Robo Analyzer

The links between code and data, as a graph you can read

Legacy system analysis tool · uEngine

If I change this function, how far does the impact reach?

Who reads this table right now, and who writes to it?

When the documents and the code disagree, which do I trust?

Whoever knows the answers has usually left the company, or isn't sure even if still here.

You can't tell just by opening the code

Until now

- Guess names and search again and again
- Investigate SQL and code separately
- Rely on people's memory and old documents

Robo Analyzer

- Calls and reads/writes in a single graph
- Jump from a table straight to functions and code lines
- The evidence stays right there on screen

Instead of starting over every time you need to check, analyze once and keep exploring.

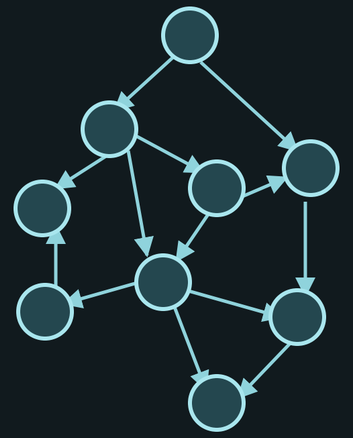
What goes in, what comes out

Source code

Schema DDL

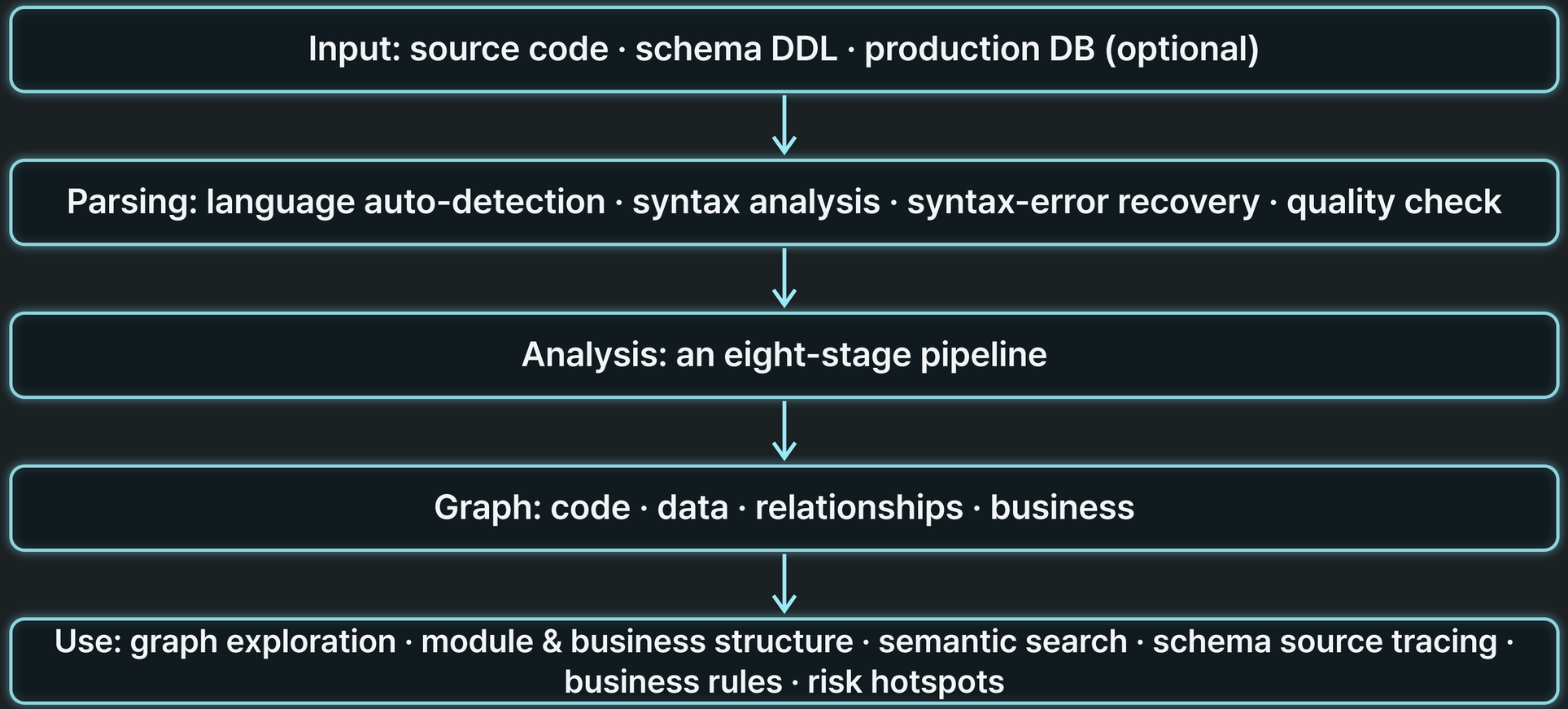
Production DB (optional)

Parse the code, find table access, and link it as relationships



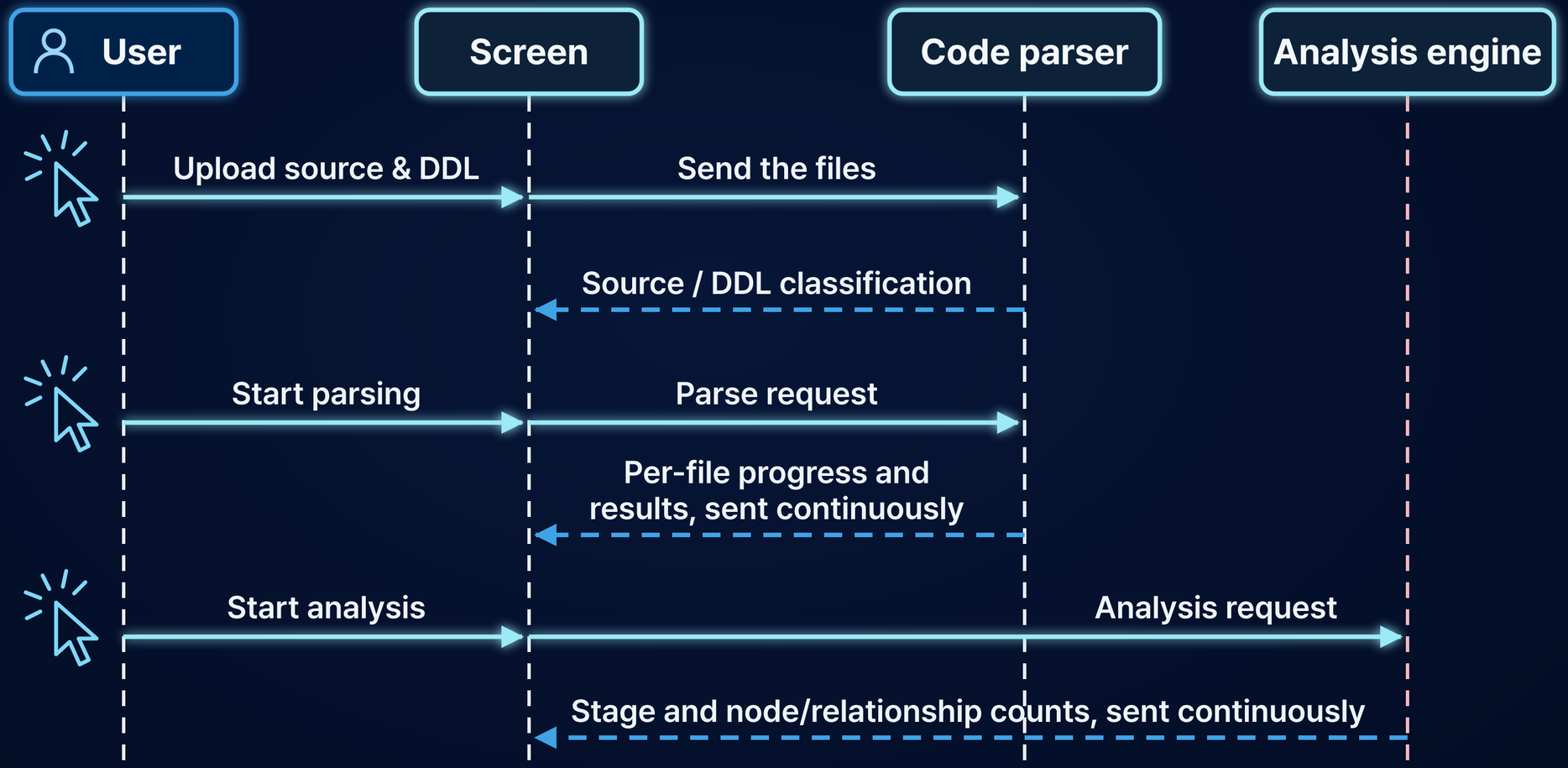
One graph linking calls and reads/writes, traceable down to the source code

The whole flow at a glance



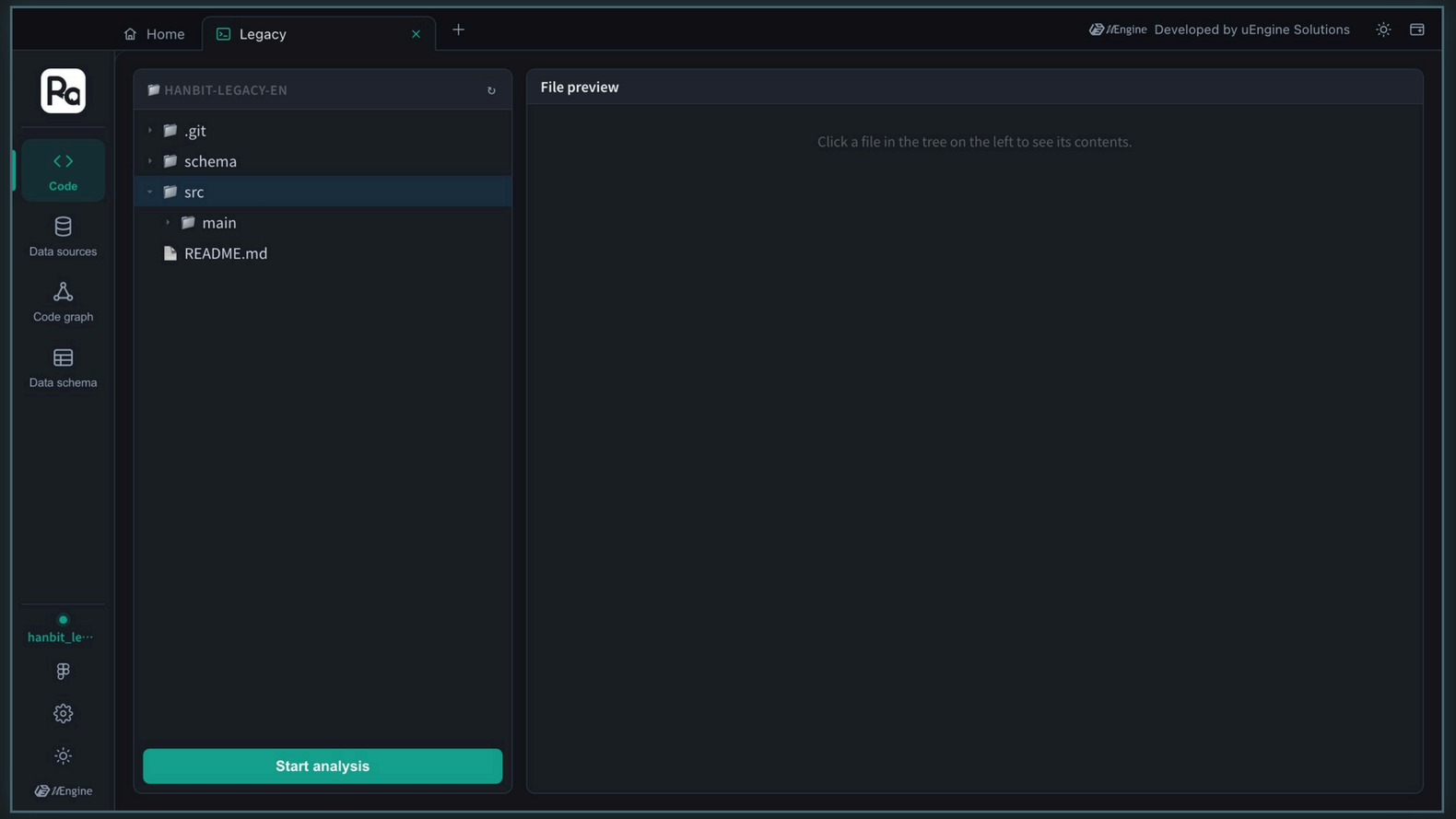
Legacy code and databases become one graph and are put back to use.

The user clicks three times



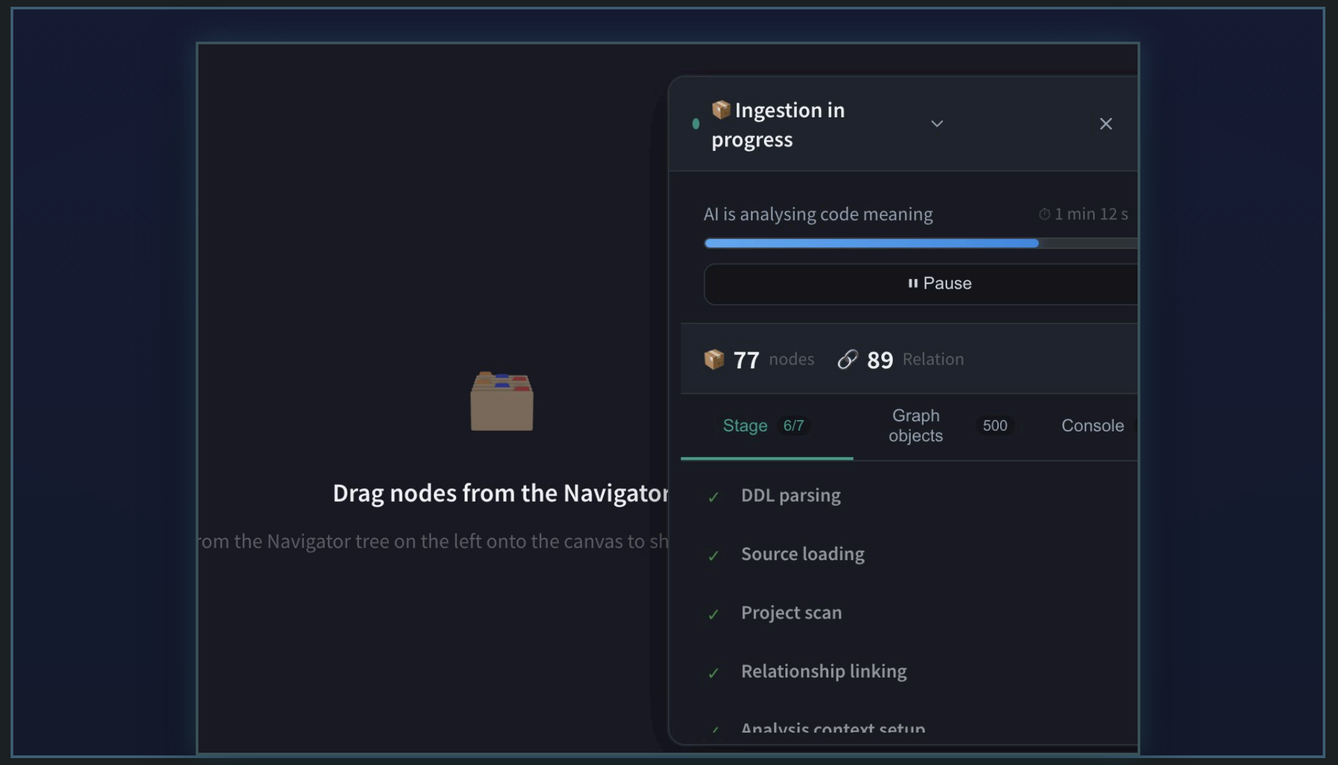
You click Upload, Parse and Analyze one by one. They don't chain automatically.

Put files in any way you like; it takes care of the rest



Upload source and DDL together and they are sorted automatically by file content.
Languages are detected automatically: Java, C, Python, Oracle PL/SQL, PostgreSQL.

You can watch the progress in real time



The eight stages run in order, and the node and relationship counts and the console keep updating before the run finishes. You can pause or stop the analysis while it runs.

Ever been handed code that won't even compile?

Common situation	Robo Analyzer
• Files with broken syntax • The build environment is gone • One parsing failure stops everything	• Healthy files are used as they are • Only the failed parts get a recovery attempt based on grammar rules • If that fails, only that part is left aside and the run continues

With the default settings, original source files are never modified.

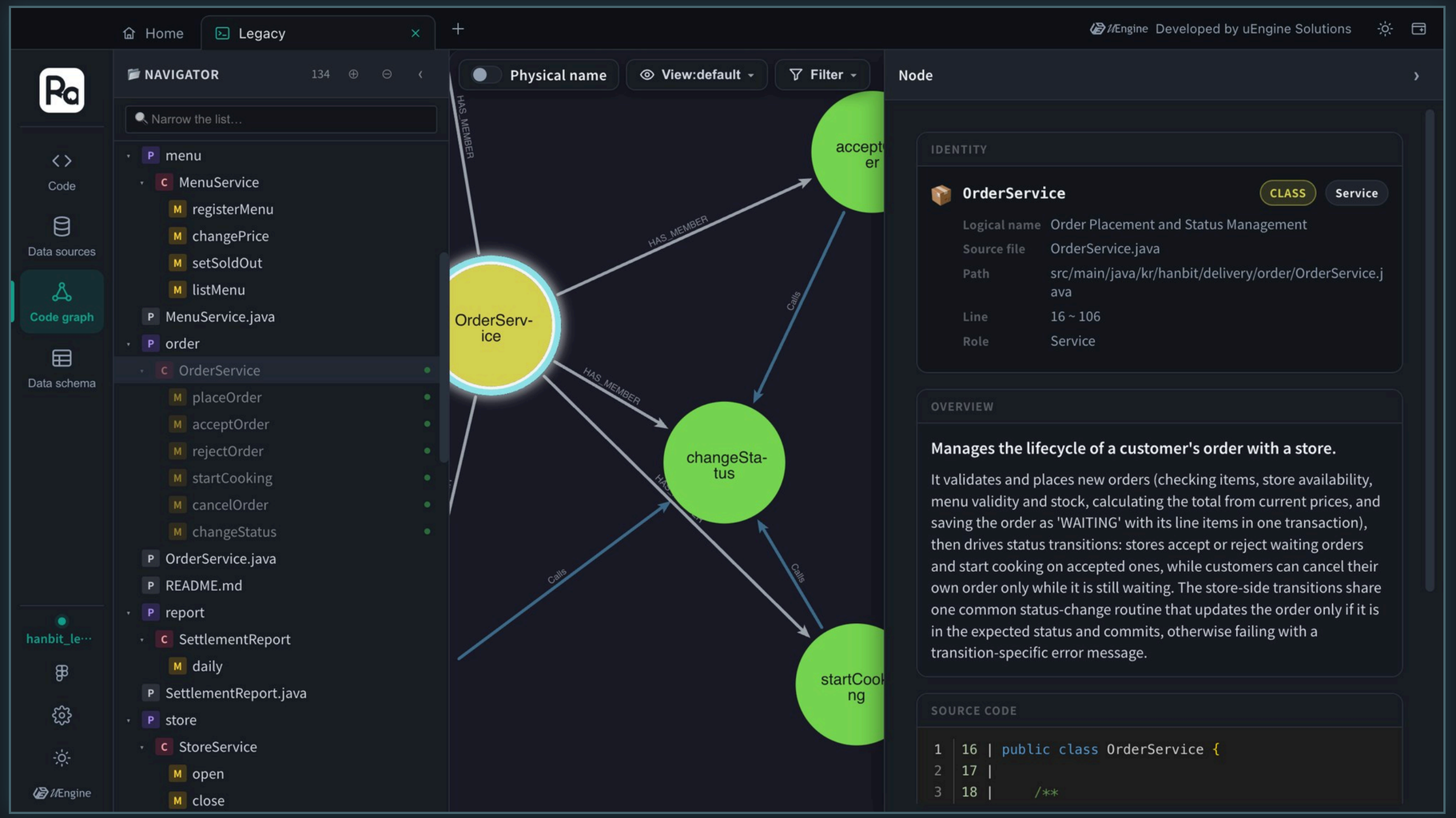
Eight stages, and determinism

1 DDL parsing → **2** AST loading → **3** External file loading → **4** Project scan → **5** Relationship linking → **6** Context → **7** Code analysis → **8** Post-processing

Structure, relationships, clusters and risk are computed without an LLM.

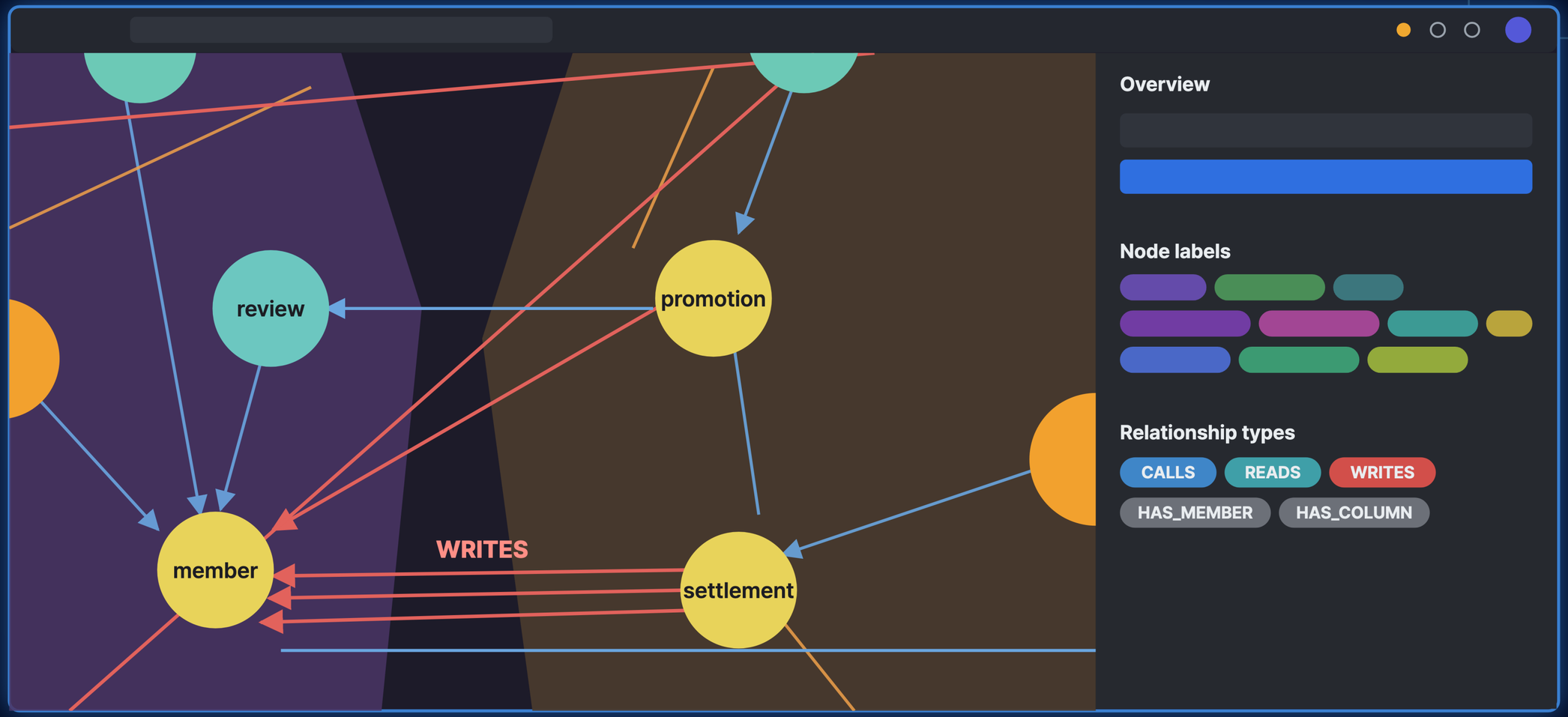
AI is used only to attach meaning, such as "what does this function do?"

Click a node, and the meaning is already attached



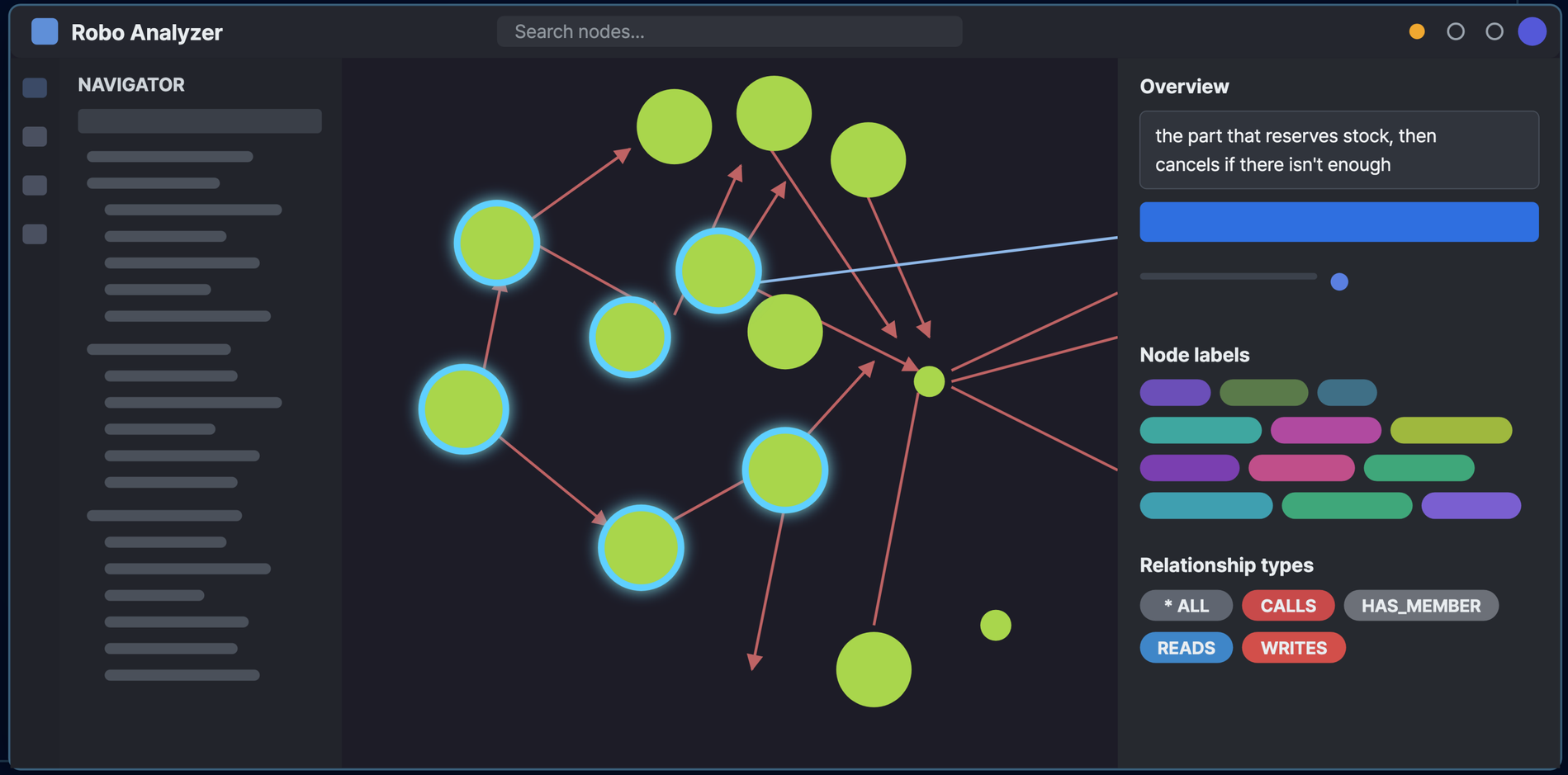
Next to the physical name: the logical name the analysis added, a [Key] summary, the function list and the original code, all in one panel.

Cohesion and coupling become visible



Inside a domain area: cohesion. Lines crossing the boundary: coupling.
Several strands of WRITES into another domain's tables are a warning sign.

Search in a business sentence, and the matches really light up



Type "the part that reserves stock, then cancels if there isn't enough" just as it is, and only the matching nodes are outlined.

It doesn't stop at showing the structure

Business rule extraction

Turns decision conditions scattered across the code into rules.

Example: "Allow the withdrawal if the balance is greater than 0."

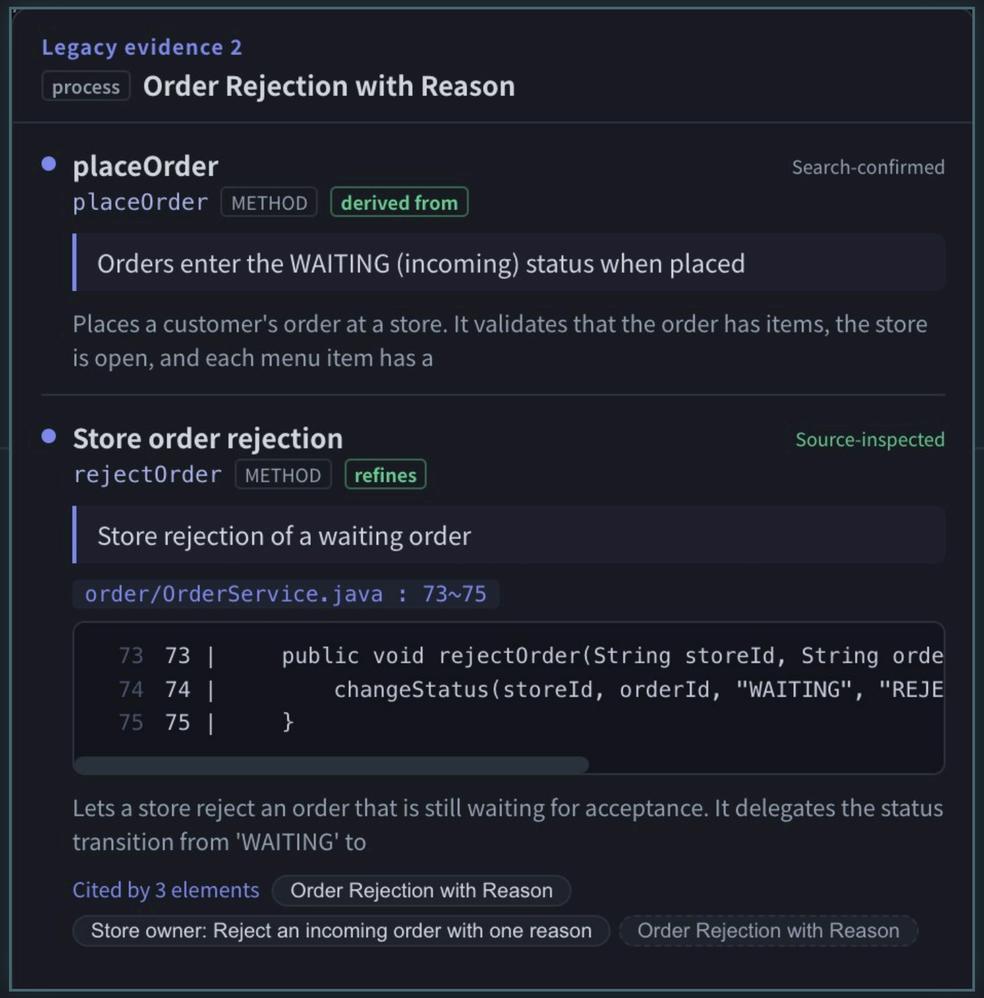
Organized into rules, cases and questions, and linked to the related tables.

Risk hotspot rating

Counts how often a function is called and how many calls it makes, code length, circular references and table coupling, and rates each one low, medium or high.

Circular references and oversized functions surface first.

Pick a source, and the actual code lines light up



Choose a read or write source on a table card, and the actual code lines are highlighted as they are.
"It reads this table" is proven with a file and line numbers.

Who needs it

Before a next-gen or modernization project

When you need to understand the current system from evidence

Handover

When you inherit a system whose owner has left or whose documentation is thin

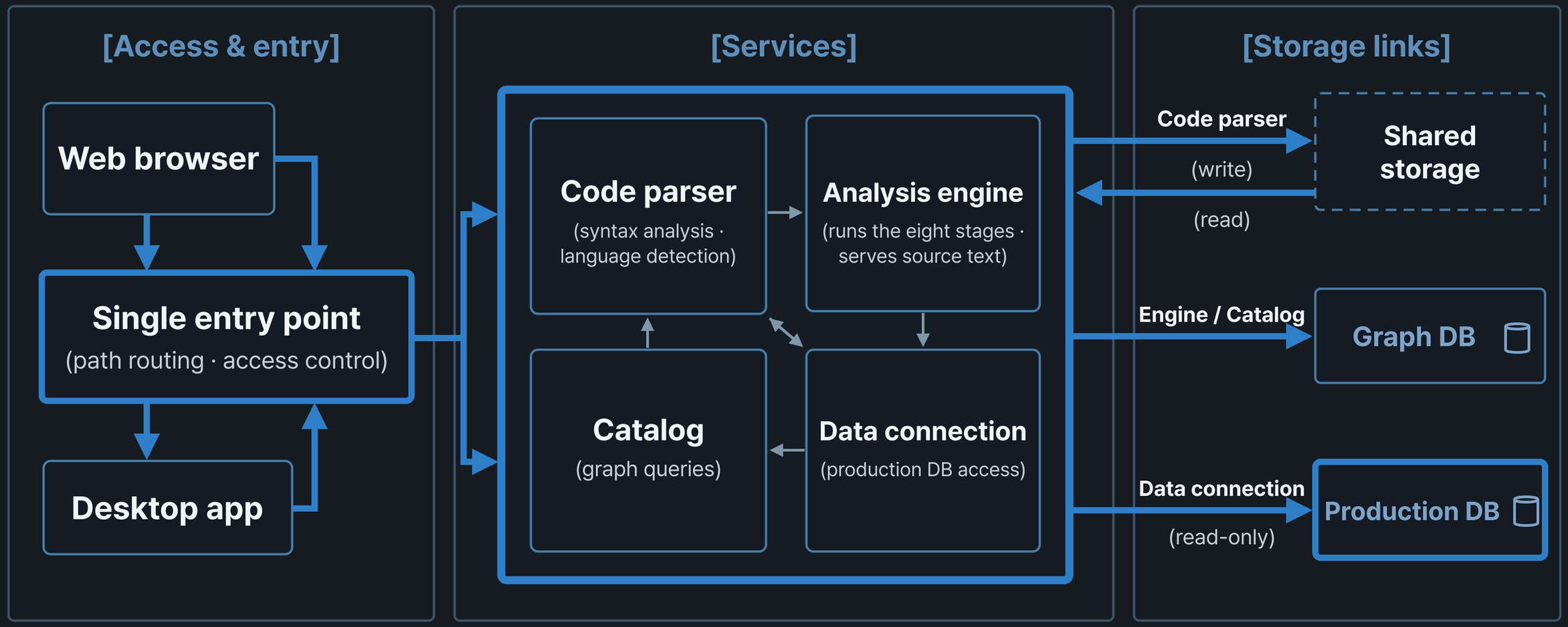
Checking change impact

When you need to see the scope of impact before changing a table or function

Docs and code disagree

When you need to check in the code itself which one is right

System architecture



Users reach it through a single address.

Deployment and access boundaries

Deployment options

- Web browser (no installation, just connect)
- Desktop app (installed)

Production DB access boundary

- Only read queries allowed; writes, updates and deletes are blocked
- The server enforces a row-count limit
- Connection passwords are never stored anywhere
- Only host, port, DB name and account name are kept

The production database is read-only.

One-page summary

Upload → Parse → Analyze → Graph → Explore

"AI doesn't make things up"

Structure, relationships, clusters and risk computed without an LLM.

"It analyzes broken code too"

Recovery is attempted only on the failed parts; the original is not modified.

"The production DB is read-only"

Writes blocked, passwords not stored, row-count limit.

Start with the system you use today, and understand it with evidence.

Robo Analyzer focuses on understanding the current system from code and data evidence.

Specification, design and implementation automation after analysis is handled by a separate product, Robo Architect.

Java · C · Python · Oracle PL/SQL · PostgreSQL

uEngine · www.uengine.org