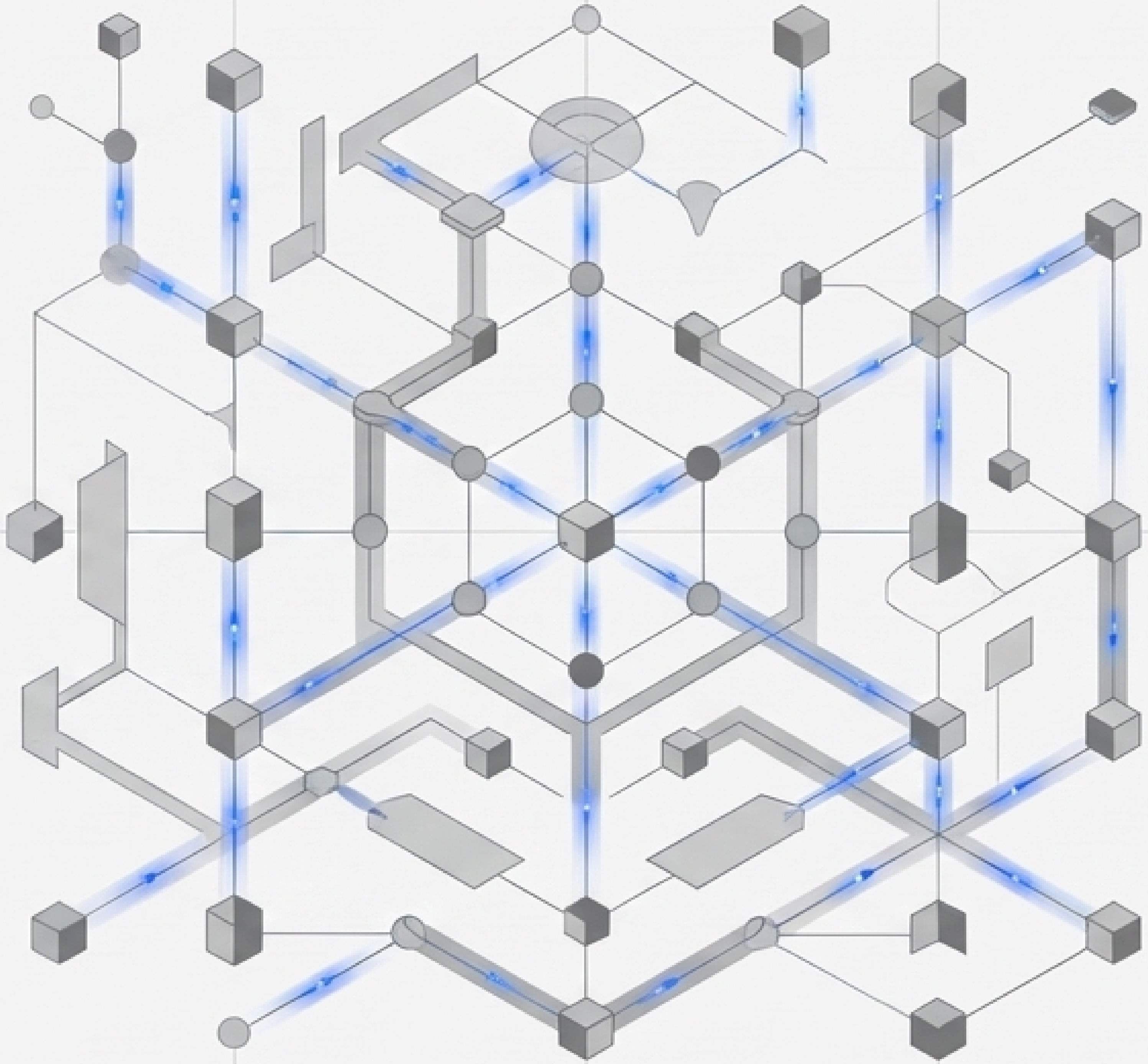


A fundamental shift from “what is” to “how to find out.”

Declarative Paradigm (Old)	Executable Paradigm (New)
Knowledge is static	Knowledge is dynamic
Links are brittle and point to values	Links are resilient and point to procedures
Requires manual, brute-force integration (“mashing”)	Handles heterogeneity natively (“meshing”)
Data becomes stale over time	Data is always current on demand
Operates on a “Closed World Assumption” for computation	Hybridizes procedural knowledge within a declarative framework

The future is executable: Enabling a more intelligent and self-managed digital world.

- **For Business Intelligence:** Delivers real-time, on-demand insights instead of relying on historical data.
- **For Mixed Reality:** Creates truly “live” and interactive augmented views of the physical world.
- **For Data Systems:** Paves the way for autonomic knowledge ecosystems that are self-configuring, self-healing, and self-optimizing.



References & Further Reading

Terziyan, V., & Kaykova, O. (2012). From Linked Data and Business Intelligence to Executable Reality. *International Journal on Advances in inIntelligent Systems, 5*(1&2), 194–208.

Terziyan, V., Shevchenko, O., & Golovianko, M. (2014). An Introduction to Knowledge Computing. *East-European Journal of Advanced Technologies,* 1/2(67).