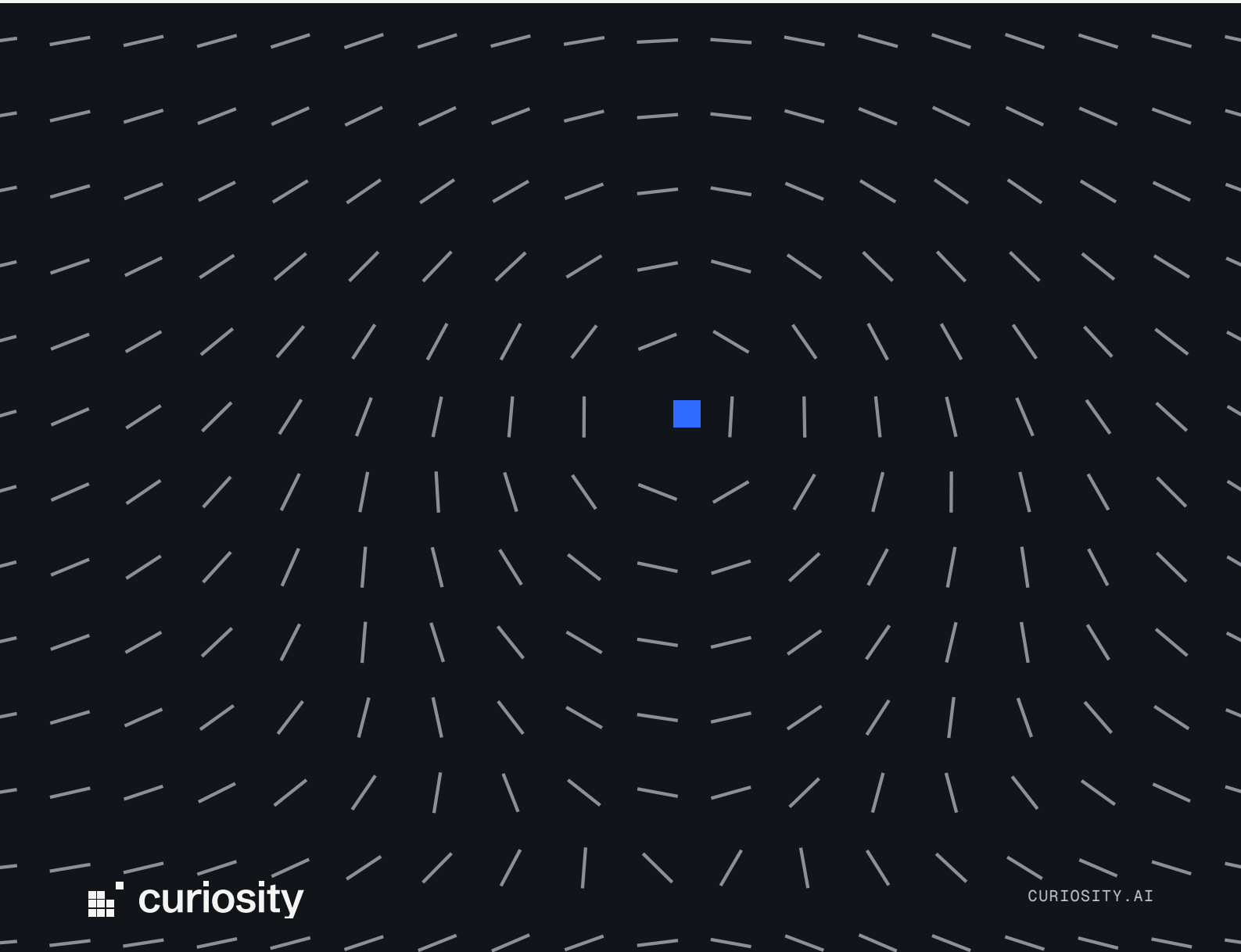


How Curiosity processes your data.

What happens to data in a Curiosity system, from the connector that loads it to the interface your users see. Each step links to its documentation page.

TOPIC	FOR	DATE
Data processing in Curiosity Studio	Business and IT professionals	October 2026



From connector to custom interface

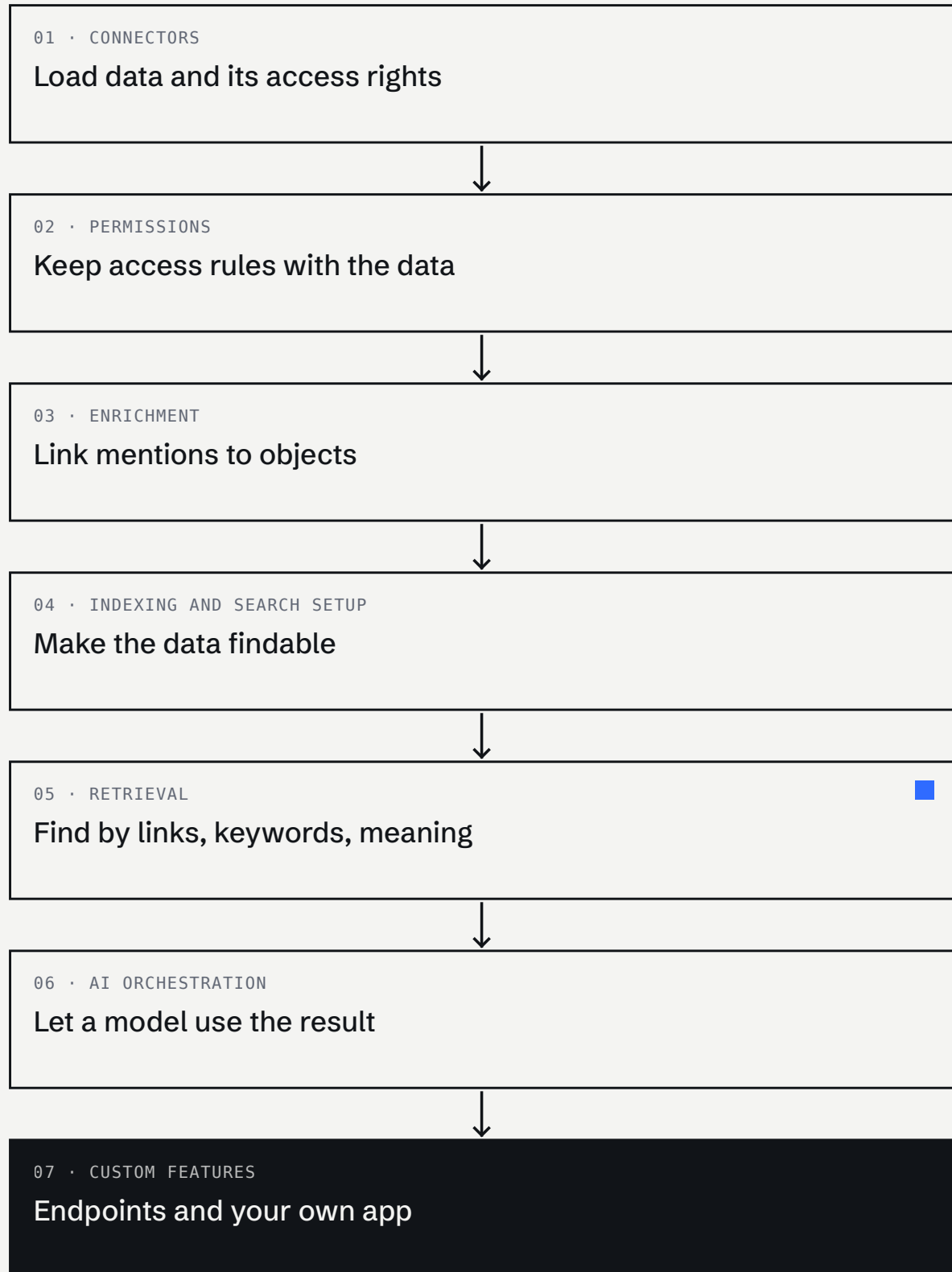
	Overview	03
	Seven steps, from source to custom app.	
01	Connectors	04
	Records and files load with their access rights.	
02	Permissions	05
	Access rules travel with the data.	
03	Enrichment	06
	Text is read, and mentions are linked to objects.	
04	Indexing and search setup	07
	Keyword and meaning-based indexes, facets and filters.	
05	Retrieval	08
	Following links, keyword search and meaning-based search.	
06	AI orchestration	09
	Prompts, skills, tools and agents on your model.	
07	Custom features and interfaces	10
	Your own logic and apps, drafted with Sudo.	

IN ONE PARAGRAPH

Curiosity connects your data once and loads it into one graph together with its permissions. It then enriches the text, indexes it for keywords and meaning, and serves the result to search, to AI agents and to your own applications. Each of them reads the same permission-aware data.

The seven steps

Steps 01 to 03 bring data in and prepare it.
Steps 04 and 05 make it findable. Steps 06
and 07 put it to work.



1.1

Connectors: load data and its access rights

A connector copies what a source system holds, and who may see it, into Curiosity. It runs on a schedule, and each item keeps the ID it has in the source, so a rerun updates items instead of copying them.

Monday	first run	copies every ticket and document
Tuesday	later run	adds the tickets that changed since Monday
Wednesday	later run	adds the tickets that changed since Tuesday

LISTING 1.1 · ONE CONNECTOR OVER A WEEK. ILLUSTRATIVE.

ITEM	SOURCE	WHAT CURIOSITY STORES
Ticket T-1042	Ticket tool	One item, linked to its customer
Contract PDF	File share	One item, linked to the ticket
Contract folder	File share	Visible to the legal team only

TABLE 1.1 · WHAT A CONNECTOR BRINGS IN. THE LAST ROW IS THE ACCESS RIGHT, MAPPED FROM THE SOURCE.

GOOD TO KNOW

Data deleted in the source is not deleted in Curiosity automatically, so the connector has to remove it. Section 03 covers how the text of documents is read.

[Connectors](#) · [Schema design](#) · [Ingestion pipelines](#)

2.1

Permissions: ingested with the data, applied on every read

Connectors attach access rules while they write, and users arrive with the teams your identity provider gives them. Each part of the system then checks the same rules.

WHERE	WHAT HAPPENS
Connector	Marks a record private, or limits it to a team or person, as the source does
Sign in	Your identity provider's groups decide a user's teams, refreshed at each sign in
Graph	A record is visible only if the user has a route to it
Search	Every query is filtered by the user's teams
Endpoints and AI	Retrieval runs as the signed-in user. The model sees only that result

TABLE 2.1 · WHERE A PERMISSION IS WRITTEN AND WHERE IT IS ENFORCED.

AT WRITE TIME	REVOKING ACCESS
Each indexed record stores who may see it, so checks stay fast.	Team changes apply at the user's next sign in. To cut access at once, remove the user in the admin settings.

READ IT THIS WAY

A record with no restriction is visible to everyone, so mirror the source's rules rather than relying on the default. The language model never decides who may read a record. It only receives what the user may already see.

[Permission model](#) · [Access control model](#) · [SSO group mapping](#)

3.1

Enrichment: turn text into linked objects

Enrichment finds the things a text mentions, such as part numbers and process references, and links each mention to the matching object in the graph. A document then connects to the parts and processes it names.

First, built-in readers extract text from files. Scans and photos go through text recognition (OCR). Then a language-processing step runs on each text field you choose. It finds mentions and links each one to an object, so a mention becomes something you can browse and filter on. Start with the simple, predictable methods and add the others only where gaps remain.

APPROACH	BEST FOR	SPEED	COST
Word list	Fixed lists: products, customers	Instant	Negligible
Pattern	Codes with a fixed format: part numbers	Instant	Negligible
Trained model	People, organizations, places in prose	Fast	Computing power
Language model (LLM)	Open-ended text, summaries	Seconds	Per use

TABLE 3.1 · THE FOUR WAYS TO CAPTURE MENTIONS, CHEAPEST FIRST.

EXAMPLE

A support ticket mentions part 2741-08. Enrichment links that mention to the part itself, so the ticket shows up whenever someone looks at the part. A new part number found by a pattern can create its own object automatically.

[NLP overview](#) · [Entity linking](#) · [Types of models](#) · [OCR](#)

4.1

Indexing and search setup: choose what is findable

You decide which fields are findable and how. Curiosity offers three ways to search them, and facets let users narrow the results.

	KEYWORD INDEX	VECTOR INDEX	HYBRID
Matches	Exact words	Meaning	Both, in one list
Best for	Identifiers, titles, short terms	Long descriptions	Mixed questions
Example	Searching 2741-08 finds records with that part number	Hydraulic leak also finds a report that says fluid loss	2741-08 fluid loss ranks exact matches first, similar reports after
Set up	Per record type and field in the admin settings	On long text fields. Existing data is processed in the background	Switch on both. Section 05 shows the merge

TABLE 4.1 · THE THREE WAYS TO SEARCH A FIELD.

FACETS AND FILTERS

Facets let users narrow results by status, priority, a linked customer or a time range. You set them in the admin settings. Documents and questions must be processed the same way for the vector index to work.

[Full text search](#) · [Embeddings](#) · [Filters and facets](#)

5.1

Retrieval: three ways to find, one ranked result

At query time Curiosity can follow links in the graph, search by keyword and search by meaning. The last two are combined into one list, and every route checks the user's permissions.



Following links

Moves along relationships: from a part to the notices that mention it, to the tickets that cite them. It answers questions about how records relate.

EXAMPLE

From part 2741-08, find the service notices that mention it and the tickets that cite those notices.



Keyword search

Matches exact terms. Right for identifiers, and you can see why a record matched. It misses a ticket that says fluid loss when the report says leak.

EXAMPLE

Searching for a ticket number returns that ticket and nothing else.



Meaning-based search

Finds the same idea in other words. Hybrid search combines it with keyword search, and the balance is adjustable: more keyword for identifiers, more meaning for narrative text.

EXAMPLE

A question about a weeping door seal finds reports about hydraulic leaks.

[Search model](#) · [Hybrid search](#) · [Ranking tuning](#)

6.1

AI orchestration: how a question reaches the model

Curiosity runs a loop between a language model you choose and the tools you define. The model asks for information, and permission-checked results come back with their sources.

A person asks a question. Curiosity sends it, with your instructions and a list of tools, to the language model you chose: OpenAI, Anthropic, Azure or a local model. The model asks for a search or an action. Curiosity runs it as the signed-in user and returns only what that user may see. The model writes a reply that cites its sources, and each turn is logged for audit.

For example, a support engineer asks which tickets mention a weeping door seal. The model asks for a search, Curiosity runs it as that engineer, and the reply lists the matching tickets with a link to each.

PROMPTS, SKILLS, TOOLS

A prompt sets the task. A skill is a written guide that teaches the assistant how your business works. A tool is a function that lets it search or act. For example, a tool searches your tickets, and a skill explains how your team approves a refund.

AGENTS

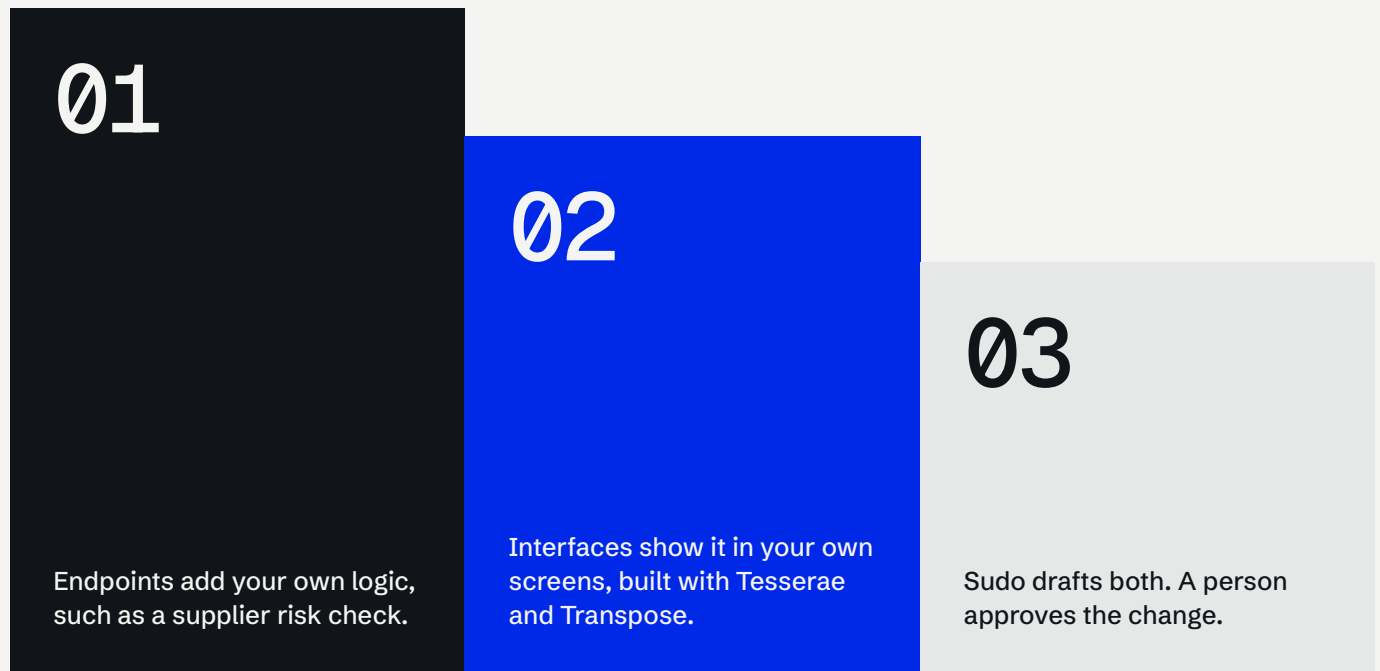
An agent saves instructions, a set of tools and a model as one reusable assistant. Apps and workflows can run it. For example, a supplier risk agent combines a ticket search, a notice search and your instructions.

DETAIL

[AI architecture](#) · [AI agents](#) · [AI tools](#) · [Skills](#) · [LLM configuration](#)

Custom features and interfaces: build on the same data

Endpoints add your own features, and interfaces put them in front of your users.



WHICH TOOL FOR WHICH JOB

- ☐ Load data from a system? Use a connector.
- ☐ Run something on a schedule? Use a scheduled task.
- ☐ Let chat trigger your logic? Use an AI tool.
- ☐ Add permission-aware logic or an action? Use an endpoint.

[Custom endpoints](#) · [Custom front-end](#) · [Tesseract](#) · [Transpose](#)

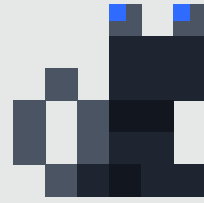
7.2

Sudo: the assistant that builds and changes your setup

Sudo is the admin assistant in Curiosity Studio. You describe what you need in a chat, and Sudo builds it.

Sudo works on your configuration: endpoints, AI tools, agents, skills, scheduled tasks, indexes, the data schema, enrichment settings and search settings. Ask for "a supplier risk check for the A320 line". Sudo reads your existing setup, drafts the endpoint, the AI tool and the settings it needs, and builds them to check that they work.

Then Sudo shows you the complete change on a review screen. You read it and approve it or discard it, and an approved change goes live as one tracked update.



KEY POINT

Describe what you need. Sudo builds it, and you review the change.

DETAIL

[Sudo](#) · [Sandbox](#) · [Approvals](#) · [Security](#)

What to take from this paper

01

Data is connected once, with its permissions.

Connectors load records, files and access rules into one graph.

02

Enrichment and indexes make it findable.

Mentions link to objects in the graph. Keyword and meaning-based indexes find exact terms and ideas.

03

Everything built on top reads the same result.

Search, AI agents, endpoints and custom interfaces all run as the user, on the same permission-aware data.

Describe it Monday. Ship it this week.

Pick the step that matters most to you and open its documentation page. Then ask us to walk through it on your own systems.

curiosity.ai/request-demo

DATA RESIDENCY

EU, on premises or private cloud

COMPLIANCE

GDPR

MEMBER

KI Bundesverband