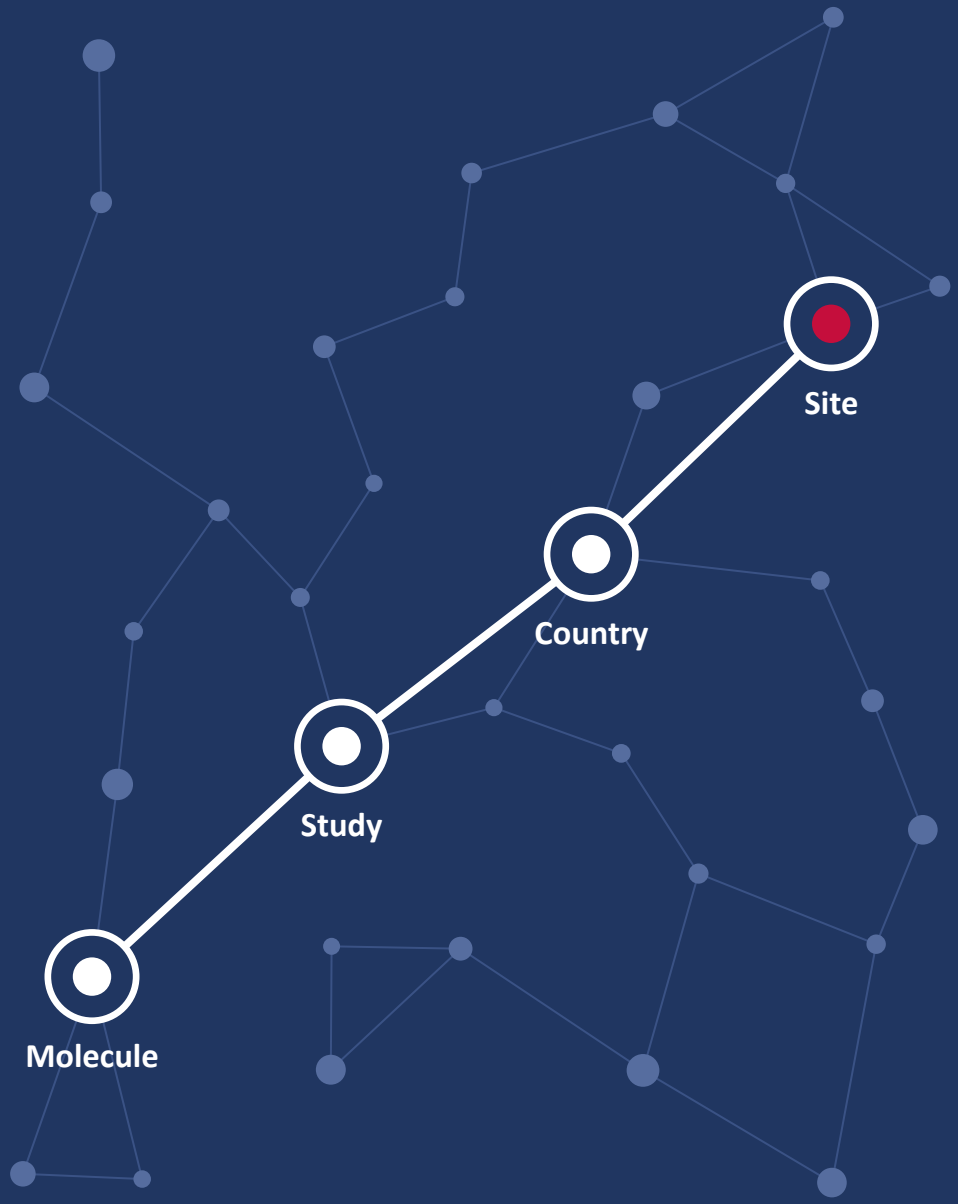
AIDDD 2026 · BOSTON · CLINICAL TRIALS TRACK

Contextualize fragmented data using GRAPHRAG for clinical trial planning.

Turning a trial knowledge graph into country, site and enrollment decisions

Mukul Virmani
Director, Clinical Data Science & Trial Analytics
Gilead Sciences · October 29, 2026



The views expressed are my own and do not necessarily represent those of Gilead Sciences.

THE PROBLEM

Study planning still runs manually via SaaS



Each plan starts over

Benchmarks are re-pulled and re-cleaned for every new protocol.

Precedent stays tacit

Which sites delivered on similar studies lives in people's heads.

Sources disagree

CTMS, EDC and the warehouse report different enrollment counts.

Tufts CSDD
150 trials · ~16,000 sites
reported 2013

11% of sites enrolled no patients

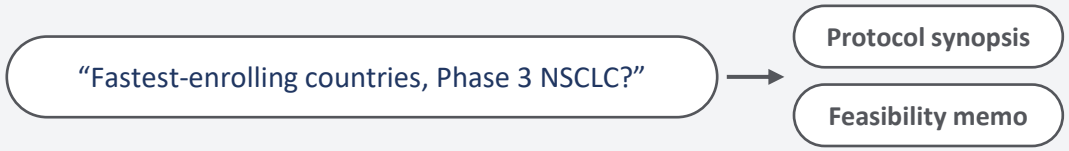
37% of sites under-enrolled

~2× typical enrollment time vs. plan

WHY NOT JUST RAG?

Flat retrieval breaks on planning questions

Vector RAG retrieves similar text



✗ ***Neither document holds the answer***

- Finds documents about enrollment, not performance
- No hierarchy: misses lung adenocarcinoma studies
- Can't aggregate across studies, countries, sites

A graph traverses and aggregates



Roll up concepts → walk to studies → aggregate PSM on the edges

- Hierarchies pull in every relevant study
- Multi-hop traversal plus aggregation, with provenance
- Vectors still find the entry point: hybrid

WHAT THE EVIDENCE SAYS: STRUCTURE HELPS, AND GRAPHS WIN ON MULTI-HOP

16% → 54%

GPT-4 accuracy on enterprise questions: raw SQL schema vs. knowledge graph

Sequeda, Allemang & Jacob, 2023

0.31 → 0.53

Llama-2-13B on hard biomedical questions: no retrieval vs. KG-RAG (SPOKE)

Soman et al., Bioinformatics, 2024

Multi-hop

Graphs win on multi-hop reasoning; plain RAG holds up on single-fact lookups

Han et al., 2025 · GraphRAG-Bench, ICLR 2026

THE IDEA

Study planning is a graph problem

Precedent

Every completed study, with its countries, sites and timelines

Performance on edges

PSM, enrollment duration, activation and start-up times



Clinical meaning

MeSH · NCIt · UMLS link indications and drug classes

Similarity

Graph-aware embeddings find analogous studies

THE OPERATING RULE

AI at the edges

Interpret the ask, retrieve precedent, explain the answer

Deterministic at the core

Graph queries, optimizers, forecasting models

The study team decides

People own the plan and its trade-offs

01

PART 1 OF 3

Build the graph

From a clinical trials database to a domain-enriched knowledge graph

01 Build the graph — 02 Plan the study — 03 Make it trustworthy

From 40+ tables to a property graph

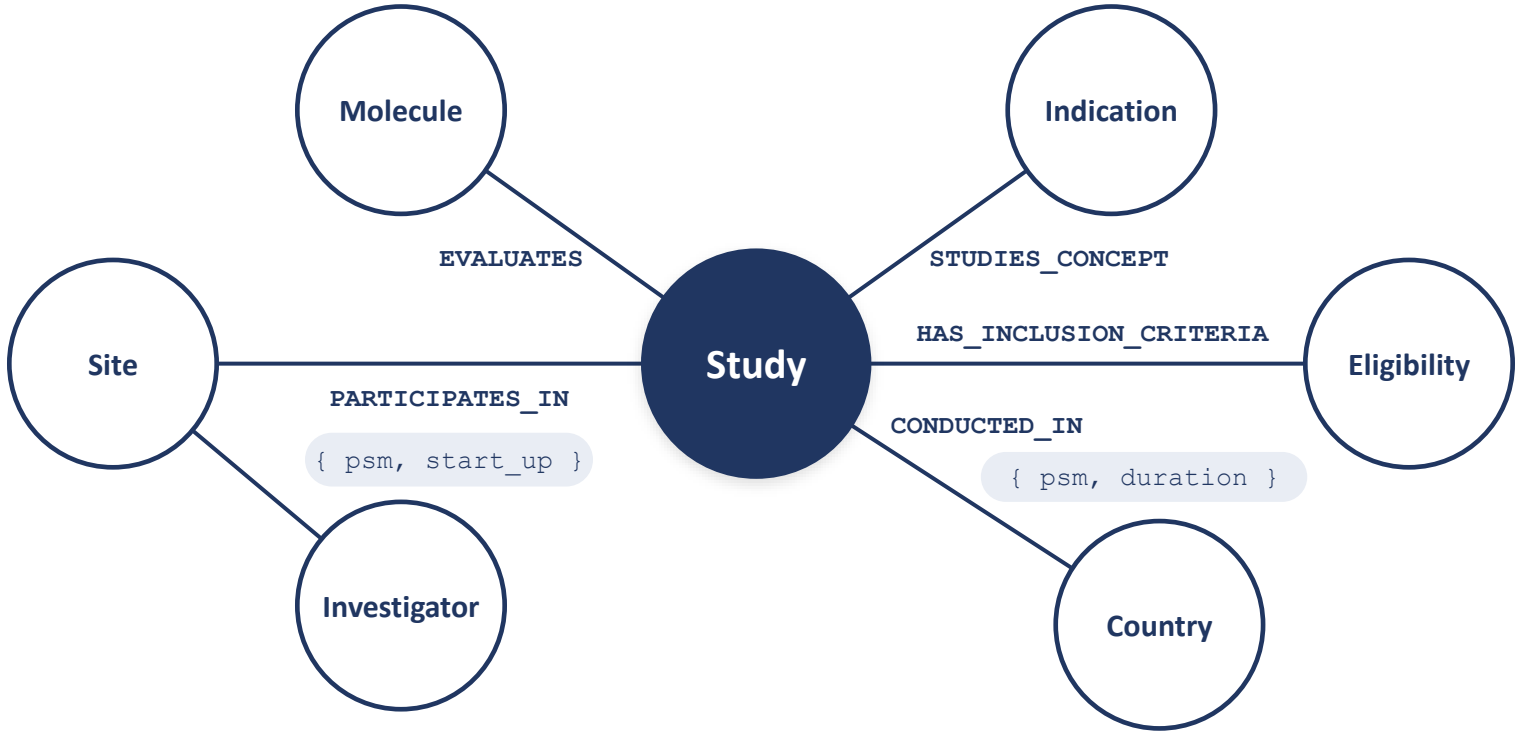
40+ relational source tables (PostgreSQL)

↓

2.5M nodes

3.5M edges

15–30 min for a full reload of the graph



DESIGN RULES THAT SAVED US PAIN

Metrics live on edges

Enrollment rate belongs to a site–study link

Separate terminologies

Linked by mapping edges, never merged

Pre-join at ingestion

Nested entities joined in ETL, not at query time

History stays in

Terminated studies and dropped sites are precedent

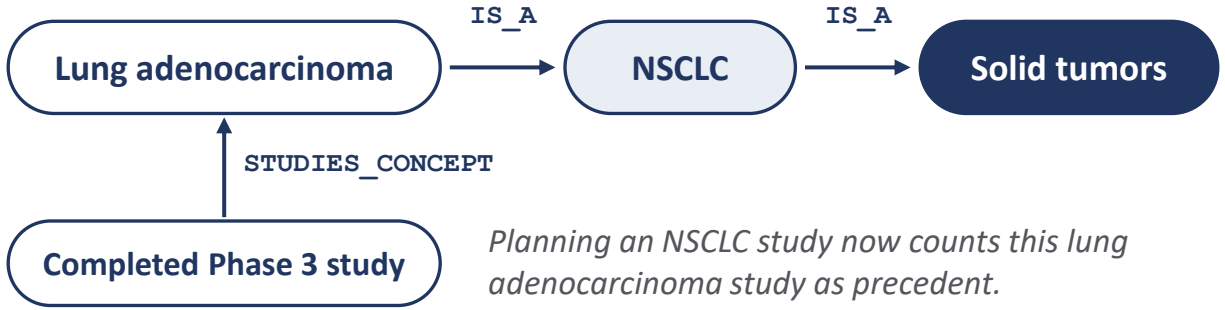
Ontologies give the graph clinical meaning

DRUG HIERARCHY



A new ACE-inhibitor program inherits precedent from the whole class.

DISEASE HIERARCHY



Planning an NSCLC study now counts this lung adenocarcinoma study as precedent.

SOURCES

MeSH 5,813 terms linked from our data

NCIt Oncology concepts

UMLS Cross-ontology mappings

NEW EDGE TYPES

`IS_A` `MAPPED_TO` `INHIBITS` `INDICATED_FOR`

No ontology is complete

Keep a text-match fallback



Hybrid beats purist: every lookup falls back to text match.

IN PRODUCTION

Embed the neighborhood, not just the node

Node + neighbors
A study with its arms, sites and concepts (1 hop)

→

Faceted template
Most informative fields first

→

MedEmbed-large
BGE-large base · ~335M params · 1024-d

→

Vectors in graph
Find analogous studies, then traverse

ILLUSTRATIVE `[Title] Phase 3 study in mCRPC [Phase] 3 [Indication] mCRPC [Eligibility] post-ADT [Neighbors] arms, sites`

HOURS TO EMBED ~6M GRAPH ENTITIES



500 → 150 tokens per template after field prioritization

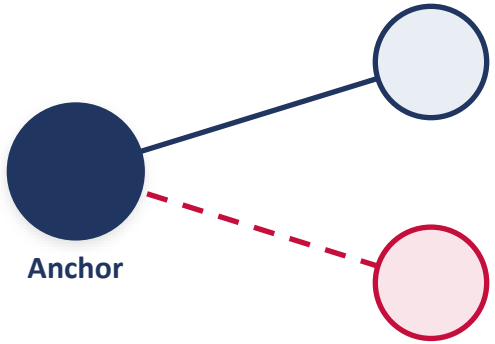
1 GPU one A10G on Databricks, S3 checkpoint per batch

25×+ faster: 50+ hours on CPU → ~2 hours on one GPU

IN PRODUCTION

Graph-guided fine-tuning tightens embeddings

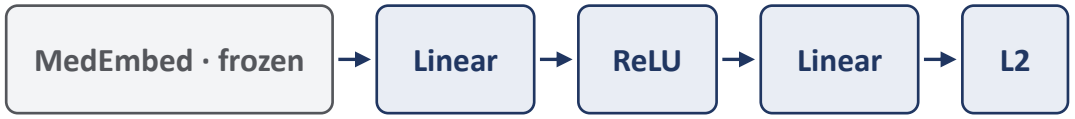
TRIPLETS MINED FROM THE GRAPH



Positive connected by IS_A or STUDIES_CONCEPT

Hard negative different label, but the base model thinks it's close

A SMALL PROJECTION HEAD ON A FROZEN MODEL



525K parameters · 30 minutes on CPU

Calinski–Harabasz higher is better	2,847 → 3,422	**+20%**
Davies–Bouldin lower is better	0.42 → 0.34	**–19%**
Inertia lower is tighter	12.5M → 10.1M	**–19%**

1M-node stratified sample, 10-fold CV. Cluster-quality metrics: a tighter, better-separated space, not a retrieval-accuracy benchmark.

SAME QUERY, BEFORE AND AFTER

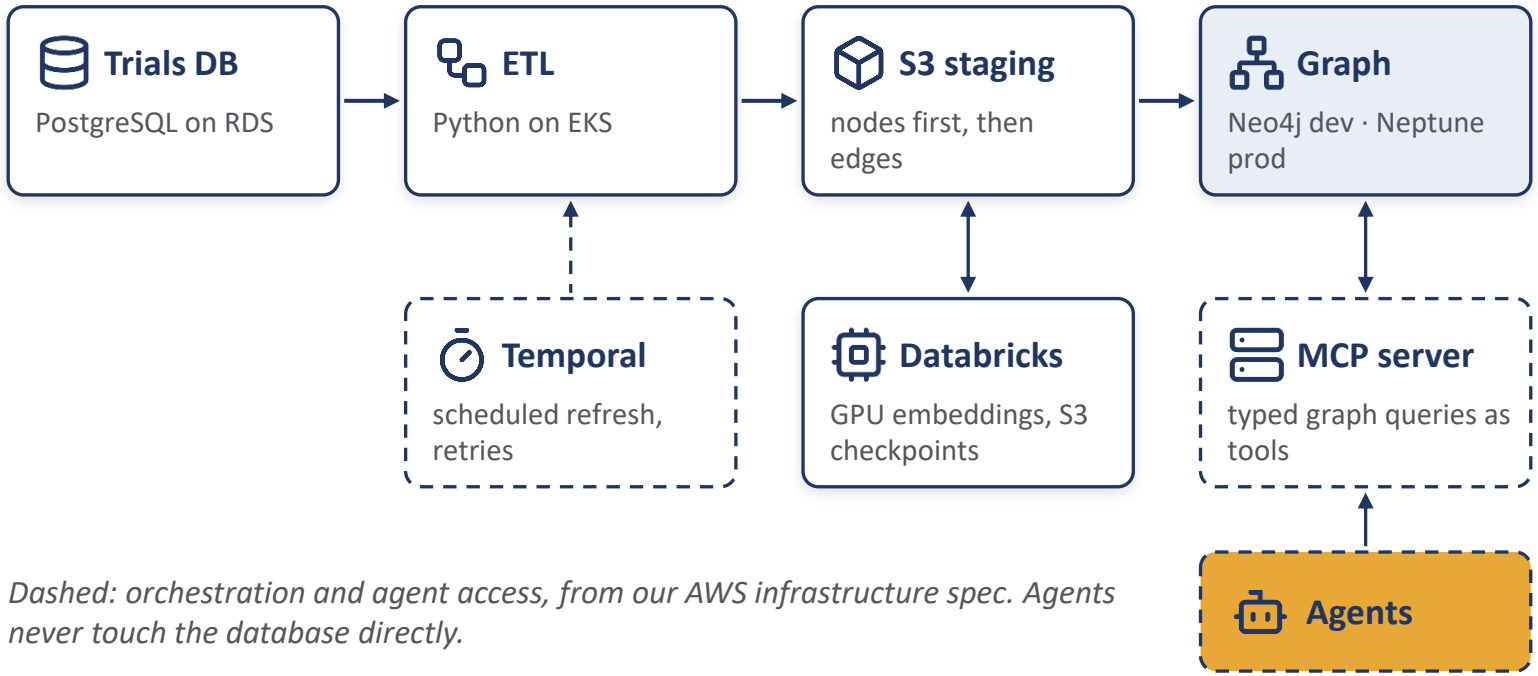
"metastatic lung adenocarcinoma"

Base model → melanoma trial ✗

Fine-tuned → NSCLC trials ✓

IN PRODUCTION

Architecture, speed and cost



Dashed: orchestration and agent access, from our AWS infrastructure spec. Agents never touch the database directly.

45 ms
p50 vector search, top-100
p99 120 ms

12 ms
p50 one-hop traversal
p99 35 ms

$330/mo
graph + embedding run cost
Neptune $200 · Databricks $100 · S3 $30

FIRST PRODUCTION USE CASE: TRIAL MATCHING

"65-year-old male, metastatic prostate cancer, failed ADT" → embed → vector search → graph traversal → **Phase 3 enzalutamide trial, mCRPC post-ADT**
top match · similarity score 0.94

Why Neo4j for development, Neptune for production

Neo4j · development

- Fastest local iteration: desktop and Docker
- Rich graph algorithms and analytics tooling
- Mature drivers and developer experience

Amazon Neptune · production

- Managed, in-VPC, IAM-authenticated
- Native S3 bulk loader, KMS encryption, CloudWatch
- openCypher and Gremlin on one engine

The rule that makes the split safe: graph queries stay in openCypher behind one abstraction layer; vector search sits behind its own interface, because that part differs between engines.

PART 1 · BUILD THE GRAPH

Graph schema at a glance

Operational layer

NODE LABELS

Molecule · Study · Country · Site
Investigator · Indication
EligibilityCriterion · Arm

RELATIONSHIPS

`PARTICIPATES_IN` · `CONDUCTED_IN`
`LOCATED_IN` · `EVALUATES`
`STUDIES_CONCEPT` · `HAS_ARM`

Ontology layer

NODE LABELS

MeSH · NCIt · UMLS concepts
Drug and disease hierarchies

RELATIONSHIPS

`IS_A` · `MAPPED_TO`
`INHIBITS` · `INDICATED_FOR`

Re-planning layer

NODE LABELS

Signal · Intervention
Recommendation · Decision
Outcome · Benchmark

RELATIONSHIPS

`CANDIDATE_FOR` · `GROUNDED_IN`
`DECIDED_BY` · `RESULTED_IN`
`SIMILAR_TO`

On every node and edge: `data_class` · `valid_from / valid_to` · `recorded_from / recorded_to`

DESIGN PRINCIPLE

Performance and cost detail

Measure	Value
Graph size	2.5M nodes · 3.5M edges
Embeddings	~6M graph entities · 1024-d · 24 GB (6 GB compressed)
Embedding run	~2 h on one A10G GPU (CPU: 50+ h), checkpointed to S3 per batch
Fine-tuning	525K-parameter projection head · 30 min on CPU
Vector search, top-100	45 ms p50 · 120 ms p99
One-hop traversal	12 ms p50 · 35 ms p99
Full graph reload	15–30 min
Refresh cadence	Weekly batch (change-data-capture streaming next)
Monthly cost	$330 for graph + embedding services: Neptune $200 · Databricks $100 · S3 $30

02

PART 2 OF 3

Plan the study

Molecule, study, country and site decisions from historical precedent

01 Build the graph — **02 Plan the study** — 03 Make it trustworthy

Four planning levels, one graph

Molecule

QUESTIONS

- Where has this molecule or its class been studied?
- In which indications and designs?

ON THE GRAPH

Class and mechanism edges (IS_A, INHIBITS), indications

Study

QUESTIONS

- Which completed studies are most like this one?
- How long did they take to enroll?

ON THE GRAPH

Enrollment duration, screen-fail rate, design, eligibility

Country

QUESTIONS

- Which countries enrolled fastest?
- How long did start-up take?

ON THE GRAPH

Country PSM and enrollment duration on CONDUCTED_IN

Site

QUESTIONS

- Which sites delivered on similar studies?
- How fast do they activate?

ON THE GRAPH

Site PSM, activation and start-up on PARTICIPATES_IN

```
// One traversal across all four levels
MATCH (m:Molecule)<-[:EVALUATES]-(s:Study)-[r:CONDUCTED_IN]->(c:Country),
      (site:Site)-[p:PARTICIPATES_IN]->(s), (site)-[:LOCATED_IN]->(c)
RETURN m, s, c, site, r.country_psm, p.median_start_up_months
```

Similar studies supply the benchmarks

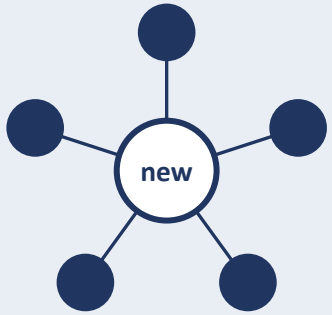
The new study

FOR EXAMPLE

- Phase 3, placebo-controlled
- NSCLC, EGFR-mutant
- ~160 sites, global
- Key eligibility from the synopsis

Agent extracts the attributes

k most similar studies



SIMILAR ON

indication + subtypes
design class
endpoint family
population
geography

Vector similarity + graph structure

Benchmarks with evidence

COMPUTED BY QUERY

- Enrollment duration
- PSM by country
- Screen-fail rate
- Start-up and activation time

Each with its studies, count and as-of date

Confidence scales with evidence

Five close analogs give a tight benchmark; two loose ones give a wide range, flagged as such.

Priors are labelled as priors

Benchmark-derived numbers are shown apart from the study's own data.

The graph computes; the agent explains

THE PLANNER'S ASK

"Up to 10 countries, US required. 160 sites, capped at AR 15 · JP 25 · DE 20 · US 40. Then suggest 3 more countries and place 30 more sites."

Agent turns the ask into constraints

TYPED QUERIES DO THE MATH

1. **Rank countries**
   Across similar studies, US kept in
2. **Allocate 160 sites**
   Under per-country caps, scaled to the total
3. **Suggest 3 more countries**
   Each backed by ≥ 2 comparable studies
4. **Place 30 more sites**
   Weighted by start-up time

```
// Step 1: rank countries
MATCH (s:Study)-[r:CONDUCTED_IN]->(c)
WHERE s.study_id IN $similar
WITH c, count(DISTINCT s) AS n,
  avg(r.country_psm) AS psm,
  avg(r.enrollment_duration) AS dur
WHERE n >= 2
RETURN c.country_id, psm, dur, n
ORDER BY psm DESC, dur ASC

// every row carries n, its evidence
// steps 2-4 run on the same graph
```

Agent frames the ask → **Queries compute every number** → **Study team decides**

Rerun the query, get the same answer. Every number traces back to its studies.

The agent's output the recommendation, why each country and site made the list, and the graph path behind every number

LESSONS

Six lessons from building it

01

Stream, don’t load

6M × 1024-d vectors ≈ 24 GB. Memory-mapped arrays, not RAM.

02

Measure retrieval, not just clusters

Cluster quality is a proxy. Hold retrieval to recall@k on known matches.

03

No ontology is complete

Keep a text-match fallback. Hybrid beats purist.

04

Develop on Neo4j, run on Neptune

Only if graph queries stay in openCypher behind one layer.

05

Reconcile before you retrieve

One definition of “enrolled” across CTMS, EDC and the warehouse.

06

Keep the model out of the math

Agents frame and explain; queries, optimizers and forecasts compute.

TAKEAWAYS

Takeaways and what's next

1

Study planning is a graph problem

Precedent, performance and geography are relationships, not rows.

2

Make retrieval clinical

Ontologies + graph-guided embeddings, for a few hundred dollars a month.

3

AI at the edges

Queries, optimizers and forecasts at the core; study teams decide.

4

Design governance into the data

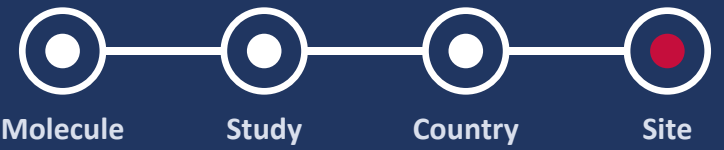
Blinding, as-of benchmarks, reconciled sources and provenance.

What's next

NEAR TERM	Change-data-capture streaming to replace the weekly batch refresh
NEAR TERM	Explainable recommendations: the graph path with every answer
IN DEVELOPMENT	Scenario simulation: what-if plans with P50 / P80 enrollment dates
DESIGN	In-flight early warning and next best action

Thank you

Questions and conversation welcome



Mukul Virmani

Director, Clinical Data Science & Trial Analytics · Gilead Sciences