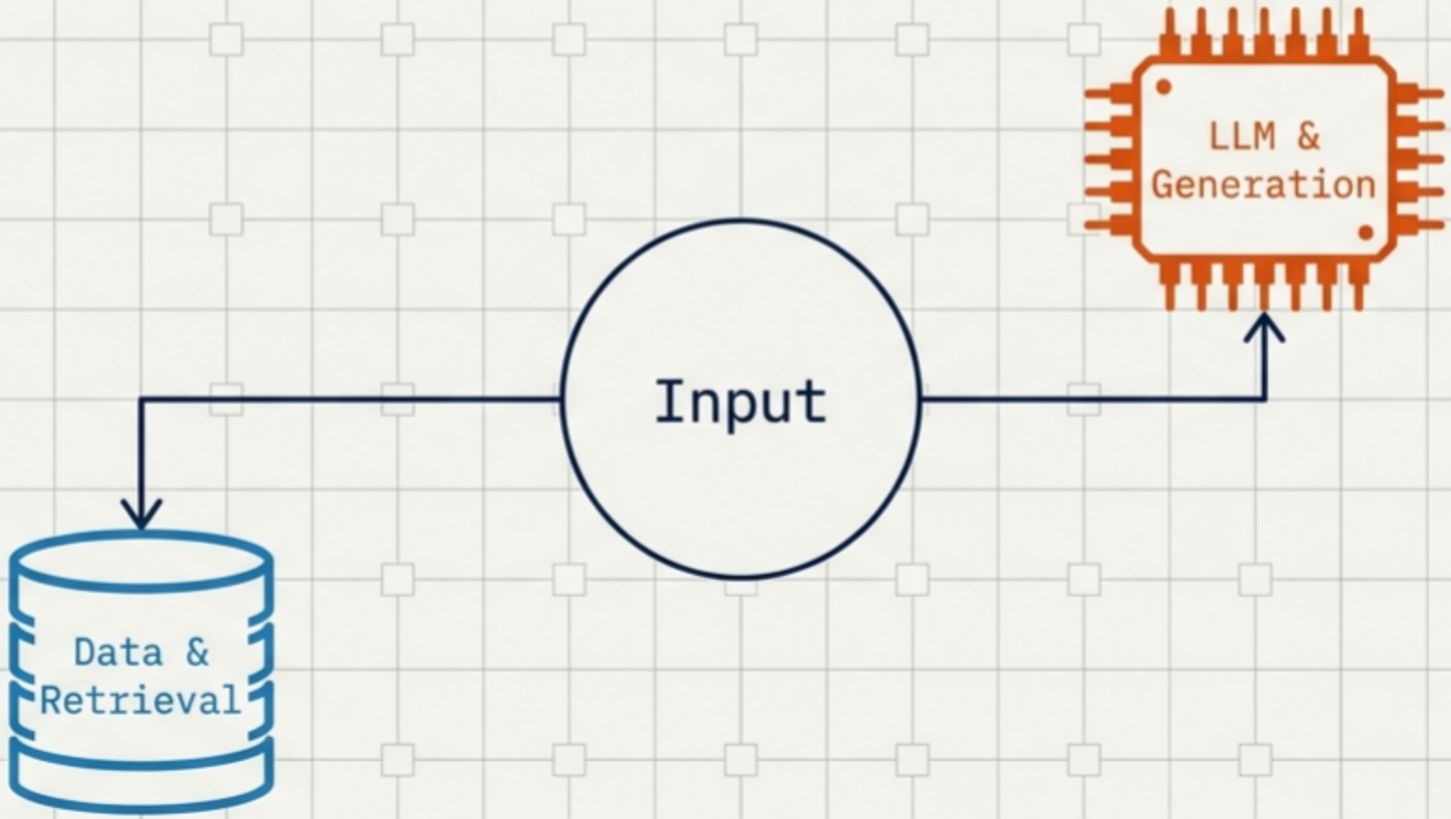
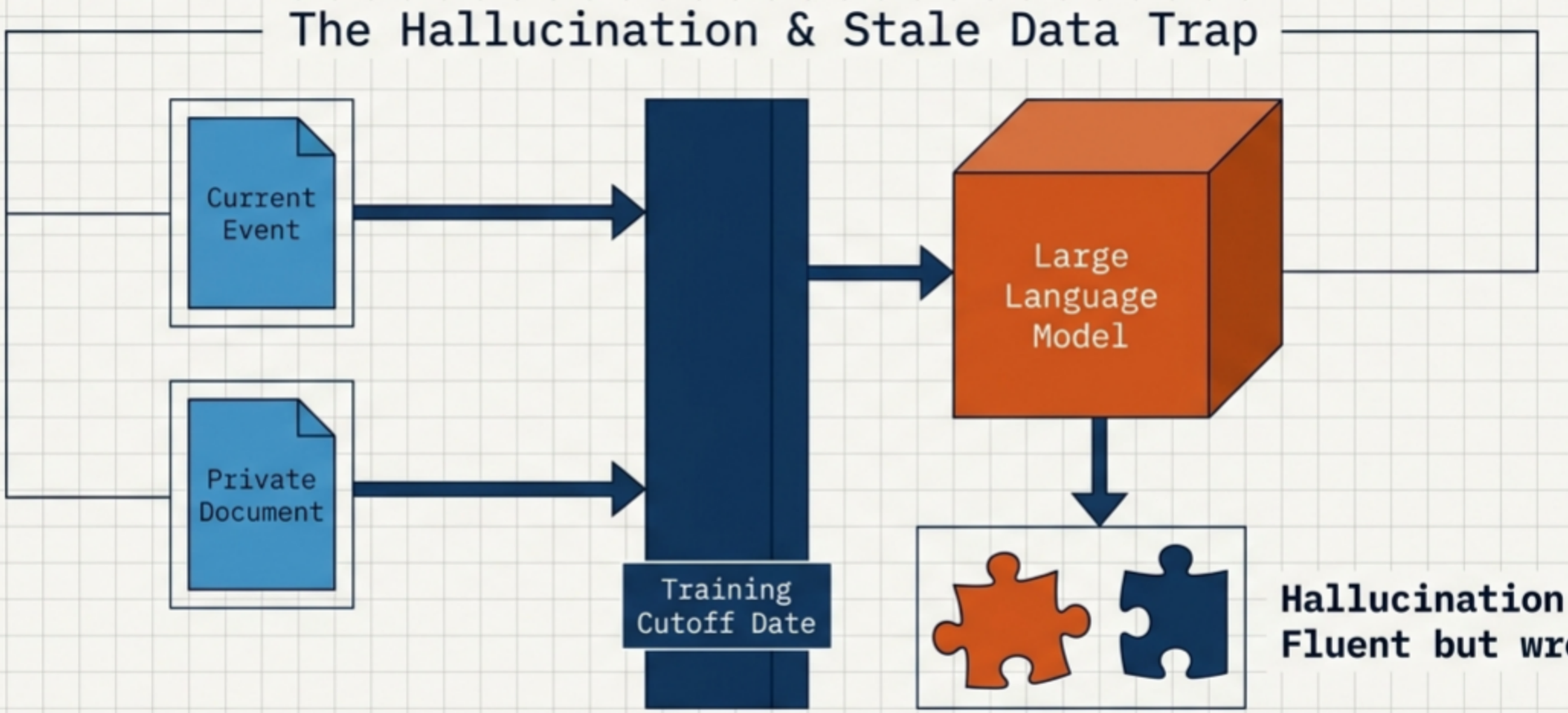
Architecting Trust in AI

A blueprint for building grounded, citable, and accurate systems using Retrieval-Augmented Generation (RAG).

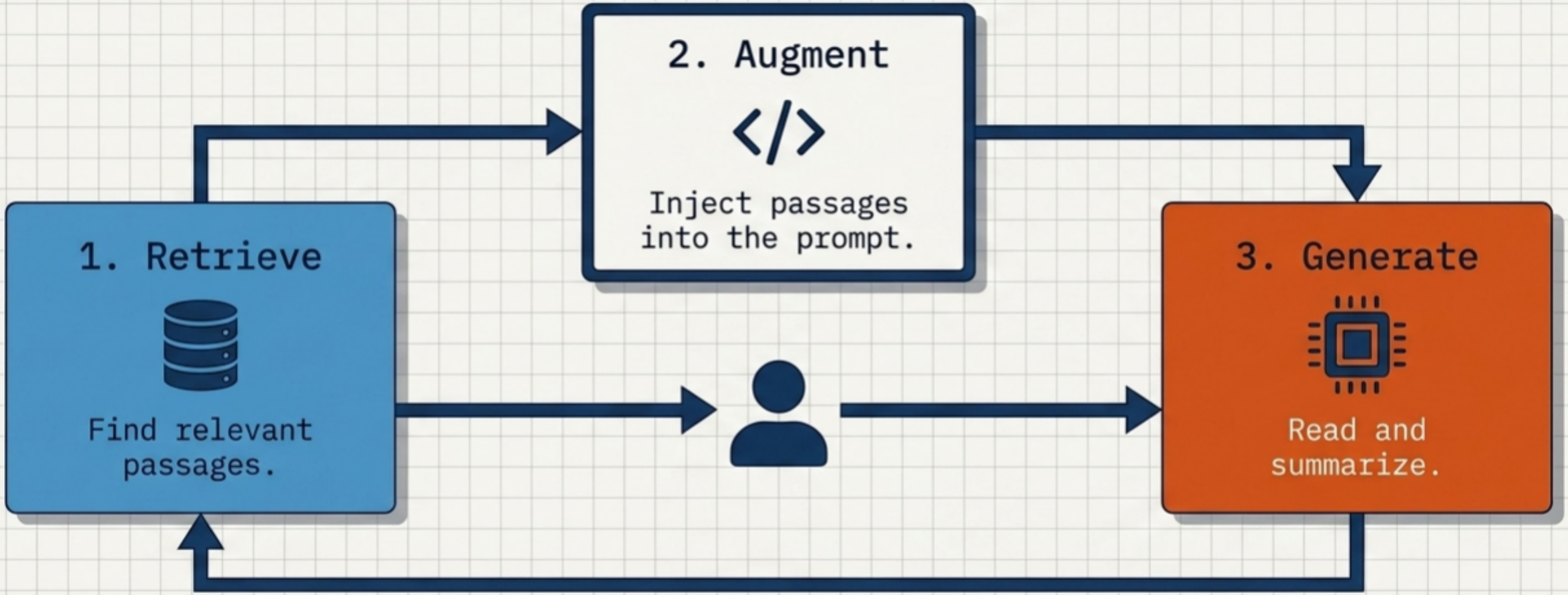


Bare LLMs suffer from the illusion of knowledge



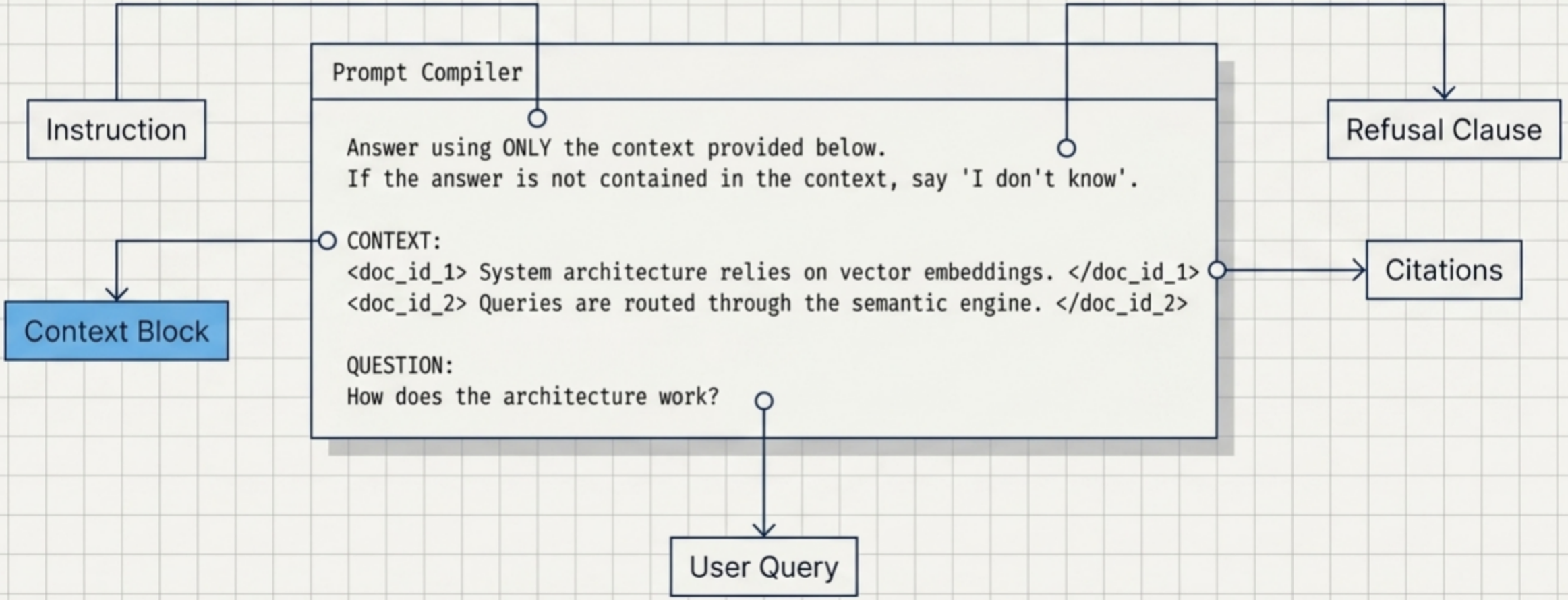
An isolated model only knows its training data. It cannot reference your private documents, and it cannot dynamically update its knowledge.

The RAG pattern structures data into a continuous, grounded loop



System division of labor: Retrieval supplies the facts. The model reads and summarizes.

Strict prompt architecture forces the model to adhere to the context



Citations anchor generated text directly to verifiable facts

The Citation Anchor Metaphor

The system architecture relies on vector embeddings [1].

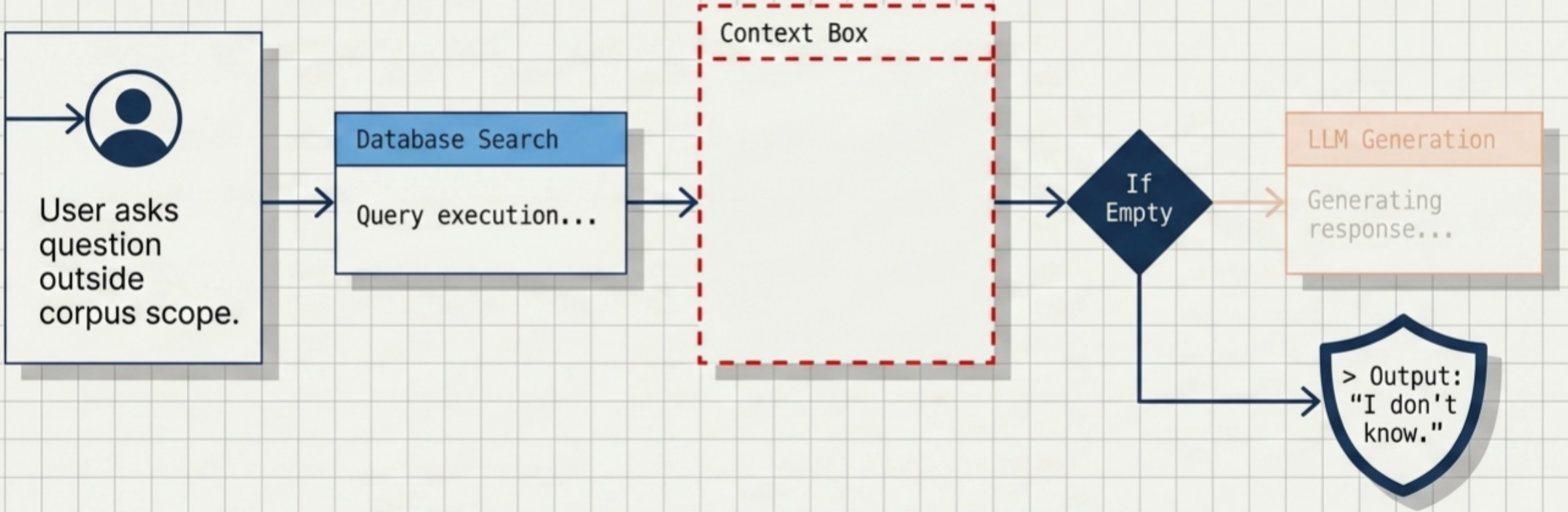
source: architecture_v2.pdf

The system architecture relience relies on vector embeddings to semante. Our systems while solors are connecting and ensure soluting the semantic meaning.
Our modernized system architecture relies on vector embeddings to store semantic meaning.

Retrieved passages are injected with ID tags.

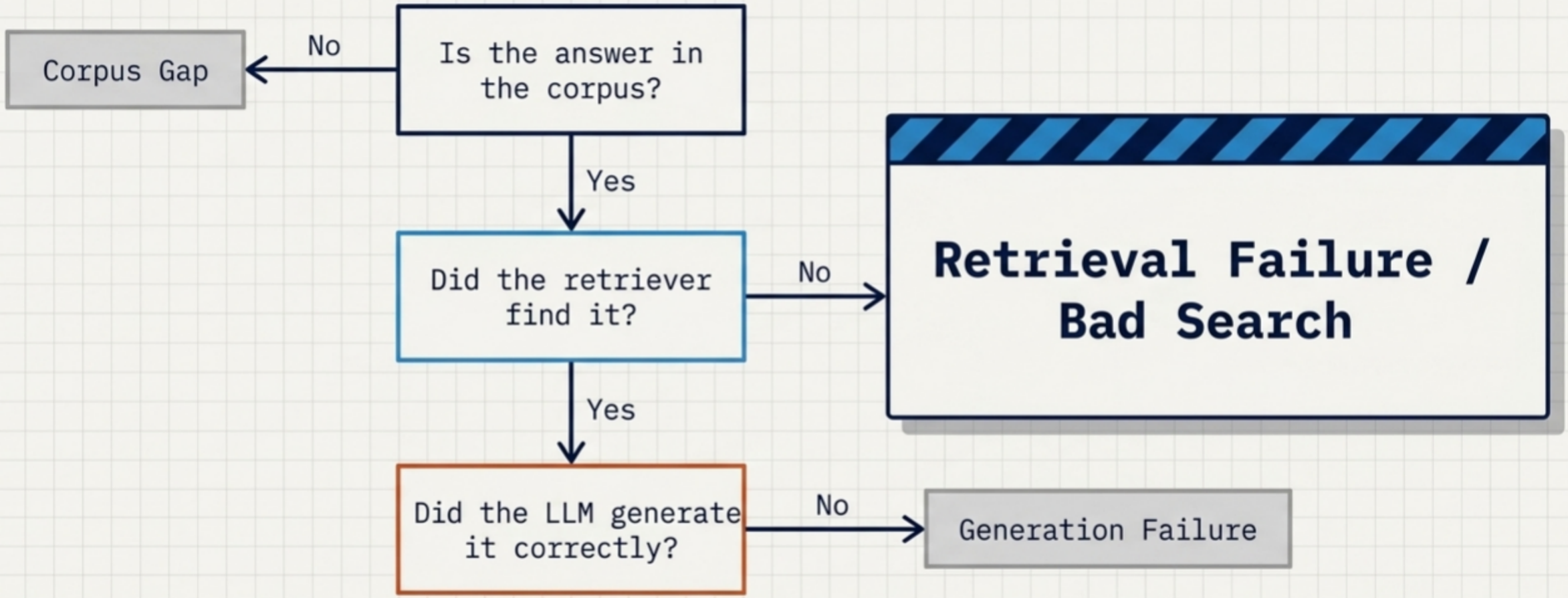
The model is instructed to append these IDs to the facts it generates, ensuring every claim is fully traceable.

A well-architected system knows exactly when to refuse



When asked something the corpus cannot answer, a grounded system **refuses instead of inventing.**

Most generation failures are actually retrieval failures



If the answer wasn't in the retrieved passages, the model has nothing correct to use. Improve the retriever before blaming the LLM.

Aligning the solution with your architectural constraints

RAG

Where Knowledge Lives
Injected at `query time`.

Update Frequency
Easy and instantaneous.

Citation & Traceability
High (`citable origins`).

Speed & Implementation
Requires external infrastructure.

Fine-Tuning

Where Knowledge Lives
Baked into `model weights`.

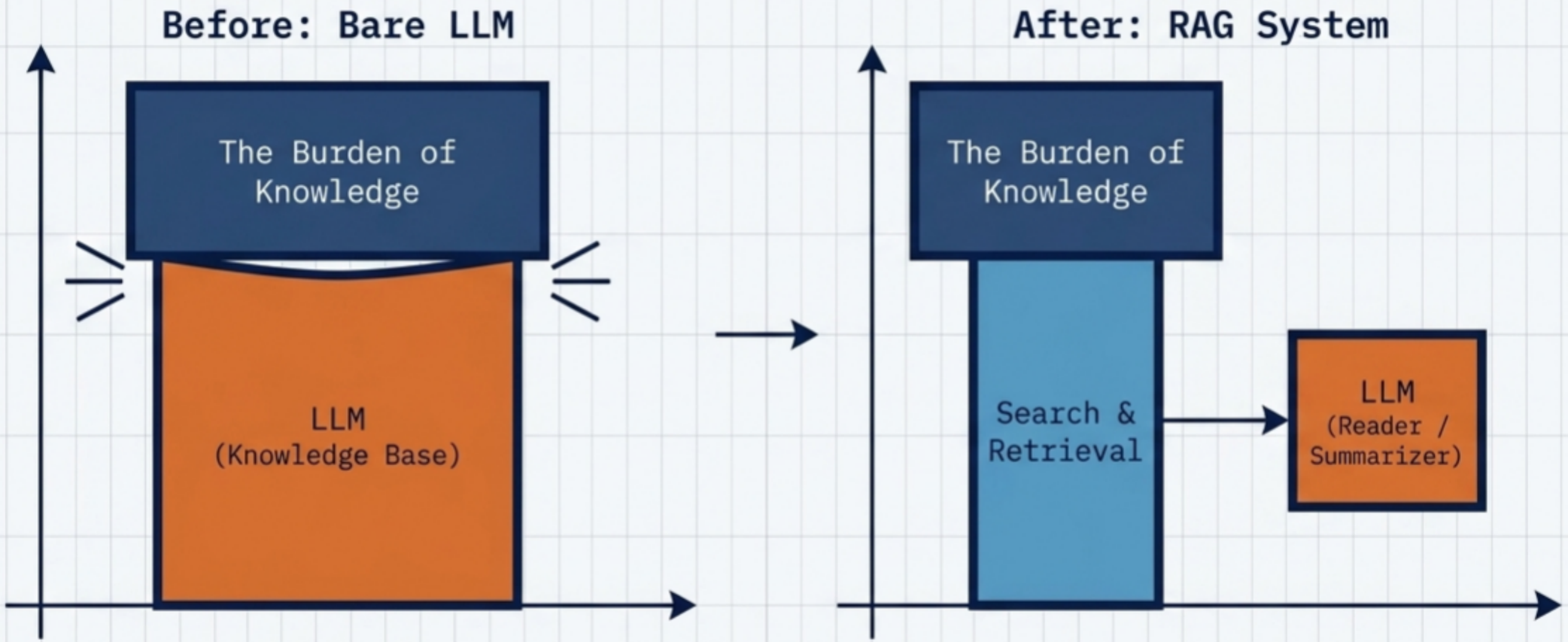
Update Frequency
Hard (requires `retraining`)

Citation & Traceability
Low (`black box generation`)

Speed & Implementation
Fast `generation once trained`.

They are complementary tools, not strictly competitors.

RAG demotes the LLM from an oracle to a synthesizer



By shifting the burden of factual retrieval away from the neural network and onto a dedicated search system, we eliminate the root cause of hallucination.

The essential vocabulary of semantic search architecture

LLM

Large Language Model.

Hallucination

Fluent but factually incorrect generation.

RAG

Retrieval-Augmented Generation.

Grounding

Anchoring model outputs to provided facts.

Context Window

The limited space for injected retrieval passages.

Refusal

Systemic instruction to admit a lack of knowledge.

Citation

Traceable links from generation to source.

Validating your mental model of retrieval-augmented systems

Q: What two problems does RAG solve?	A: Hallucination and stale knowledge, by grounding answers in current passages.
Q: Which two prompt clauses reduce hallucination?	A: ‘Use only the context’ and ‘say I don’t know if it isn’t there’.
Q: Why are most RAG failures actually retrieval failures?	A: If the answer isn’t in the retrieved passages, the model has nothing to use.