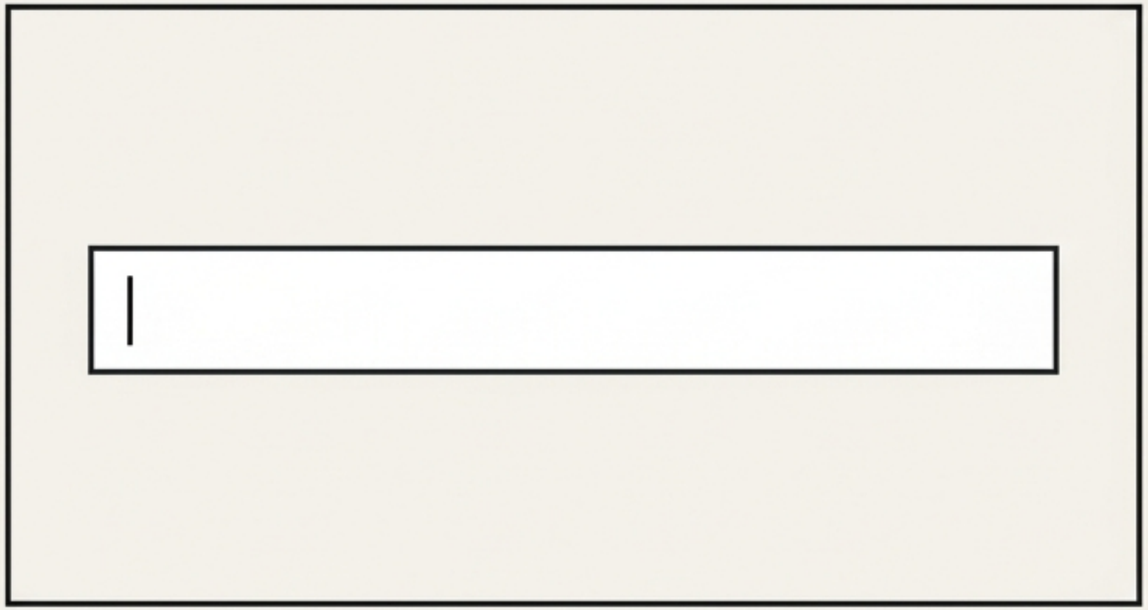
WHO ELSE?
The Architecture of Discovery

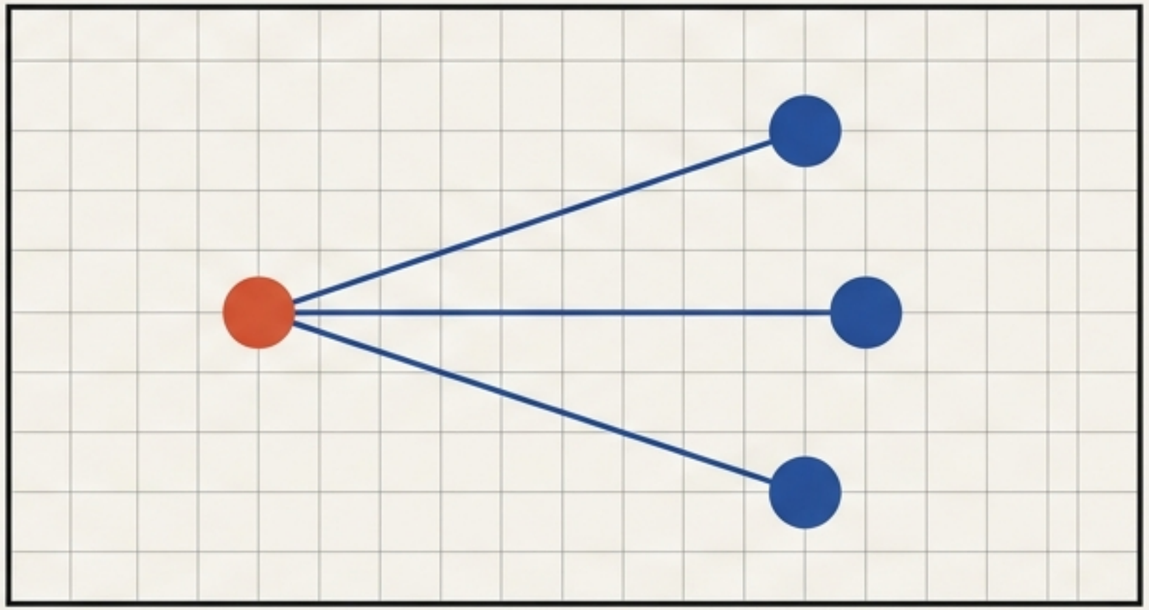
A thought experiment in language, discovery,
and self-replicating infrastructure.

Human Inquiry Begins With a Partial World

Most software begins on the machine side: an empty box demanding a perfect query. But a person rarely approaches a problem with total ignorance. They know one doctor and need another. They know one tool and need an alternative. They know one path through a bureaucracy and need to know what exists nearby. The internet demands we name the target. Human inquiry demands we expand from an anchor.



The Empty Box



The Known Anchor

The Linguistic Primitive

“Who Else?” is not merely a phrase. It is a computational pattern—a minimal discovery operator that opens a set beyond an established frame.

$$[\text{Known Anchor } A^*] + [\text{“Who Else?”}] = [\text{Adjacent Possibility } \{x\}]$$

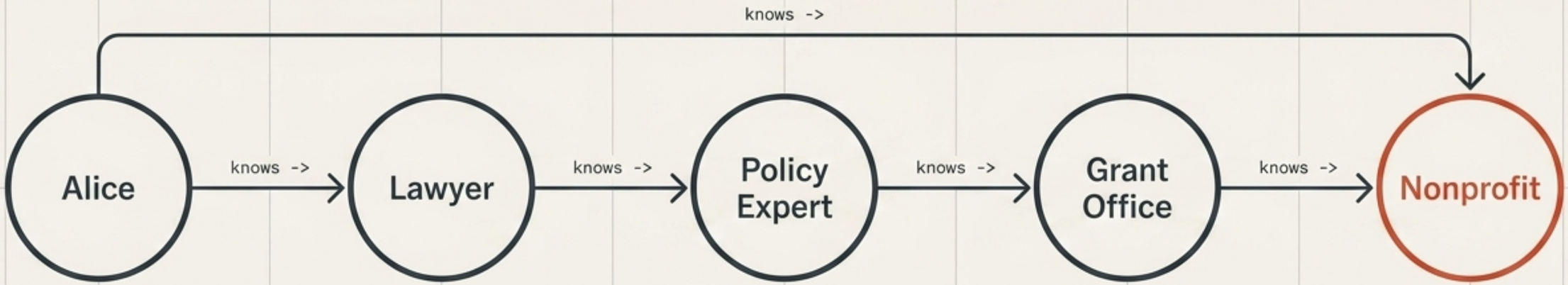
The Presuppositional Anchor Rule: The operation is felicitous only when a world is already partly populated. It asks for entities that satisfy the relevant predicate, but are not already in the known anchor set.

Search Retrieves Matches; Discovery Expands Context

Dimension	Search	Discovery (“Who Else?”)
Starting State	The Empty Box	The Known Anchor
Core Operation	“What matches my query?”	“What exists beyond what I know?”
Target Output	Exact Match	Productive Surprise
Underlying Model	Web of Documents	Living Graph of Entities

Cognition is a Path Through a Living Graph

People do not hold a complete table of all possible entities and retrieve rows on demand. They move by association, comparison, analogy, and social routing. One known thing leads to another. The user does not need to know the correct category, vocabulary, or market map. They only need one starting point. “Who else?” is the linguistic form of that movement.



The Self-Replicating Mechanism

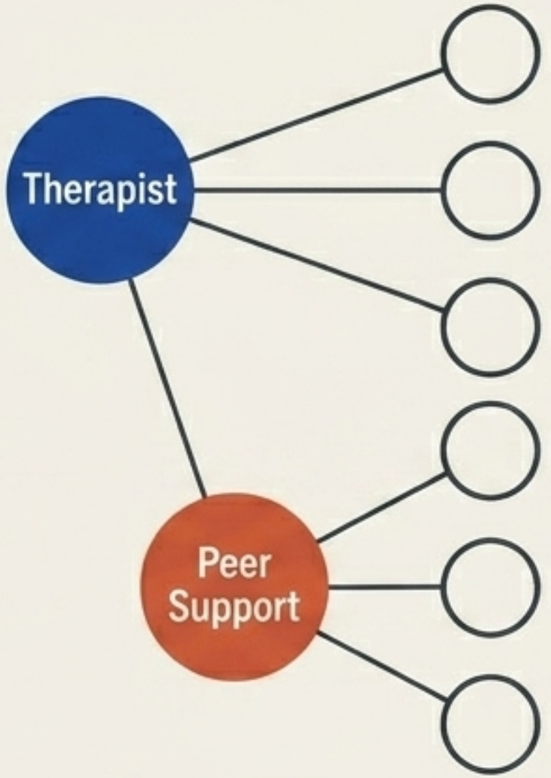
Every answer can become a new anchor.

1. **Anchor:** Therapist for anxiety -> “Who else?”
2. **Anchor:** Therapist for anxiety -> “Who else?”
2. **Answer Set:** Group clinics, peer support, telehealth.
3. **New Anchor:** Peer support communities -> “Who else?”

The system grows because its outputs become its next inputs.

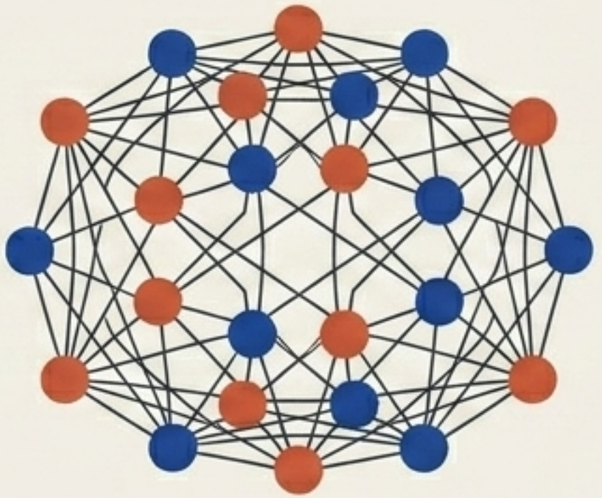
Growth is not just user acquisition.
Growth is graph thickening.

Step 1



Step 2

Step 3
Thickened Graph



The Technical Architecture

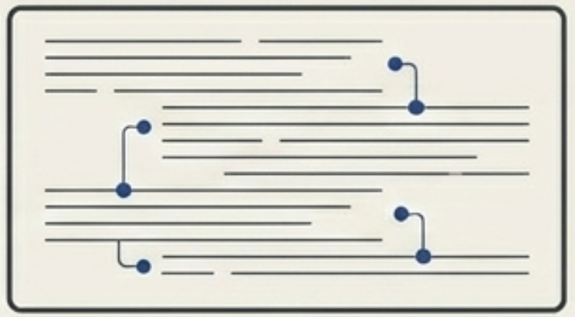
The system does not treat the internet as pages. It treats the world as entities and discovery-oriented relations.

The 6 Core Entity Types:
Platform, Organization, Person, Community, Tool, Locale.

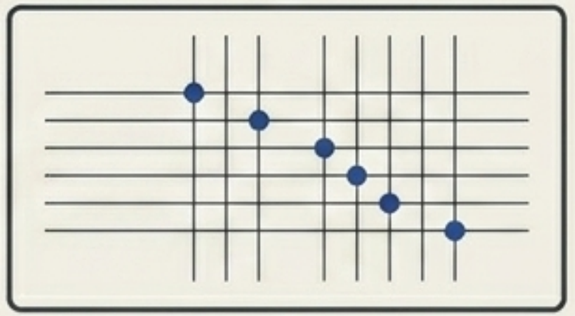
The Discovery Relations:
ALSO_ANSWERS, COMPLEMENTS, COMPETES_WITH, ENABLES, SUPERSEDES, TRUSTED_BY.

Relations do not merely describe the world. They support the next useful question.

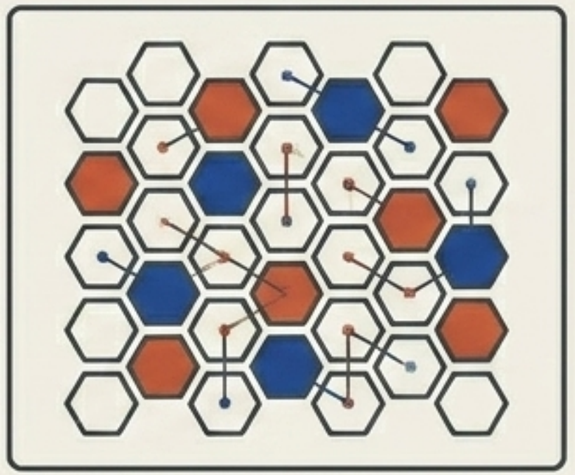
1. Input Parsing
(Natural Language)



2. Frame Mapping
(Intent Catalog)



3. Entity Graph



The Ranking Formula

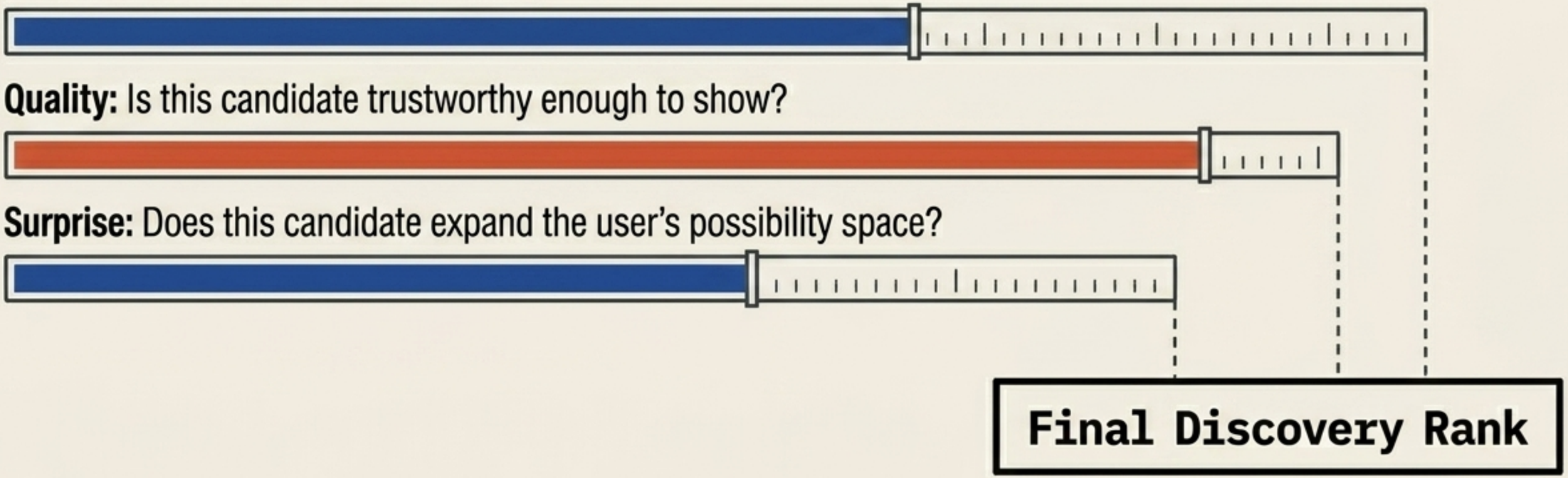
A discovery engine must return the most useful adjacent entities, not just the closest text match. Sometimes the best result is the one the user didn't know how to ask for.

Rank = Relevance × Quality × Surprise

Relevance: Does this candidate answer the frame?

Quality: Is this candidate trustworthy enough to show?

Surprise: Does this candidate expand the user's possibility space?

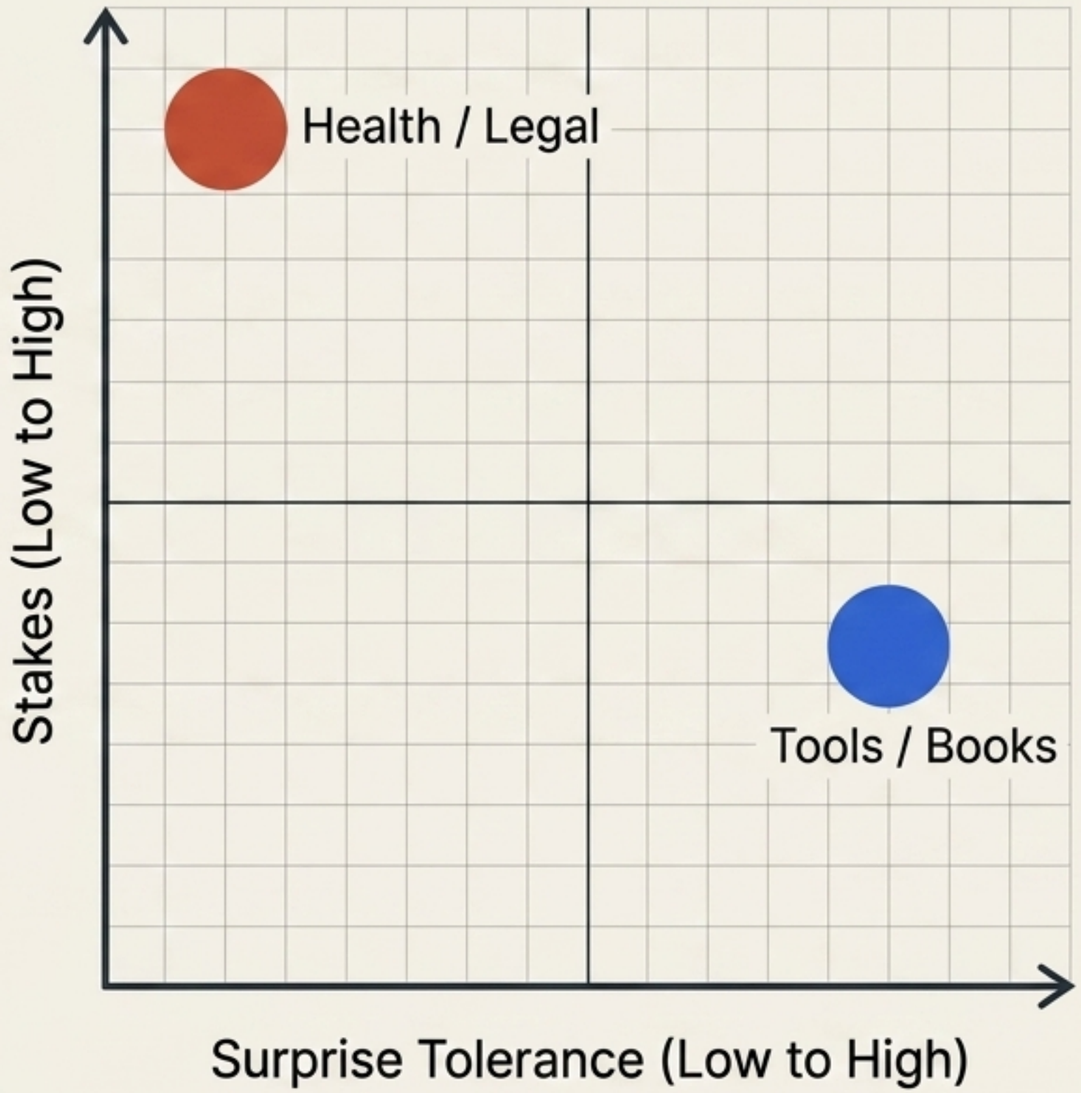


Quality is Multiplicative

The word “else” always expands. But good expansion is bounded.

For high-stakes categories (health, law, crisis), quality dominates surprise. The system must know when not to be clever.

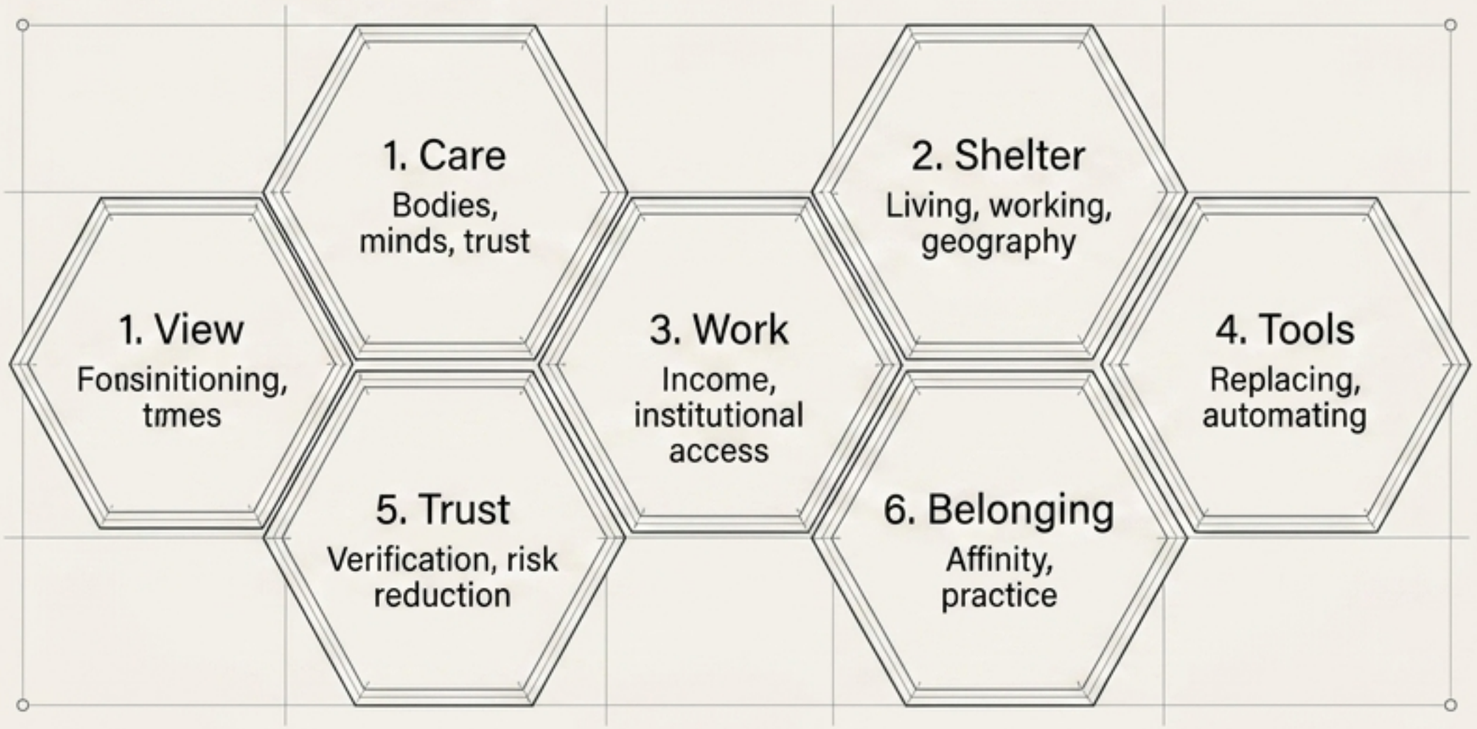
A candidate with high relevance and high surprise but low trust must be suppressed. A zero in trust zeroes out the result.



The Intent Taxonomy

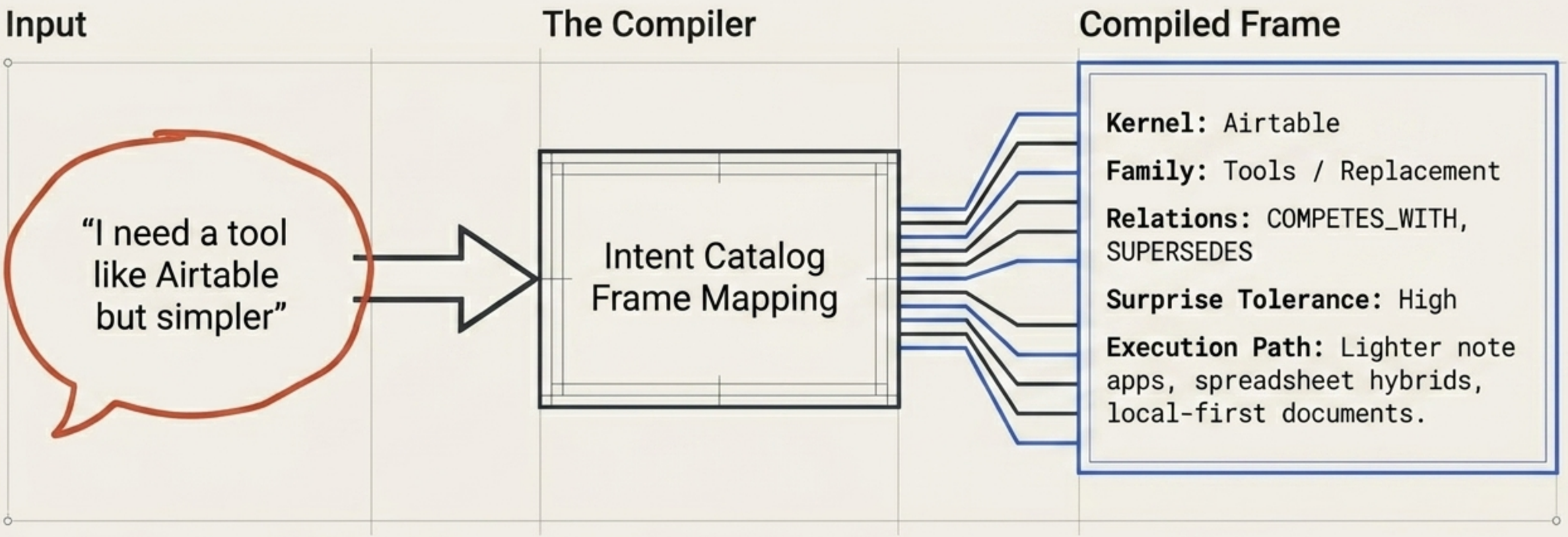
If “Who Else?” can apply to anything, the product becomes infinite and unlaunchable. We bound the system by mapping recurring human requests into six need-families.

Bounded Market Map



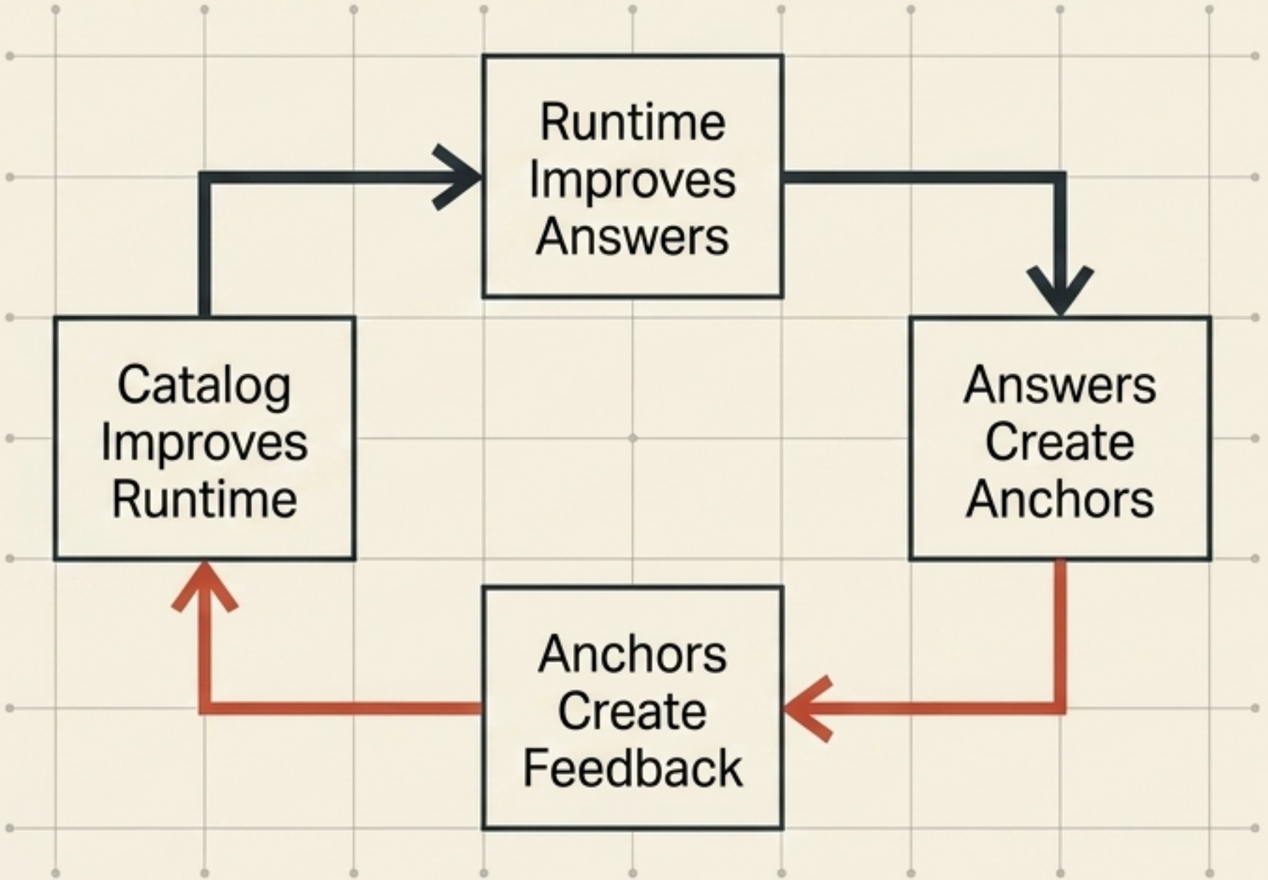
From Catalog to Compiler

The Intent Catalog is not a list of keywords. It behaves like a compiler, translating messy human urgency into structured graph execution.



The Build-System

Each vertical is not a separate company. Each vertical is an instance of the exact same grammar and shared graph runtime. The same runtime serves care, shelter, work, and tools because the catalog entries compile into shared constraints. Launching one category permanently reduces the cost and friction of launching the next.

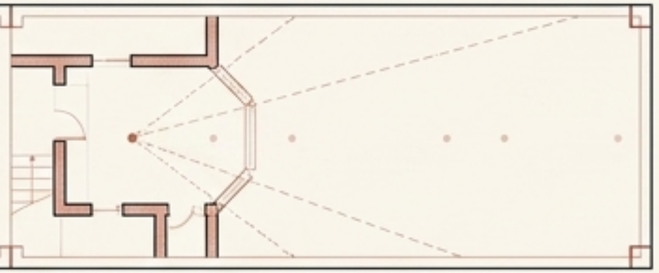


The Fundable Thesis

Who Else? names a missing internet layer: recursive discovery from known anchors into adjacent possibility.

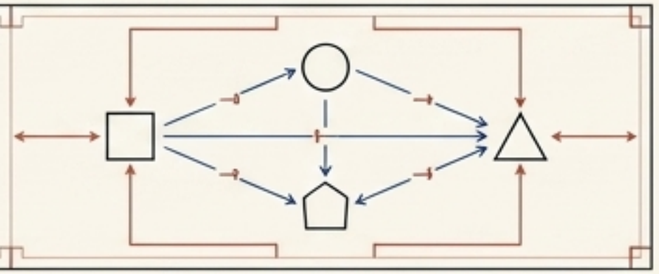
1. Underserved Behavior

Users start with partial knowledge. Search and marketplaces underserve partial knowledge.



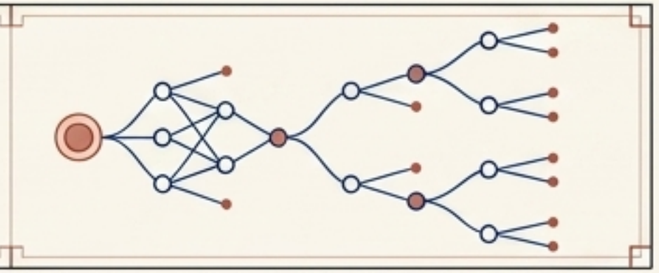
2. Bounded Execution

A small set of entity types and relation types can cover the vast majority of recurring discovery needs.

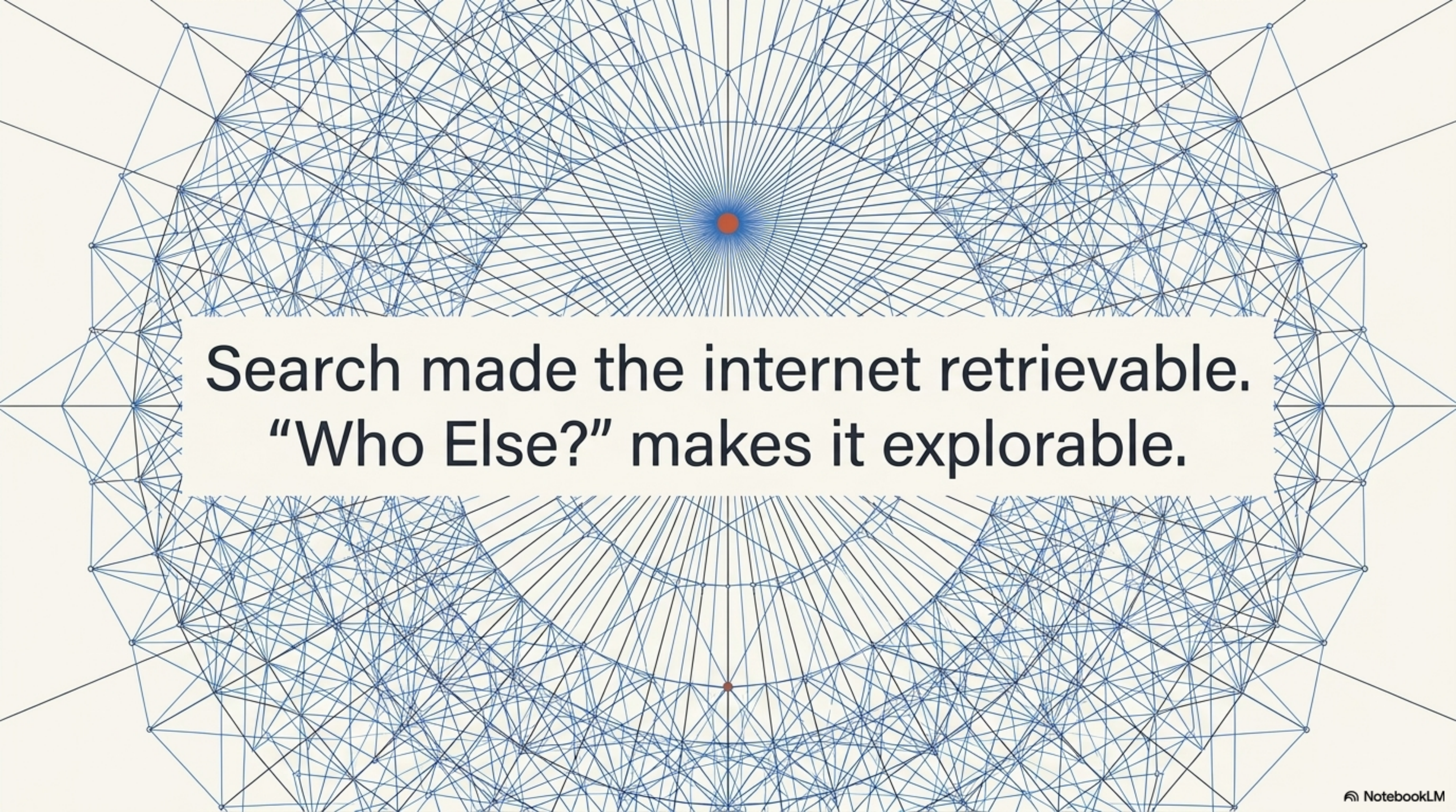


3. Compound Scaling

Recursive graph expansion improves with use. Every answered query becomes a seed for future answers.



Conclusion: **The system doesn’t just answer questions. It grows by answering them.**


Search made the internet retrievable.
“Who Else?” makes it explorable.