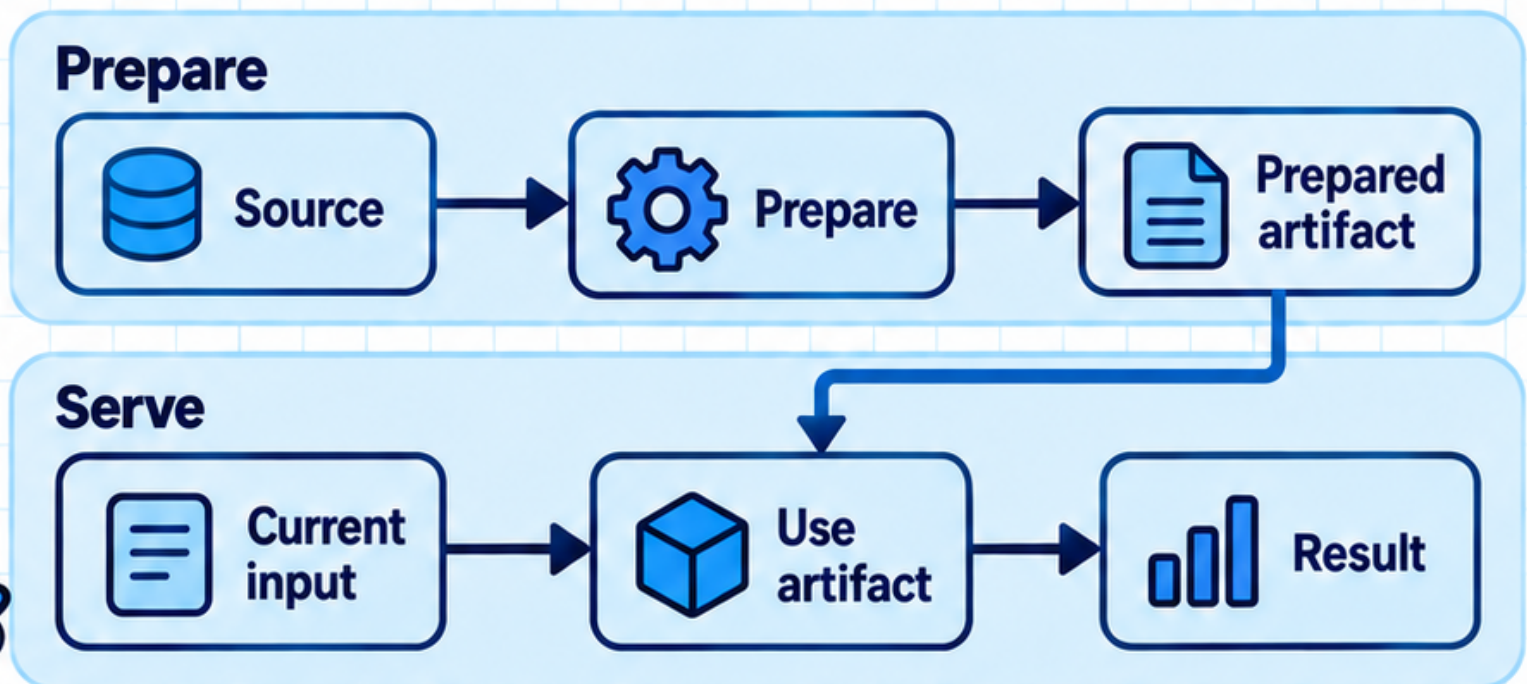


TWO PATHS. ONE WORKING SYSTEM.

Data preparation and request paths

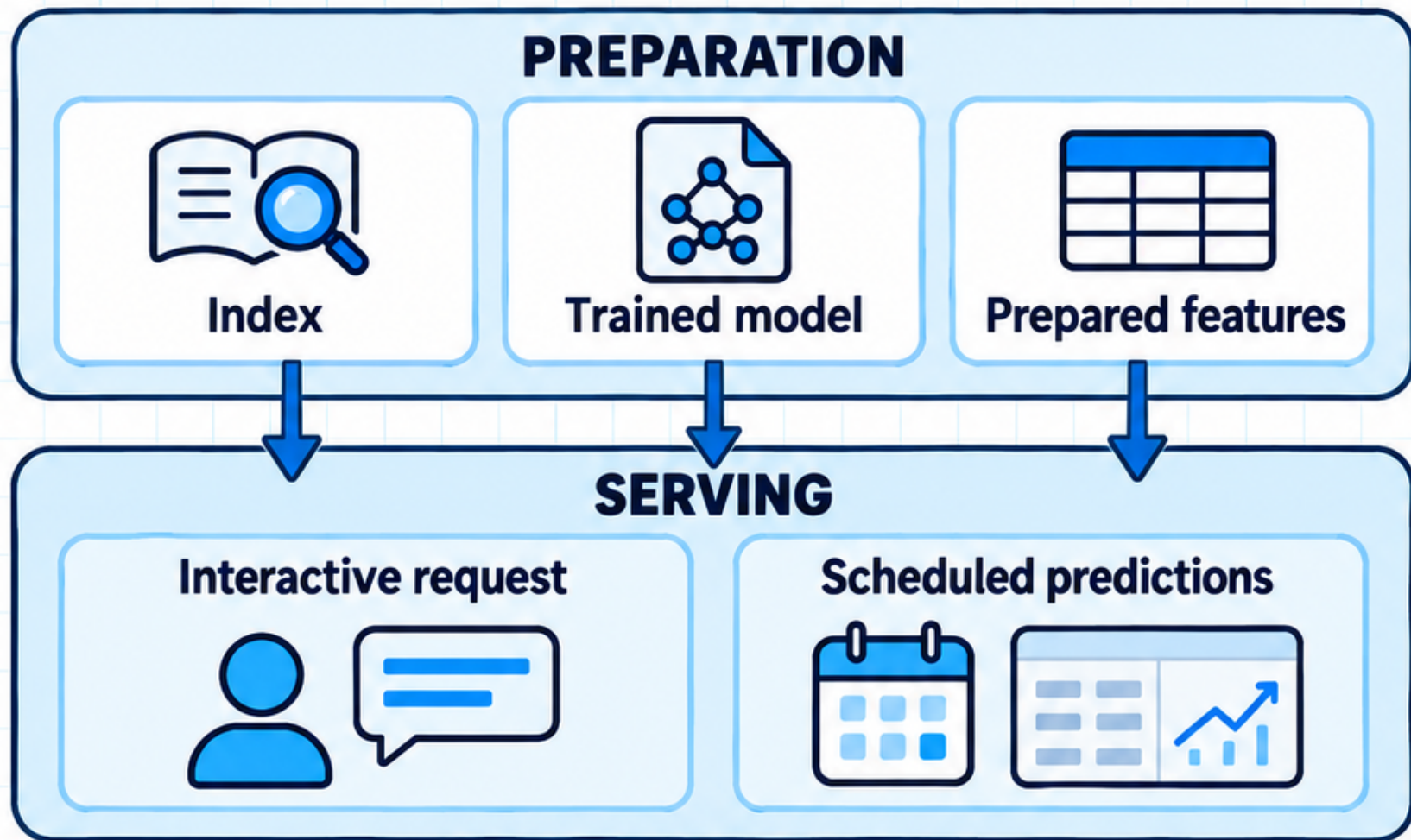
Some work prepares the data or model.
Other work produces a result for a user.



Today: trace both paths through a policy assistant and a demand forecast.

SEPARATE PREPARATION FROM SERVING

The preparation path creates something the application will use: an index, a trained model, or prepared features.



The serving path uses those artifacts to answer a question or make a prediction.

Serving can happen during a user request or in a scheduled batch.



START WITH THE SOURCE OF TRUTH

Our policy assistant starts with an approved return document.

ACCEPTED SOURCE



The document has an owner, a version, and an effective date.
A draft policy is not automatically an approved source.

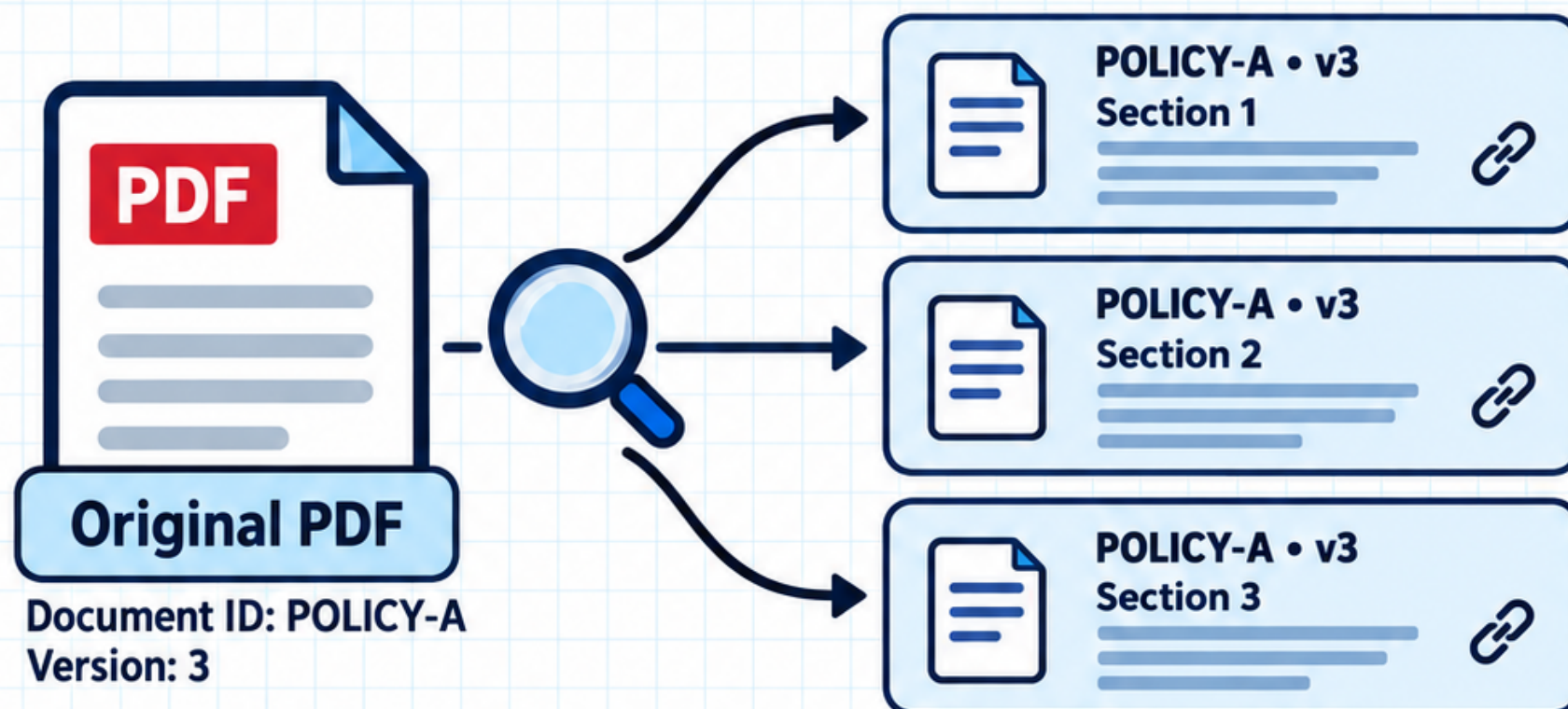


Before drawing a pipeline, identify who controls the source and which changes the system must accept.

KEEP ORIGINALS AND PREPARED RECORDS DISTINCT

The original PDF preserves the published document.

Prepared records contain extracted text, section boundaries, source references, and search metadata.

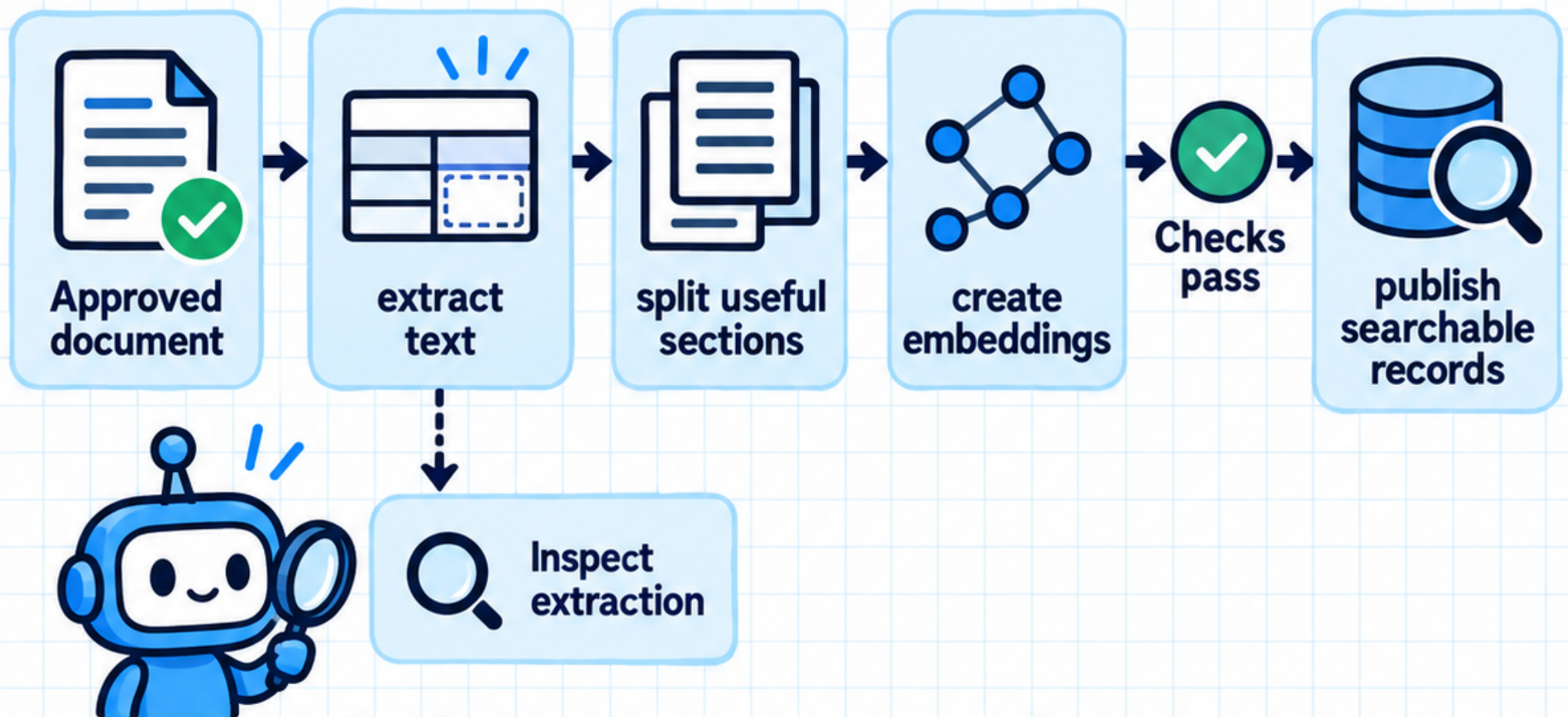


Fictional example



These records serve different purposes. Keeping their relationship lets the team inspect extraction errors and rebuild the search index after a processing change.

TRACE THE PREPARATION PATH



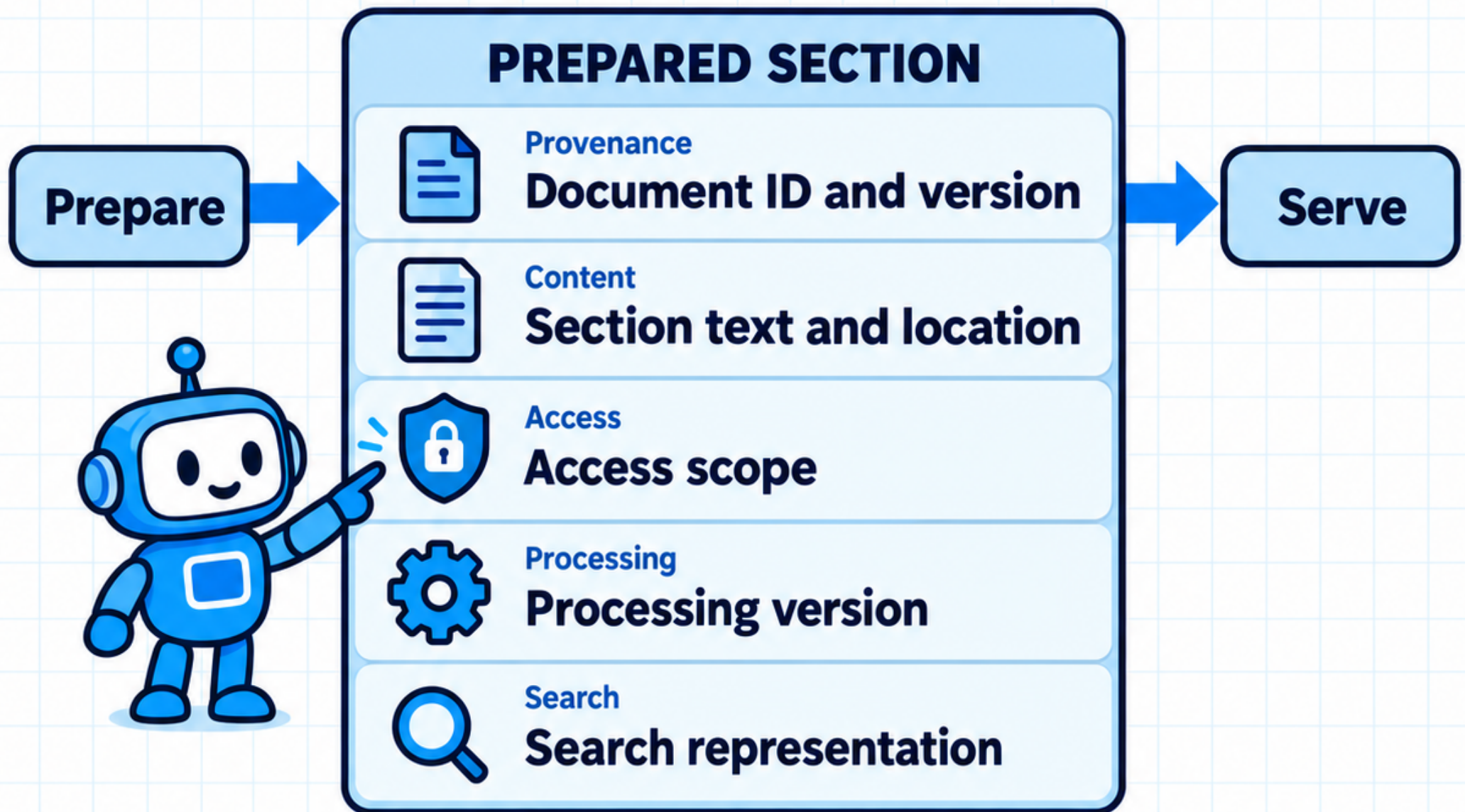
Each step has an output to inspect.



If extraction loses a table or an exception, later retrieval can miss essential evidence. A successful pipeline job does not prove that its content is correct.

GIVE THE HANDOFF A CONTRACT

A prepared section can carry:



The serving path depends on these fields. A handoff contract states which fields are required and what each one means.

NOW TRACE ONE USER REQUEST



A customer asks:
“Can I return an opened item?”

PREPARATION

Approved documents are processed in advance into prepared records.



Approved documents



Prepared records

Already prepared

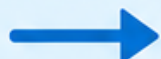
SERVING

The customer request is handled within the permitted scope.

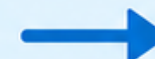


Customer

“Can I return an opened item?”



Identity and scope



Permitted search

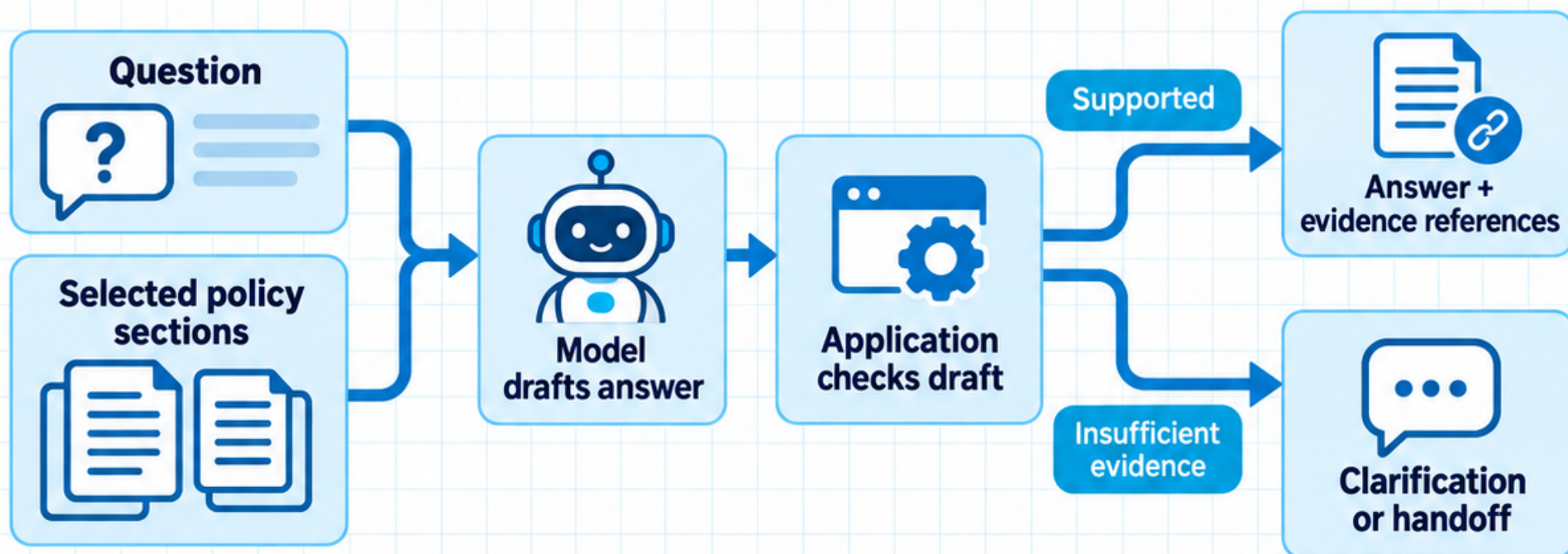
A customer asks: “Can I return an opened item?”

The application identifies the customer, validates the request, and determines the permitted policy scope.

Then it searches prepared records for relevant evidence. The question starts a retrieval request. It does not restart document preparation.

BUILD THE ANSWER FROM SELECTED EVIDENCE

The application sends the question and selected policy sections to the model.

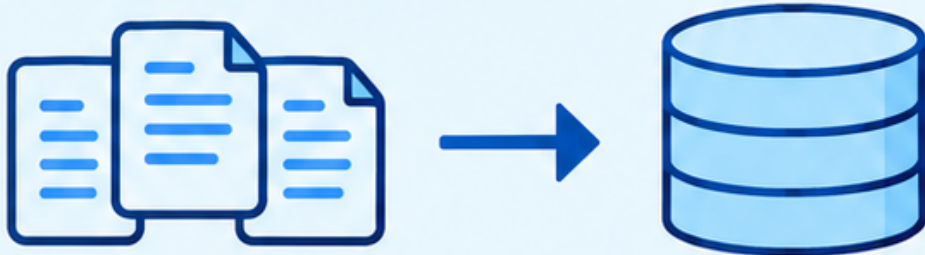


The model drafts an answer. The application checks the draft, attaches source references, and returns the result.

If the evidence does not answer the question, the defined response can be clarification or handoff instead of a guessed policy.

RETRIEVAL IS DIFFERENT FROM MODEL TRAINING

RETRIEVAL UPDATE



Update documents

Search index

Adding a policy to a search index changes the knowledge available for retrieval.

Changes available evidence

MODEL TRAINING



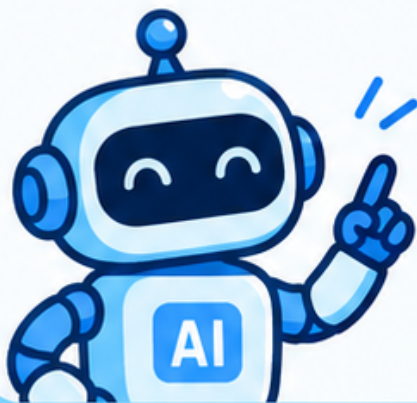
Training data

Training

Model parameters

Training changes model parameters.

Changes learned parameters



A RAG application can use an existing model and update its documents independently. The design still needs compatible embeddings, useful retrieval, and evaluation of the resulting answers.

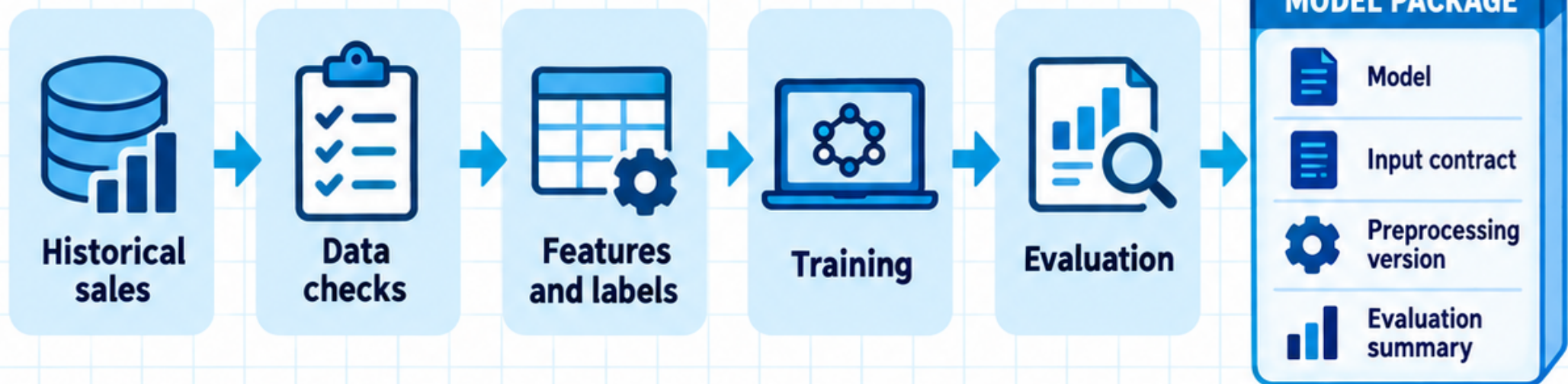


Existing answer model

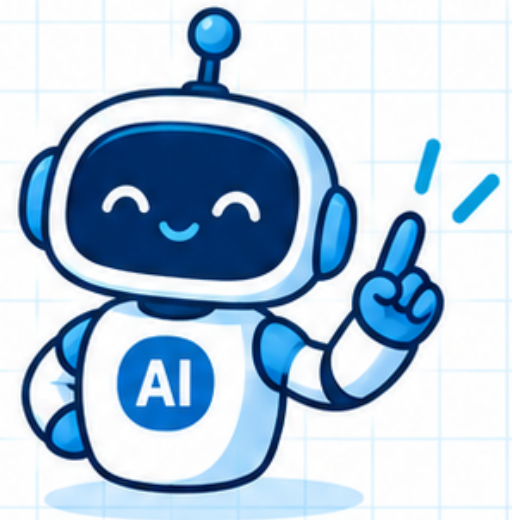
THE ML PREPARATION PATH CREATES A MODEL

For a demand forecast, preparation can include:

Historical sales → data checks → features and labels → training → evaluation → approved model artifact.

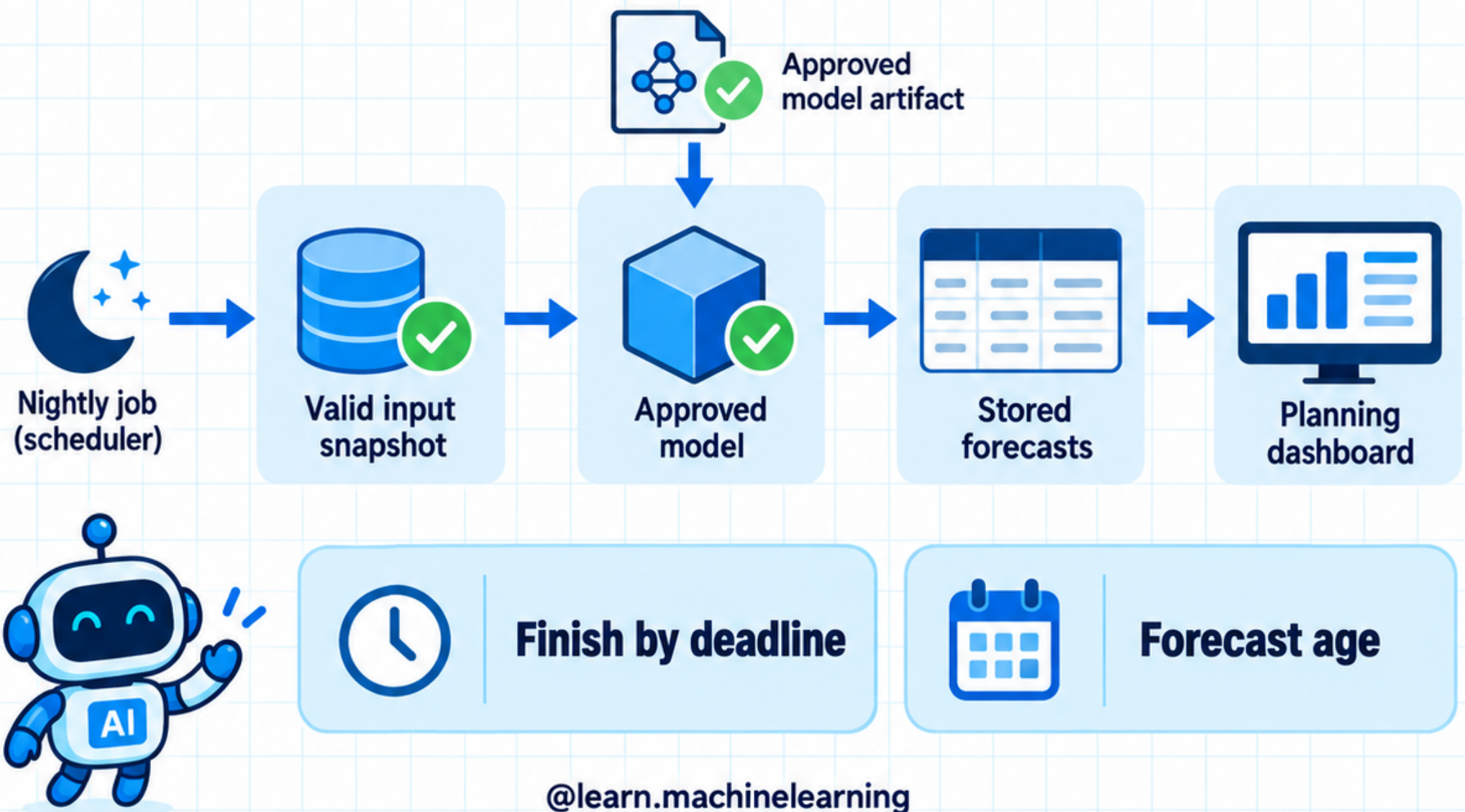


The artifact needs its preprocessing rules and version information. A model file alone cannot explain which inputs it expects or how the team evaluated it.



BATCH INFERENCE IS ALSO A SERVING PATH

A nightly job loads the approved model and the latest valid inputs. It predicts tomorrow's demand and stores the results for a planning dashboard. No customer waits on an open chat request. The useful response-time requirement is the batch deadline and the age of the displayed forecast.

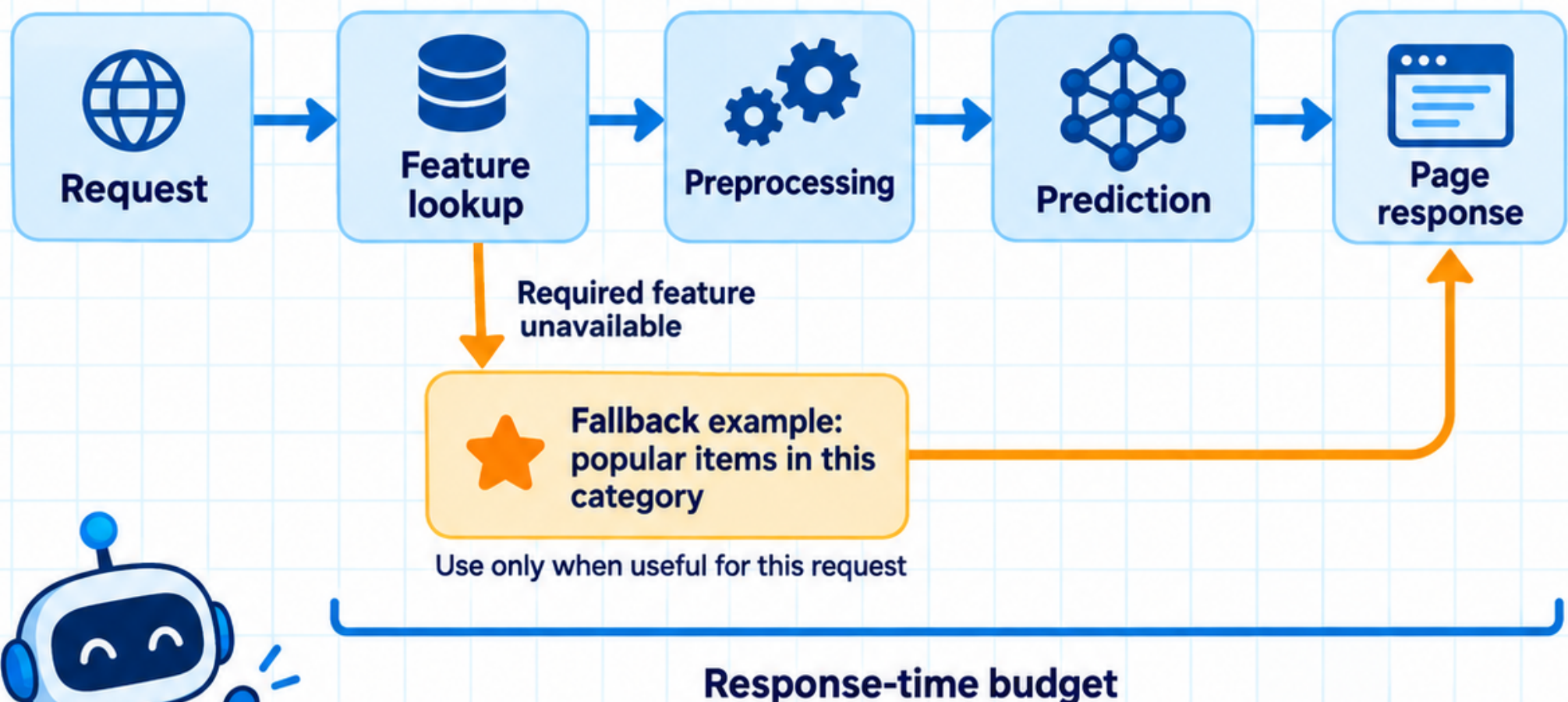


ONLINE INFERENCE HAS A DIFFERENT BUDGET

An online recommendation request needs features and a model result before the page responds.

Feature lookup, preprocessing, and prediction all consume time.

If a required feature is unavailable, the application needs a defined fallback. The design must state whether that fallback remains useful for this particular request.



TOOLS HAVE SPECIFIC JOBS

Object storage: preserve source files and model artifacts.

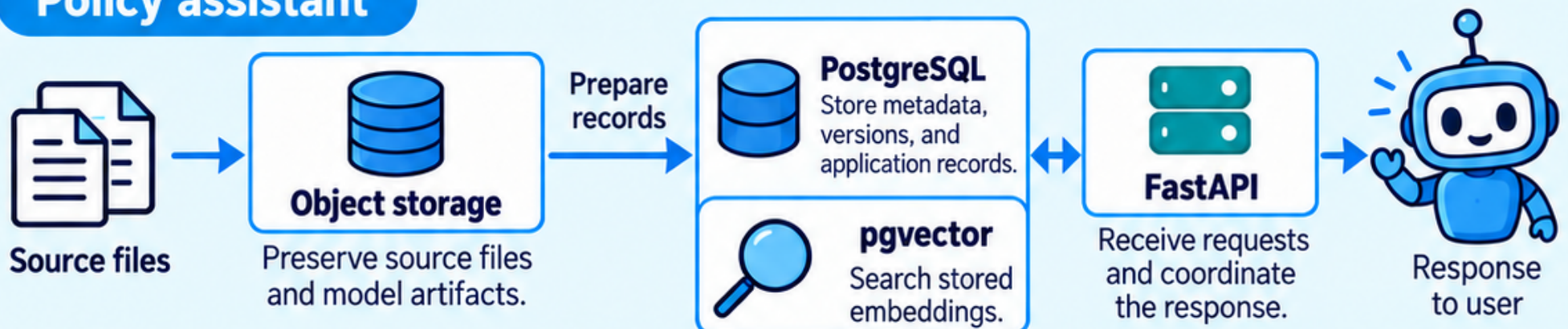
PostgreSQL: store metadata, versions, and application records.

pgvector: search stored embeddings.

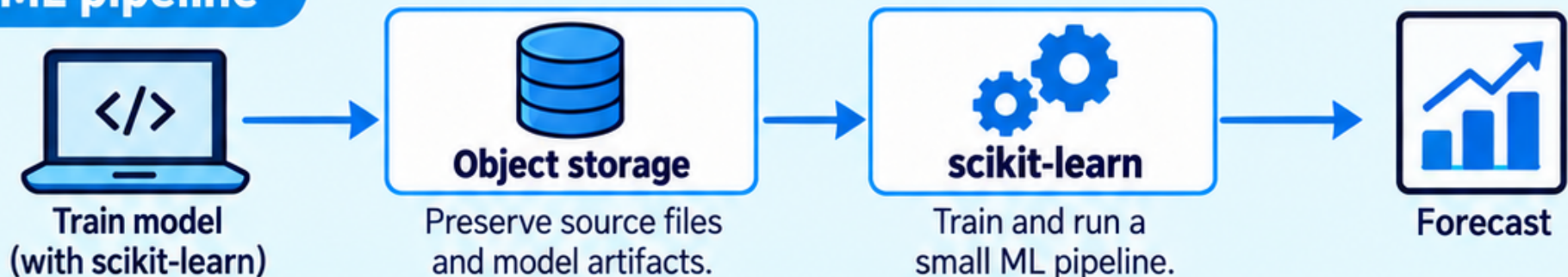
FastAPI: receive requests and coordinate the response.

scikit-learn: train and run a small ML pipeline.

Policy assistant



ML pipeline



Choose each tool for its responsibility in the flow.

CONNECT THE PATHS WITH VERSIONS

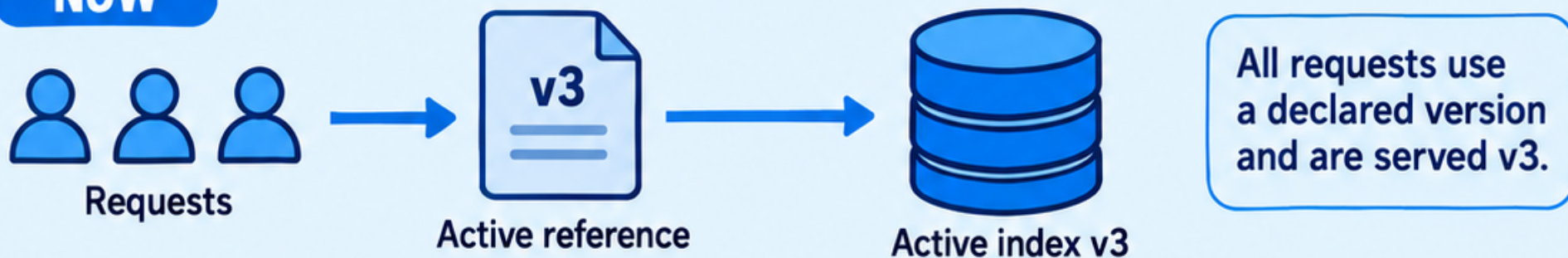
Suppose a new policy index is only half prepared.

Serving directly from that incomplete update can mix old and new records.

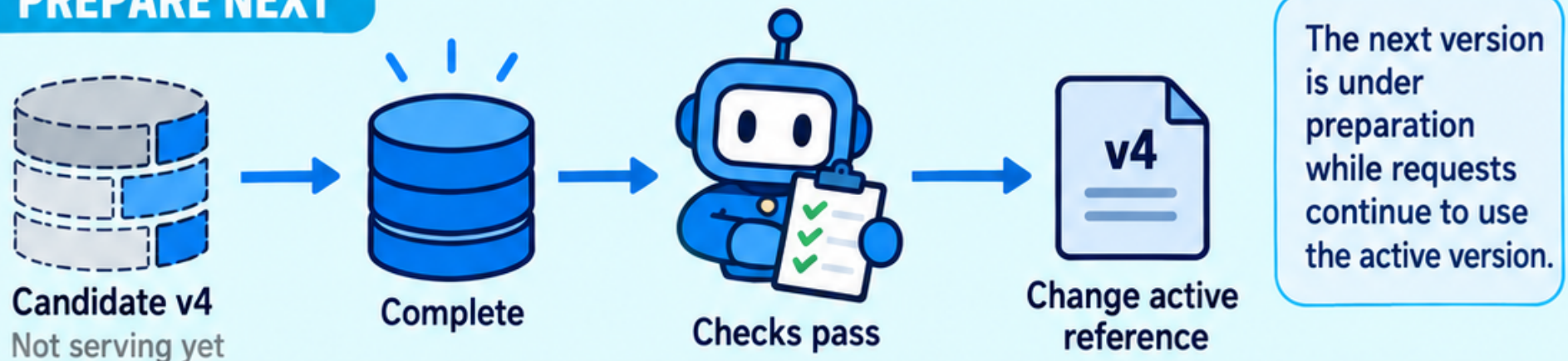
One design prepares a complete version, checks it, then changes the active reference. Requests use a declared version while the next version is under preparation.

One design option • Illustrative versions

NOW



PREPARE NEXT

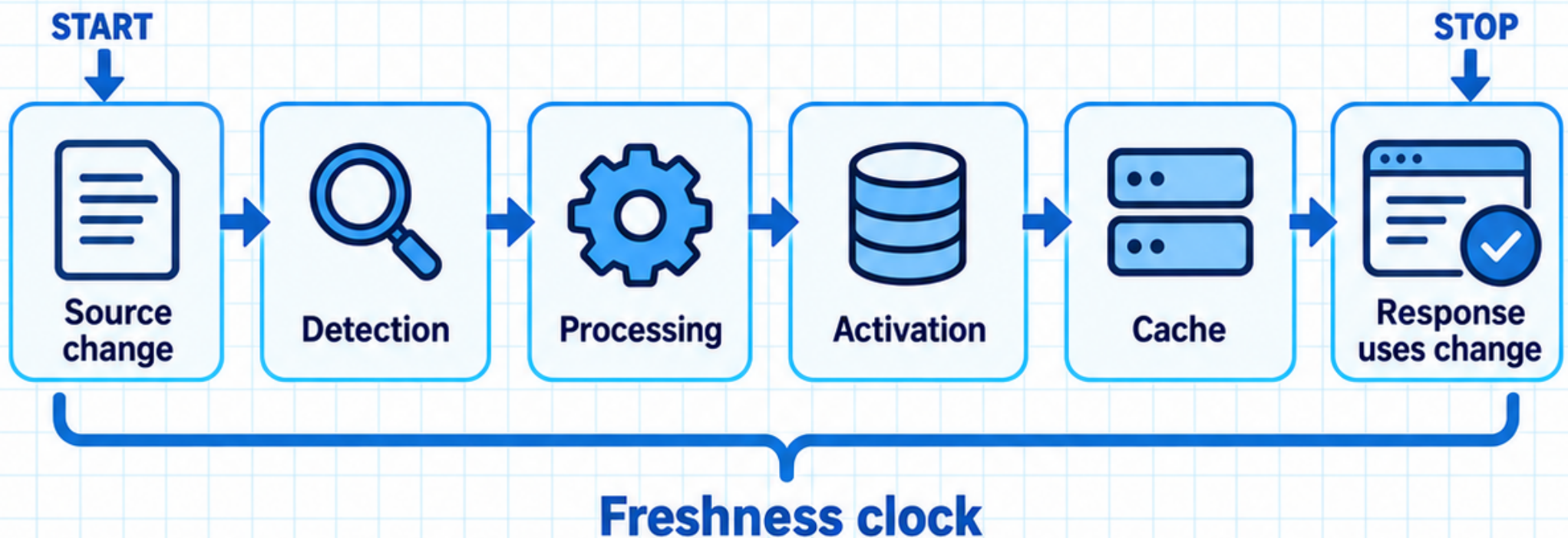


FRESHNESS CROSSES BOTH PATHS

Freshness is the delay between a relevant source change and the result that uses it.

The delay can include detection, processing, publication, and a cached response.

Measuring only the ingestion job misses part of the path. Start and stop the freshness clock at explicitly defined events.



GIVE EACH PATH A FAILURE RESPONSE

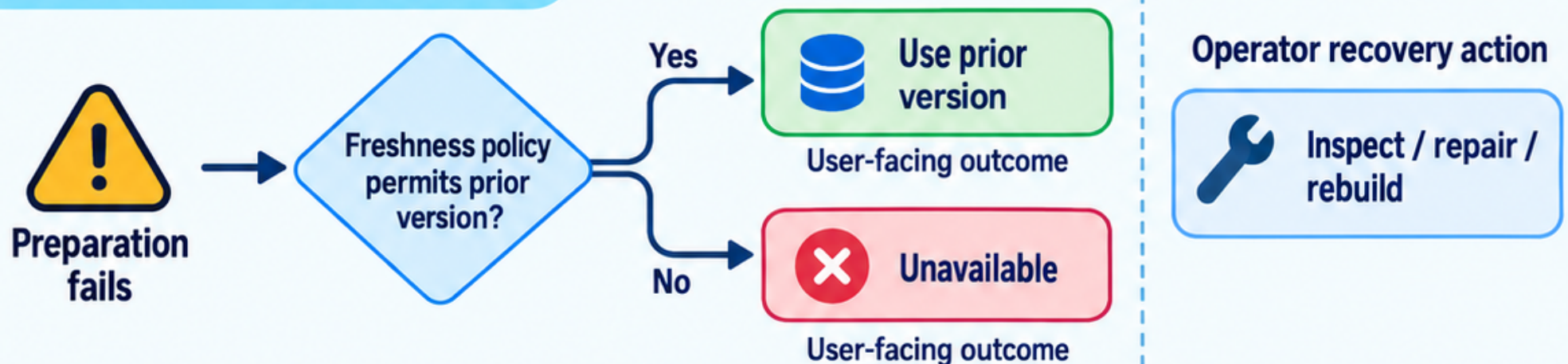
If preparation fails and the freshness policy permits it, keep the previous version. Otherwise mark the result unavailable.

Serving fails: return a defined error, fallback, or handoff.

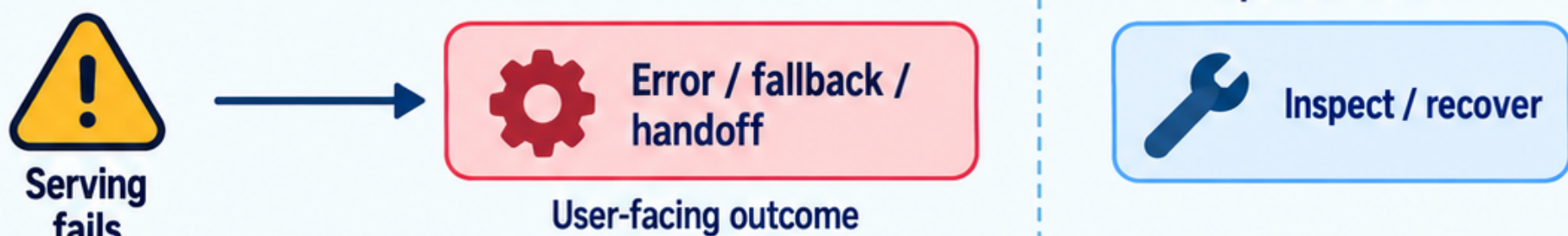
Record the affected version and failure stage. The user response and the operator's recovery action are separate design decisions.



PREPARATION FAILURE



SERVING FAILURE



MEASURE THE CONNECTION, NOT JUST THE BOXES

For the policy assistant, measure update delay and answer latency separately.
For the forecast, measure data age and batch completion separately.
Then check the output quality. A fast answer from the wrong policy and an on-time forecast from broken inputs can both fail the product requirement.

Policy assistant



What to measure

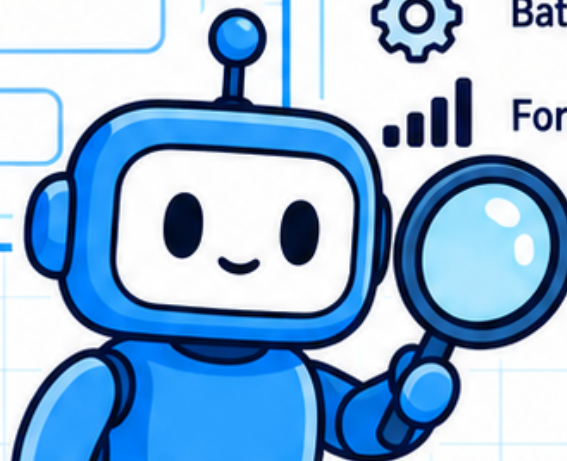
- Update delay
- Answer latency
- Answer quality

Demand forecast



What to measure

- Data age
- Batch completion
- Forecast quality



YOUR TURN

Draw both paths for a policy assistant or a demand forecast.
Label the source, prepared artifact, serving step, owner, and version.
Add one preparation failure and one serving failure. State the response to each.

PREPARE

Source: _____

Preparation steps: _____

Prepared artifact: _____

Owner: _____

Version: _____



Preparation failure + response: _____

Handoff

SERVE

Request or batch input: _____

Serving step: _____

Result: _____

Owner: _____

Artifact version: _____



Serving failure + response: _____



DAY 5: API and identity boundaries

A clear handoff makes the whole system easier to reason about.