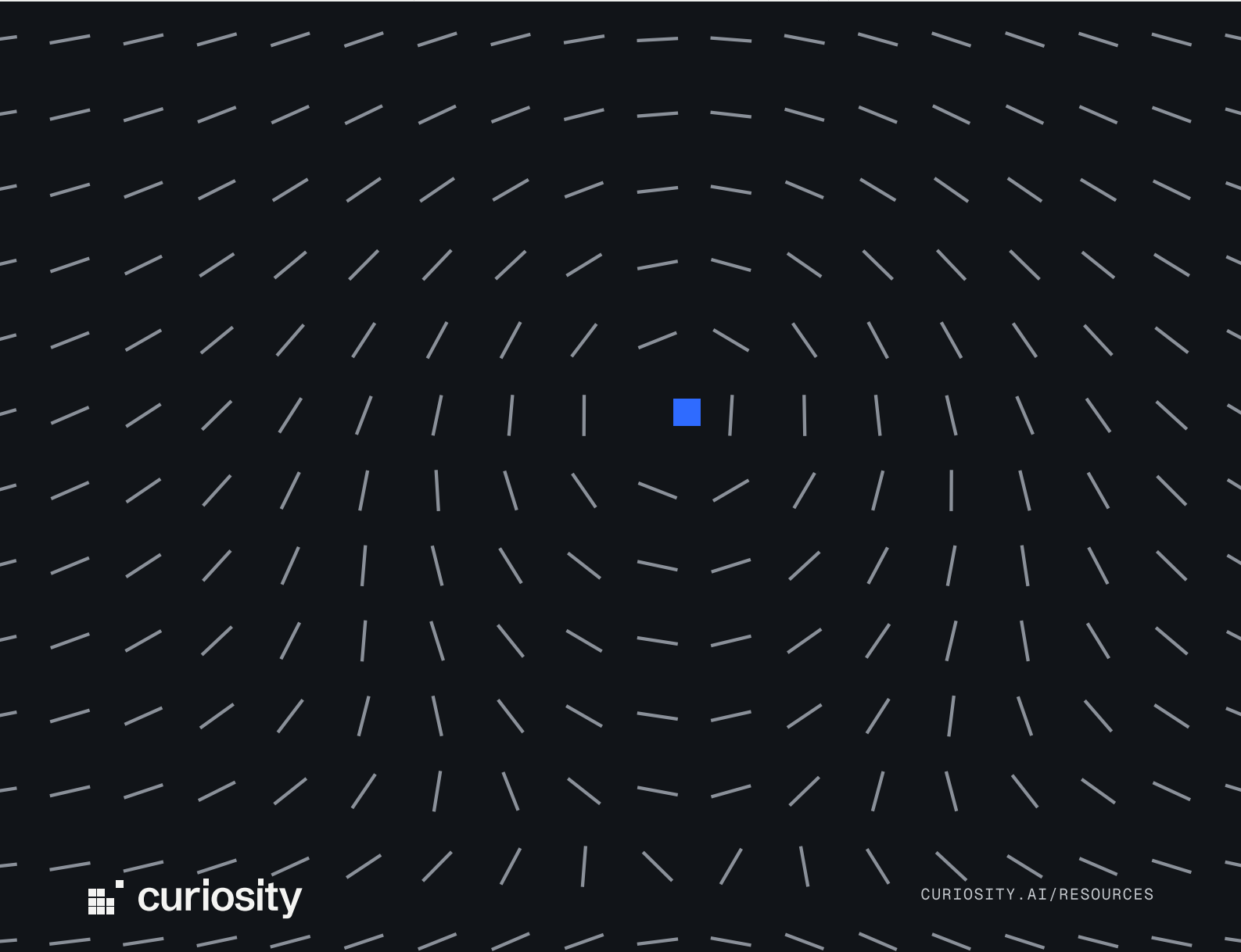


Modern enterprise search

Search is not one thing.

The main kinds of search used in enterprise systems, how each works, and where each belongs. Written for the architects who have to choose.

TOPIC	FOR	DATE
Search architecture	Architects and engineering leads	October 2026



Six layers, and where each one belongs

01	Search is not one thing Why most disagreements are about which layer is meant.	03
02	Where keyword still wins Exact retrieval for part numbers and identifiers.	06
03	What vectors add Meaning beyond the words a field report uses.	09
04	Hybrid and graph retrieval Ranking that knows how records relate.	11
05	Agents that search Search as a tool, under the user’s permissions.	13
06	How Curiosity implements it All six layers, in one system.	15
.	Glossary	20

IN ONE PARAGRAPH

Most arguments about search are about which layer is meant. A part number needs an exact match. A field report needs meaning. An agent needs both, and the permissions of the person asking. This paper takes the layers apart so you can choose them on purpose.

01

Search is not one thing

Why most arguments about search are about which layer is meant

The six layers, from exact match to agents

What each layer needs from the data underneath

1.1

Six layers, one question each

Name the layer before you argue about search. Most disagreements end there.

Ask five engineers what search means and you get five systems. One means the box that finds a part number. One means a ranked list. One means a chat window that answers in sentences.

The layers build on each other. An agent is only as good as the search it calls. A hybrid ranking is only as good as the two lists it merges.

Keyword search matches the words in the question to the words in a record. Fuzzy search allows for typos and variants. Vector search matches meaning, so a description finds a record written in other words.

So the first design decision is not which product to buy. It is which layers your questions need, and what each one needs from the data underneath.

Hybrid search ranks both kinds of match in one list. Graph retrieval follows the links between records: a part, the notice about it, the tickets it caused. Agents use all of these as tools.

LAYER

One way of deciding which records match a question, and in what order.

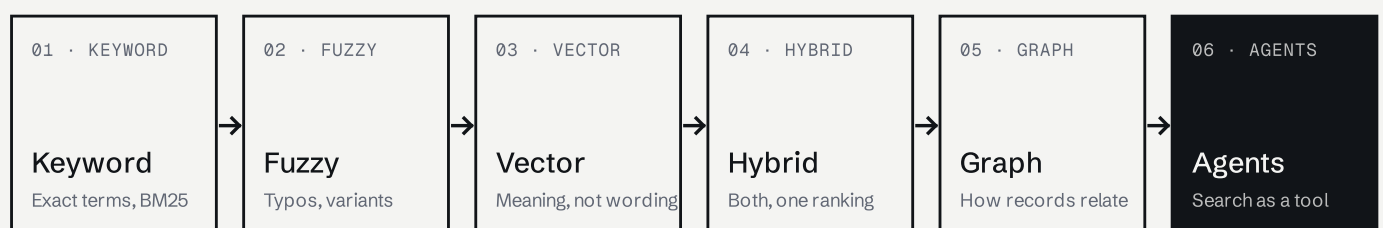


FIG. 1.1 · THE SIX LAYERS OF SEARCH, EACH BUILDING ON THE LAST.

TABLE 1.1

Which layer finds what

LAYER	FINDS	MISSES	EXPLAINS ITSELF	HANDLES WORDING
Keyword	Exact ids, part numbers	Synonyms, wording		
Fuzzy	Typos, variants	Meaning		
Vector	Meaning, descriptions	Exact ids		
Hybrid	Both, in one list	Relations between records		
Graph	How records relate	Free text alone		
Agents	Multi-step questions	Nothing, if built on the rest		

YES IN PART NO

READ IT THIS WAY

No row is all solid. Each layer covers a gap the one above it leaves, which is the case for running them together rather than choosing one.

1.2

What each layer needs from the data

Keyword and fuzzy search need clean text fields and the identifiers kept intact. Vector search needs the meaning in the text, so scanned PDFs and field reports have to be read first.

The graph needs the links between records: which part a notice is about, which ticket it caused. Agents need all of it, plus the permissions of the person asking.

BEFORE YOU CHOOSE

List the questions people ask today. Mark each with the layer it needs. The layers you mark most are the ones to build first.

02

Where keyword still wins

Why exact retrieval is still the right tool for part numbers

How BM25 scores a match, term by term

One query against two indexes

2.1

How BM25 scores a match

A technician types A320-2741-08. There is one right answer, and a near miss is a wrong part on the line. Keyword retrieval finds the exact string, fast, and can say why it matched.

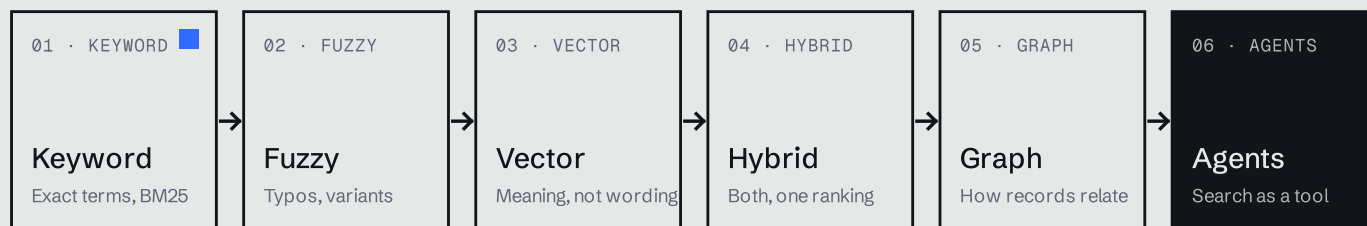


FIG. 2.1 · THE SIX LAYERS OF SEARCH. THIS SECTION IS ABOUT THE FIRST.

Each query term adds to the score. A rare term adds more than a common one. A term adds less each time it repeats, and a long document is damped so it cannot win by size alone.

Vector search is built to find what is close. For an identifier, close is the failure. That is why a production system keeps a keyword index beside the vector one.

Keyword retrieval also explains itself. A match is a set of terms that appear in the record, so an engineer can see why a result ranked where it did.

Its cost is vocabulary. A field report that says hydraulic leak will not match a ticket that says fluid loss. Section 03 picks up there.

KEY POINT

Keep keyword for identifiers. Add vectors for meaning. Never replace one with the other.

BM25

Ranks by how often a term appears, damped by document length and by how common the term is.

2.2

One query, two indexes, one ranking

The same request runs against both indexes. The exact match is boosted, the close matches follow, and the user's permissions are applied before anything is ranked.

```
// one query, two indexes, one ranking
var q = Query.Parse("A320-2741-08 fluid loss");
var hits = graph.Search(q)
    .Keyword(boost: 2.0) // exact ids
    .Vector(k: 50)       // meaning
    .AsUser(ctx.User);   // permissions
```

LISTING 2.1 · ILLUSTRATIVE, NOT THE API

QUERY	KEYWORD	VECTOR	RIGHT TOOL
A320-2741-08	Exact	Near misses	Keyword
fluid loss, aft	Misses synonyms	Finds leaks	Vector
leak on 2741-08	Partial	Partial	Hybrid

2.3

When the exact match is missing

A query for a part number that is not in the index should return nothing, and say so. A list of near misses invites the wrong part.

A good system shows the empty result and offers the close matches as a separate list, marked as close, so the engineer chooses with open eyes.

DESIGN RULE

Exact results and close results are two lists until a person or a ranking rule decides they are one.

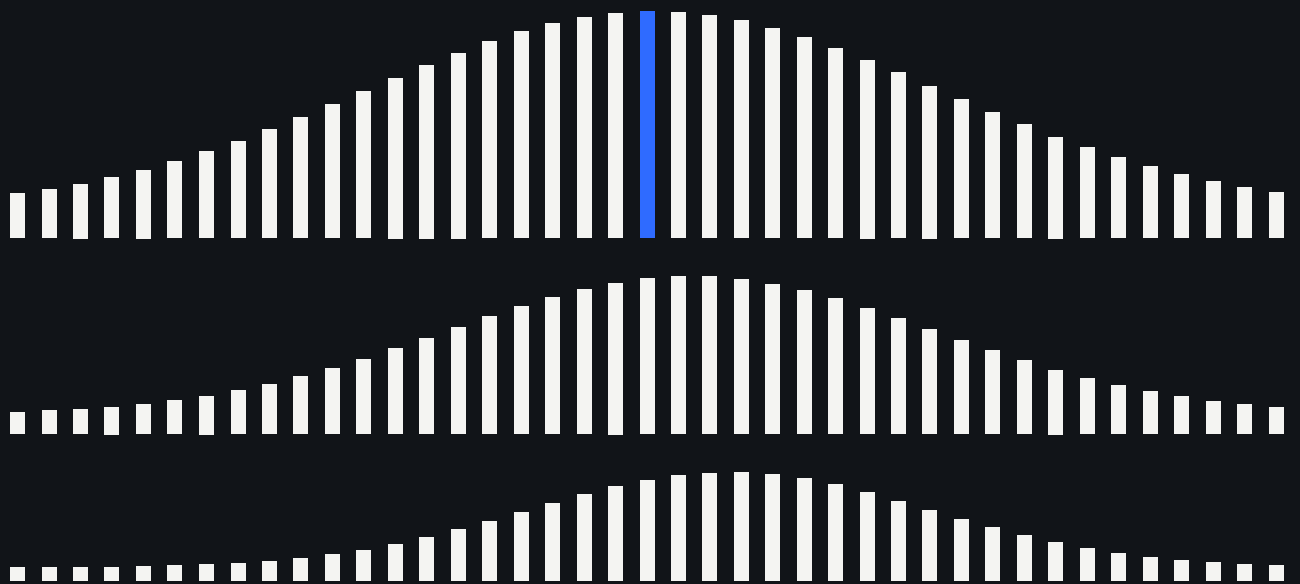
03

What vectors add

How a record becomes a point in a space of meaning

Why close is right for field reports and wrong for part numbers

What it costs to run, and to explain



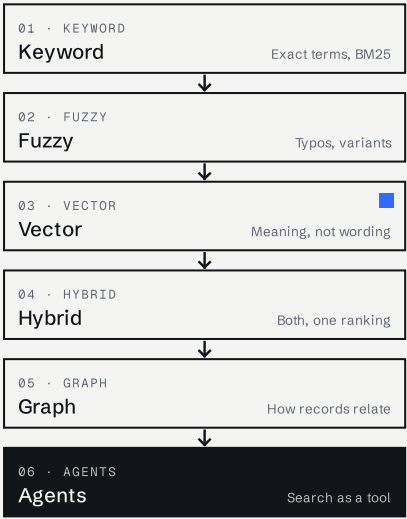
3.1

Meaning instead of words

A field report says the aft door seal was weeping fluid. The ticket about the same fault says hydraulic leak, door 4. No word is shared, so keyword search treats them as unrelated.

Vector search turns each record into a point in a space where distance stands for meaning. Records about the same fault land close together, whatever words their authors chose.

It also costs more to explain. A keyword match lists the terms it found. A vector match is a distance, and a reviewer has to trust it or open the record.



That is right for descriptions and wrong for identifiers. Part 2741-08 and part 2741-09 sit close in meaning, and they are different parts.

Use it where people describe, beside the keyword index, and let a hybrid ranking set the order. Section 04 shows how.

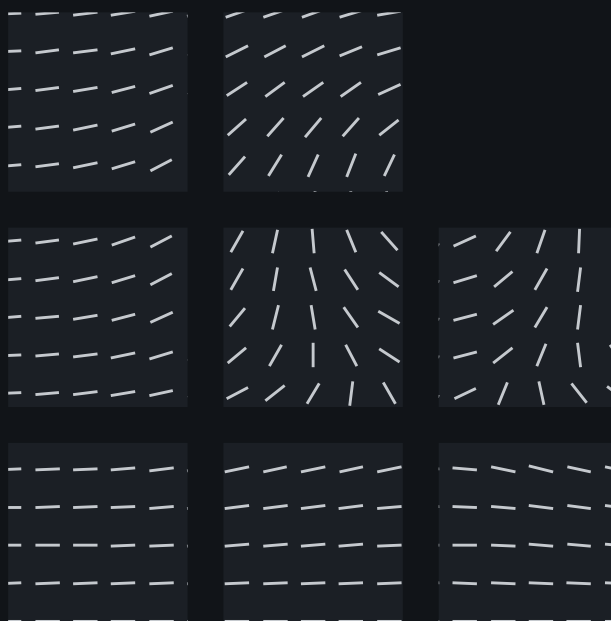
FIG. 3.1 · THE SIX LAYERS. THIS SECTION IS ABOUT THE THIRD.

HNSW

An index that finds the nearest vectors quickly, without comparing the question against every record.

KEY POINT

Use vectors where people describe a thing. Keep keyword where they name it.



04

Hybrid and graph retrieval

Exact and close matches, ranked as one list

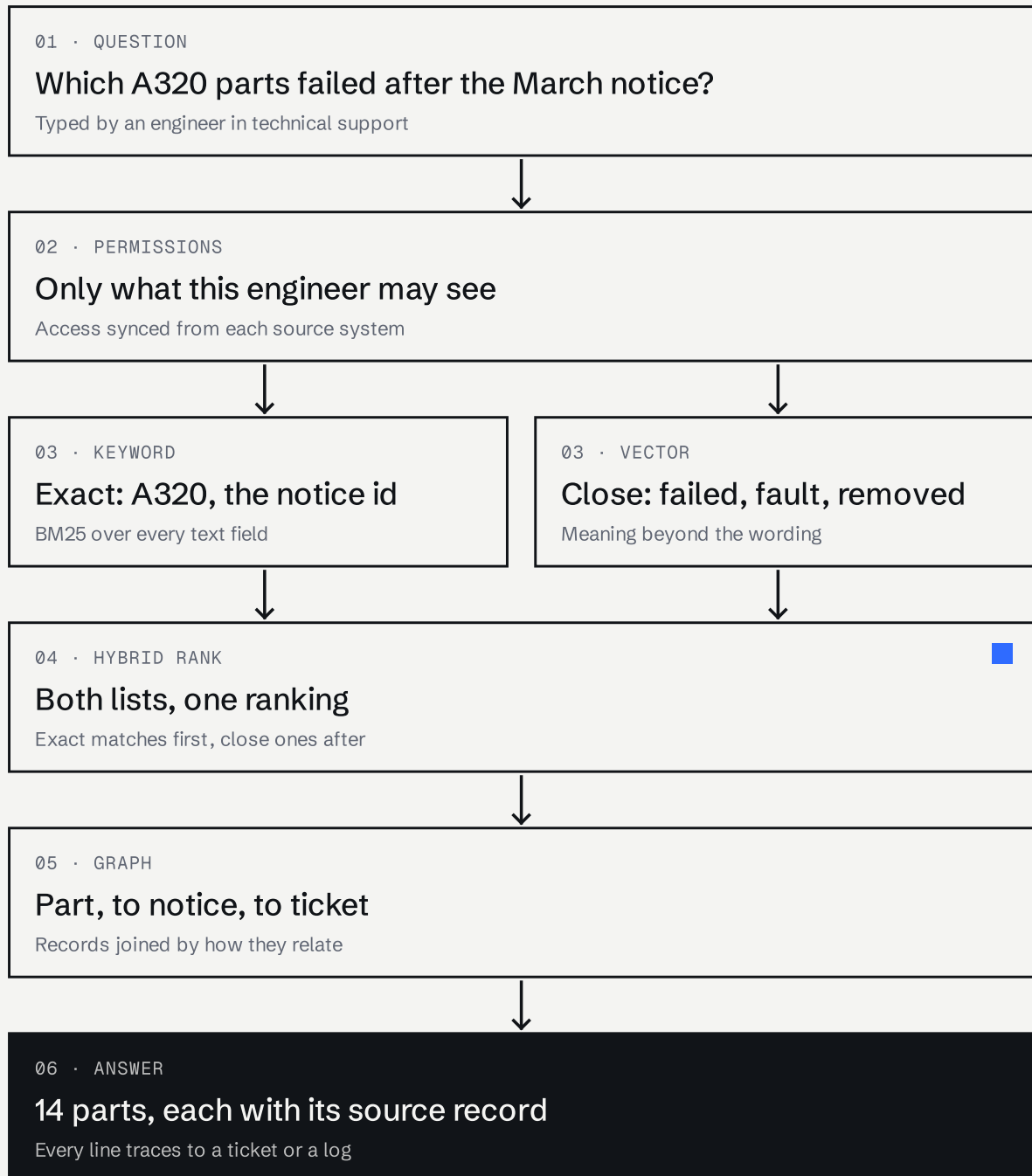
Following a part to its notice and its tickets

The query path, step by step

FIGURE 4.1

The query path

One question, from the moment it is typed to the answer with its sources. Keyword and vector run side by side.



02

Permissions apply before search, so a result the engineer may not open is never ranked.

04

The marked step: exact and close matches ranked as one list, exact first.

06

The answer cites records, so a reviewer can open each source.

05

Agents that search

Search as a tool an agent calls, several times

Why permissions have to hold at every step

What a reviewer needs to see afterward

5.1

An agent calls search, again and again

Ask an agent which suppliers are behind the delays on the A320 line. It does not search once. It finds the delayed orders, then the parts on them, then the supplier notices about those parts.

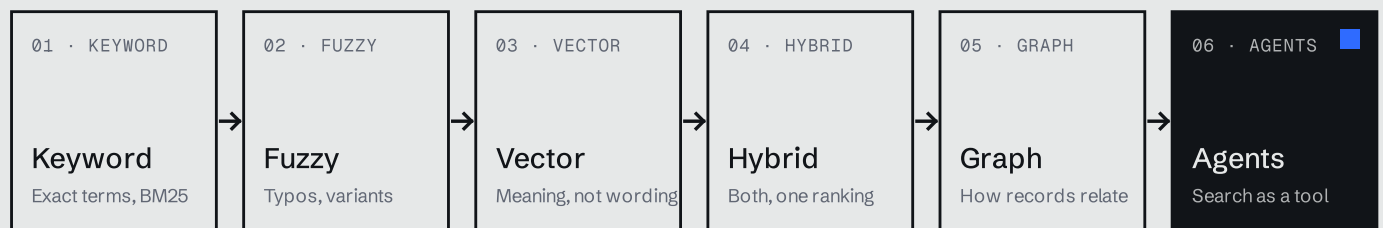


FIG. 5.1 · THE SIX LAYERS. THIS SECTION IS ABOUT THE LAST.

Each of those steps is a search, and each one runs as the person who asked. If one step ignores permissions, the answer can hold a record the user could never open.

That is why permissions belong in the search layer, not in the agent. An agent that filters afterward has already read what it should not have seen.

The agent also needs exact matches. A supplier notice is found by its id, and a close match on an id is the wrong notice.

Finally, every step leaves a trail. A reviewer should be able to open each search the agent ran, with its results, and see how the answer was built.

KEY POINT

An agent is only as careful as the search it calls. Permissions hold at every step, or not at all.

TOOL CALL

A request the agent makes to another system, here a search, with the user's identity attached.

06

How Curiosity implements it

All six layers in one system, on your infrastructure

Where it runs today, and for whom

How to start, in three steps

AIRBUS

“By intelligently enhancing our search efficiency, Curiosity lets Airbus technical support quickly find information across millions of documents.”

SERVICES INNOVATION TEAM · AIRLINE SERVICES AT AIRBUS

THE PROBLEM

Forty years of technical data across CRM systems, network drives and engineering databases, in a vocabulary of acronyms and reference numbers that defeated ordinary search.

WHAT CHANGED

One entry point over every source, the engineering references extracted and joined in a graph, filters by aircraft type, program and reference number.

Running where the data is.

Across Curiosity's production deployments, on premises and in private clouds.

30TB+

DATA CONNECTED IN PRODUCTION

20,000+

ACTIVE USERS ACROSS DEPLOYMENTS

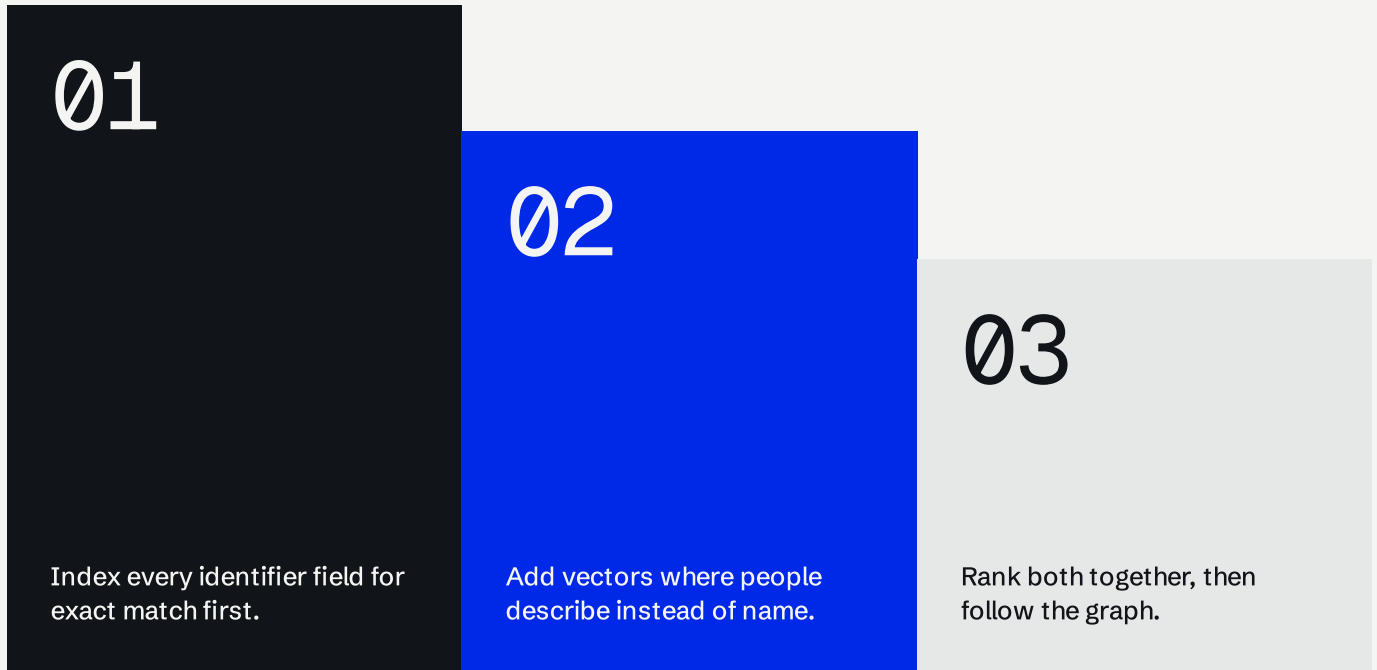
15%+

EFFICIENCY GAIN IN PRODUCTION WORKFLOWS

70+

ENTERPRISE SYSTEMS CONNECTED

Three steps, in this order



BEFORE YOU CHOOSE

- ☐ Which fields hold identifiers: part numbers, notice ids, serials?
- ☐ Where do people write in their own words: field reports, tickets?
- ☐ Who may see which record, and in which source system is that set?
- ☐ Who needs to check why a result was returned?

What to take from this paper

01

Name the layer first.

Most arguments about search are about which layer is meant.

02

Exact and close, ranked together.

Keyword for identifiers, vectors for descriptions, one list.

03

Permissions in the search, not after it.

Every step an agent takes runs as the person who asked.

Terms used in this paper

BM25

A keyword ranking: frequent terms in short records score higher, common terms count less.

HNSW

An index that finds the nearest vectors quickly without comparing against every one.

KNOWLEDGE GRAPH

Records and the typed links between them: a part, the notice about it, the ticket it caused.

VECTOR SEARCH

Search by meaning: records close in meaning to the question, whatever their words.

FUZZY MATCH

A match that allows small differences in spelling, such as a typo in a part name.

HYBRID RANKING

One list ranked from keyword and vector results together.

PERMISSION-AWARE

Results are filtered by what the user may see in the source system, at query time.

AGENT

A program that uses search as a tool to answer a question in several steps.

Describe it Monday. Ship it this week.

Curiosity connects tickets, part records, maintenance logs and field reports into one permission-aware graph, and serves it to the model you choose, on premises or in your private cloud.

curiosity.ai/request-demo

DATA RESIDENCY

EU, on premises or private cloud

COMPLIANCE

GDPR

MEMBER

KI Bundesverband