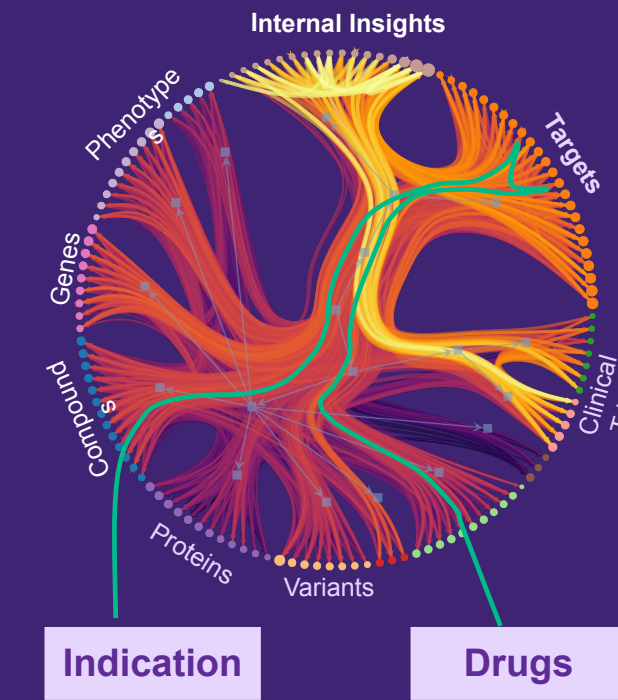




ABSTRACT & BACKGROUND

From scattered data to decisions

Drug discovery data is scattered. Public databases and internal results differ in quality and structure, so each new question means more manual data work, and AI tools lack the context they need.

Ardigen builds knowledge graphs that close this gap. We merge public and proprietary data into one graph, shaped around each client’s disease area and questions, ready for analytics, machine learning and AI agents.

ArdKG shows this approach at work. Following the PrimeKG schema¹, it links **~140,000 entities** of 10 types. Its chat app offers four ways to ask: plain-English questions, target ranking, link prediction (**MRR 0.30** on held-out data) and a literature agent whose every citation is checked against the papers it actually found. The result: testable hypotheses for target discovery and indication expansion.

METHODS / TECHNOLOGIES

Knowledge graph build

- PrimeKG schema¹: 10 entity types, 30 relation types
- Scripted build into Neo4j; ships as a Docker image with the data pre-loaded

Target prioritization via network centrality (Fig. 4)

- Personalized PageRank² seeded on a disease, drug or other asset
- Relative relevance = personalized ÷ global score, so specific targets beat generic hubs

Indication expansion via link prediction (Fig. 5)

- ComplEx-N3^{3,4} embeddings (PyKEEN⁵), tuned with Optuna / Ray Tune, tracked in MLflow
- Scores missing drug–indication, drug–target and target–disease links, each flagged known or new

Governed agentic reasoning (Figs. 3, 6)

- Plain-English questions become LLM-written, read-only Cypher queries
- Literature agent searches PubMed via MCP⁶; each cited DOI/PMID must match a retrieved paper
- An MCP server opens the graph to any AI agent

Deployment & tooling

- Dockerized Neo4j plus Python REST APIs for the prediction and agent servers
- On-premise or in a private cloud; data and models stay with the client

RESULTS AND DISCUSSION

Results

- Link prediction: **MRR 0.30** on a random 10% test split (1.36M triples); 1.0 = true answer always ranked first
- Felodipine: known targets and uses fill the top 8; **9 of the top 23** are new candidates, e.g. CACNB1 (Fig. 5)
- Ranking lifts disease-specific genes above generic hubs: NPHS2 rises from ~25,000th to ~1,250th (Fig. 4)

Discussion

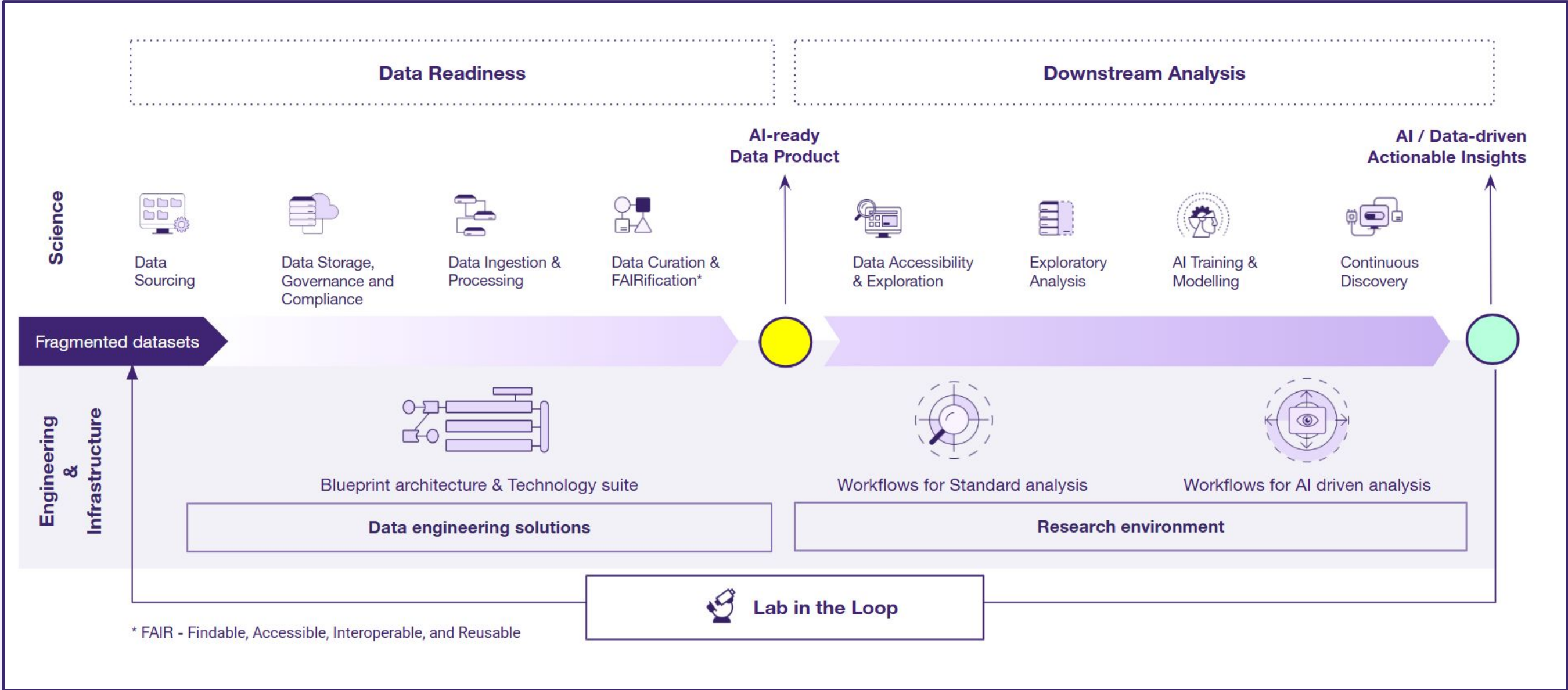
- One reusable graph can replace a new data integration for each question
- Predictions carry no explanation; graph queries and the literature agent supply the evidence to review them
- Limitations:** public data only; the random split likely flatters MRR; some candidates are implausible, so expert review stays essential
- Next:** add client data and tune models to it; a stricter test with both directions of a link in one split

REFERENCES

- Chandak P, Huang K, Zitnik M. Building a knowledge graph to enable precision medicine. *Sci Data*. 2023;10:67.
- Haveliwala TH. Topic-sensitive PageRank. *WWW* 2002.
- Trouillon T, et al. Complex embeddings for simple link prediction. *ICML* 2016.
- Lacroix T, Usunier N, Obozinski G. Canonical tensor decomposition for knowledge base completion. *ICML* 2018.
- Ali M, et al. PyKEEN 1.0: a Python library for training and evaluating knowledge graph embeddings. *JMLR*. 2021;22(82):1–6.
- Model Context Protocol specification. modelcontextprotocol.io

What it takes to put a knowledge graph to work

Knowledge graphs sit at the center of Ardigen’s data-to-decision journey



Proprietary data stays disconnected

Internal results sit outside the wider biomedical context, so much of their value goes unused.

Every question becomes a new integration

Scientists wait for specialist teams to assemble data, context and analyses.

The KG stops at search

Without analytics and workflows on top, adoption fades and AI lacks traceable context.



Prioritize with confidence

Rank targets, indications and repurposing options with connected evidence



Reveal non-obvious biology

Surface hidden gene–disease–drug–pathway links across public and internal data



Scale across programs

Reuse one knowledge layer across assets, therapeutic areas and new questions



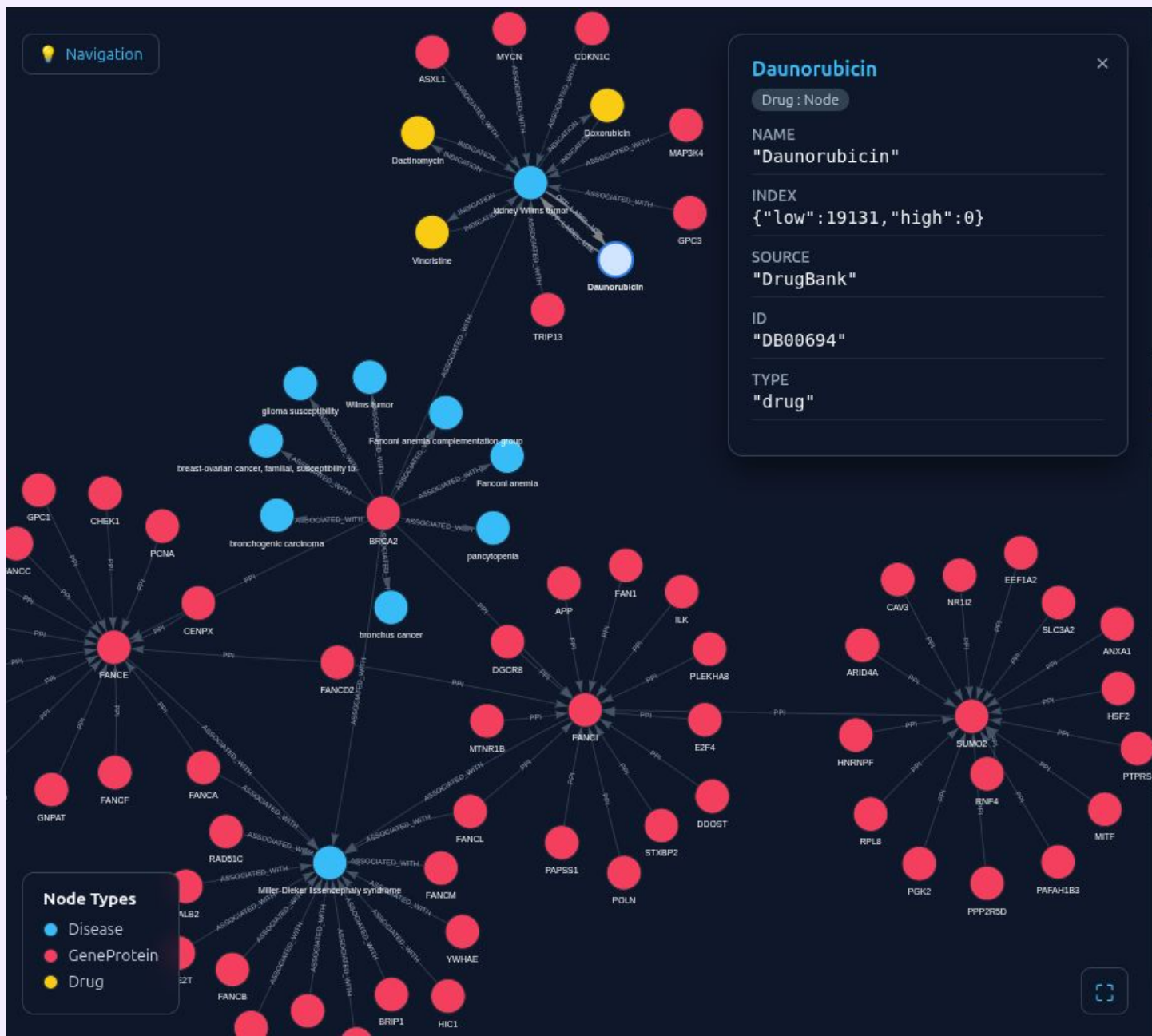
Keep every answer defensible

Trace each insight back to its source for review and go/no-go decisions

Figure 1: Ardigen’s data-to-decision journey. Scattered datasets become an AI-ready data product, such as a knowledge graph, that feeds analysis, AI modeling and continuous discovery; lab results loop back in.

Ask in plain English

Type a question; ArdKG writes the graph query and draws the answer



- No query language needed
- The generated query is checked as read-only before it runs
- Answer written from the query results, not from the model’s memory
- Drag, zoom and expand any cluster
- Click a node to see its identifiers and source database

Figure 3: Answer drawn as an interactive graph, here around the drug daunorubicin. Entity types are color-coded; clicking a node opens its full record and source.

Predict missing links

Pick an entity; the model proposes links the graph does not have yet

Predicted entity	Type	Relation	Score	In graph?
Raynaud disease	disease	off-label use	37.9	✓ known
CACNB2	gene/protein	target	26.0	✓ known
TNNC2	gene/protein	target	21.7	✓ known
Prinzmetal angina	disease	off-label use	19.6	✓ known
CACNA1H	gene/protein	target	18.8	✓ known
lymphosarcoma	disease	off-label use	17.0	new
CACNB1	gene/protein	target	11.3	new

- Known targets and uses rank at the top, as expected
- New candidates are flagged, e.g. CACNB1, another calcium-channel subunit
- Each new link is a hypothesis for expert review, not a finding

Figure 5: Link predictions for felodipine, a calcium-channel blocker: 7 of its top 23 results. Higher score = more plausible; ‘new’ = link not yet in the graph.

How ArdKG works

One graph, four ways to ask

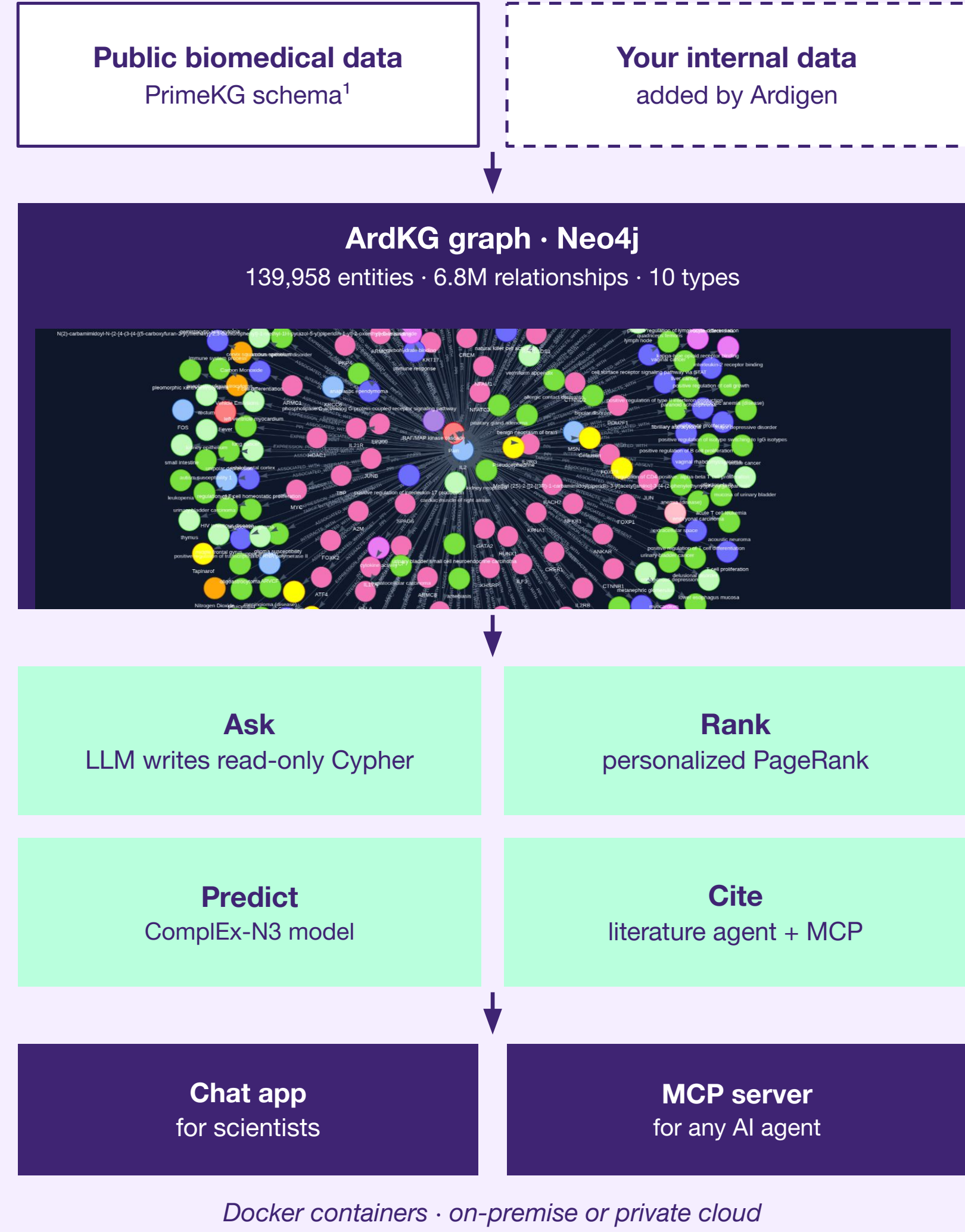


Figure 2: ArdKG architecture. Public data (and, for clients, internal data) feed one Neo4j graph. Four engines answer questions in the chat app; an MCP server also opens the graph to external AI agents.

Rank what matters

Pick a disease; see which entities matter specifically to it, not just in general

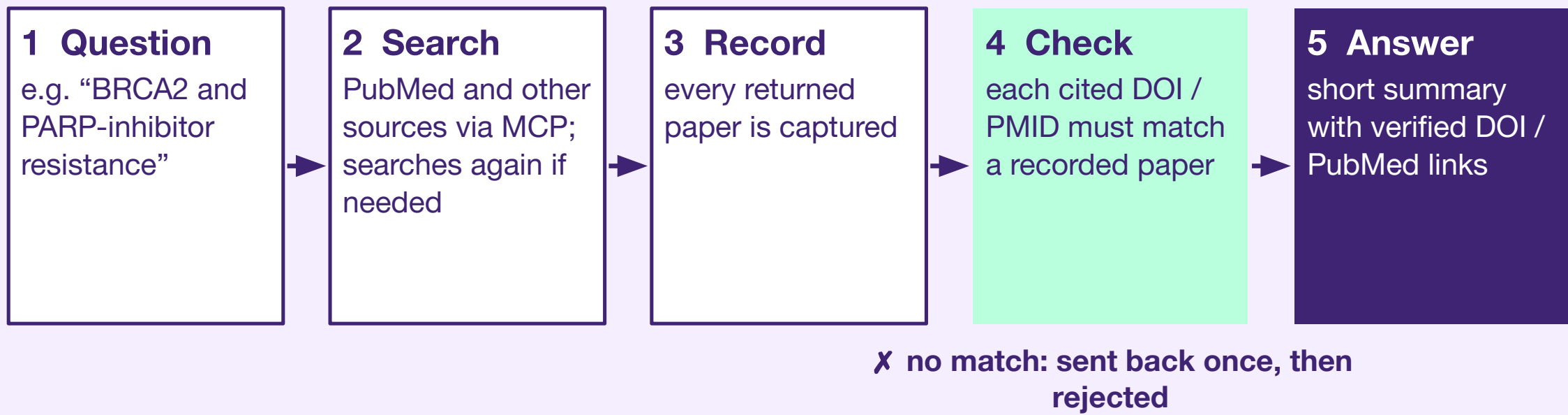
Entity	Type	Relative relevance	Global rank	Rank for query
NPHS2	gene/protein	26.5	24,997	1,257
NES	gene/protein	15.5	9,571	1,283
HPSE	gene/protein	15.4	9,375	1,266
DES	gene/protein	12.0	5,498	1,271

- Relative relevance:** how many times more relevant an entity is to the query than to the graph overall
- NPHS2 climbs from rank ~25,000 overall to ~1,250 for this query: specific to the disease, not a generic hub
- Works for any entity type: targets, drugs, phenotypes, pathways

Figure 4: Example ArdKG output: gene/protein entities ranked for a disease query by relative relevance (personalized ÷ global PageRank).

Cite the literature

Ask a research question; get a short answer with sources you can open



- Answers rely on retrieved papers, not the model’s memory
- Each search step is shown live in the app
- Easy to extend with more sources through MCP

Figure 6: How the literature agent answers. The citation check reads the recorded search results, not the model’s claims, so every cited DOI or PMID points to a paper the agent actually retrieved.

IMPACT

Your data, in context, ready for your scientists and AI agents

A graph built around your questions

Public knowledge plus your internal data, with a schema shaped to your disease area

Models tuned to your data

Pre-built graph models, adapted to your data and platform constraints

Or on the graph you already have

We add ranking, prediction models and agents on top of your existing knowledge graph

Your IP, your infrastructure

On-premise or in your private cloud; data, models and results stay yours

