

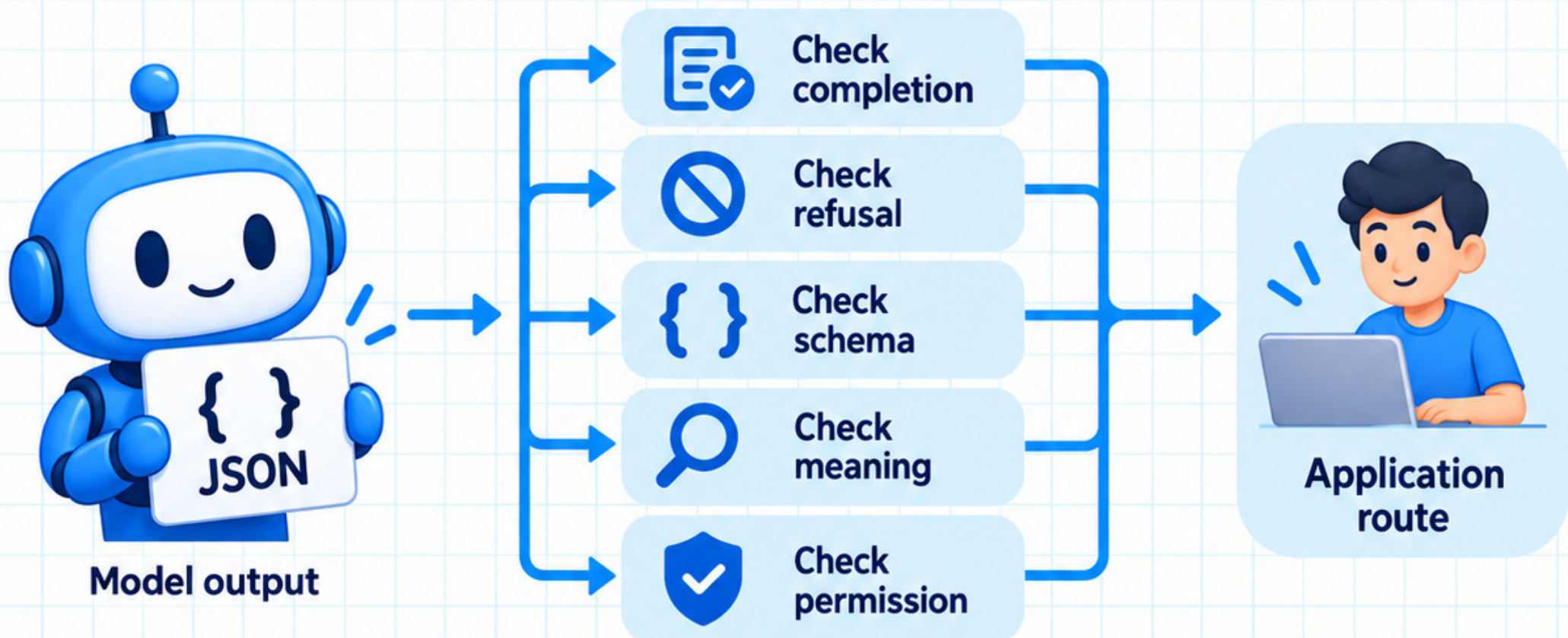
VALID JSON IS ONLY THE START

DESIGN STRUCTURED OUTPUT HANDLING

A result can have the right shape and the wrong meaning.

Today: turn a model response into a defined application branch.

Check completion, refusal, schema, meaning, and permission separately.



START WITH THE CONSUMER



Illustrative task:
classify a support
message and extract
an order ID.



The application
uses the result to
choose the next
step.

THE APPLICATION CHOOSES THE NEXT STEP



Support message



Classification result



Application router



Define the allowed
fields and missing-data
behavior first.



A classification is not
an approval to read an
order or issue a refund.

WRITE A SMALL OUTPUT CONTRACT

Illustrative required fields:

intent:

order_status, refund_request, or unknown.

order_status

refund_request

unknown

order_id:

a string, or null when absent.

"12345"

null



No extra fields.

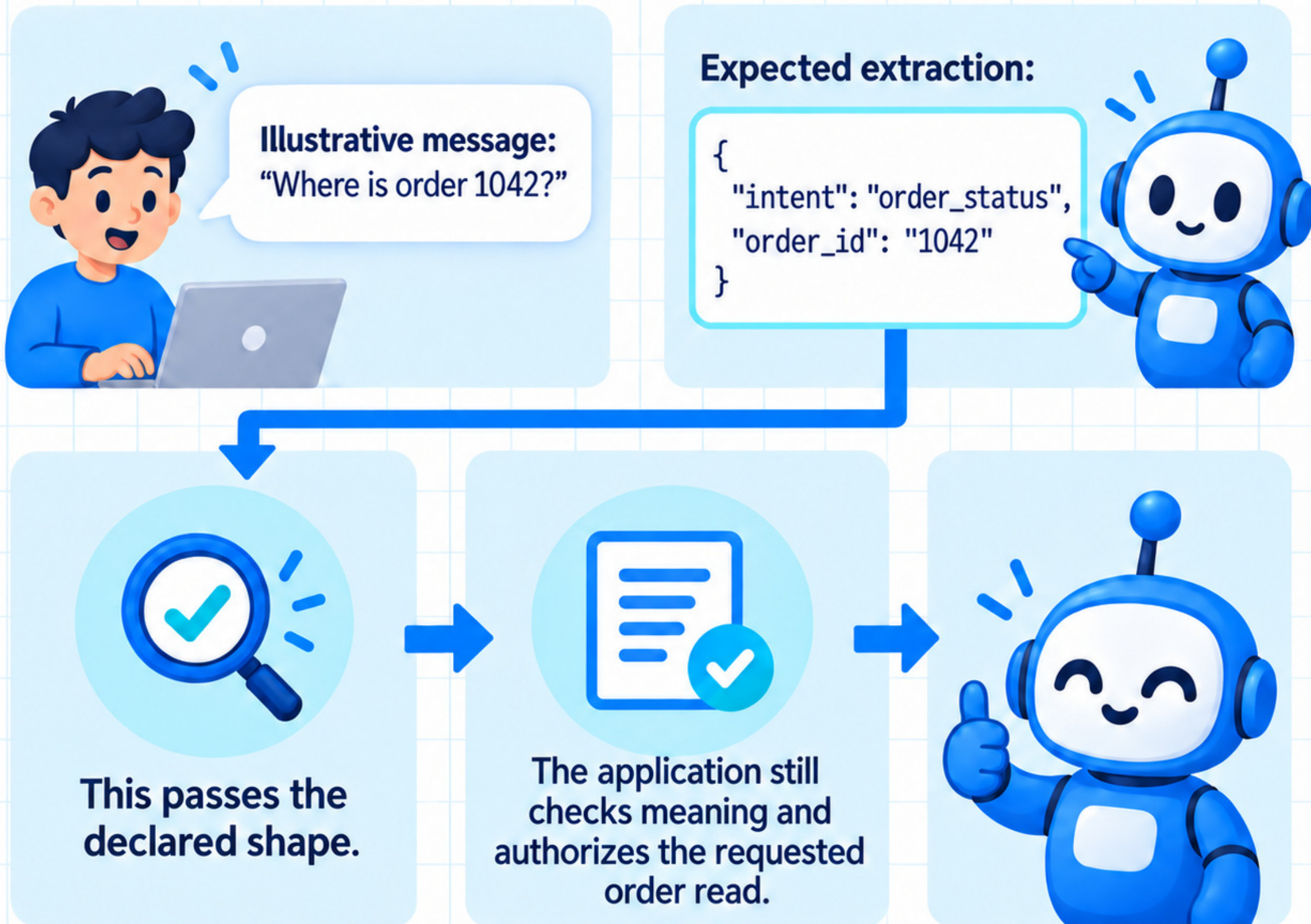


Keep identifiers as strings
so formatting is preserved.



Do not invent an ID to
satisfy the shape.

SHOW A VALID EXAMPLE



THREE LAYERS OF VALIDITY

SYNTAX:

can the text be
parsed as JSON?



SCHEMA:

are fields, types,
and values allowed?



MEANING:

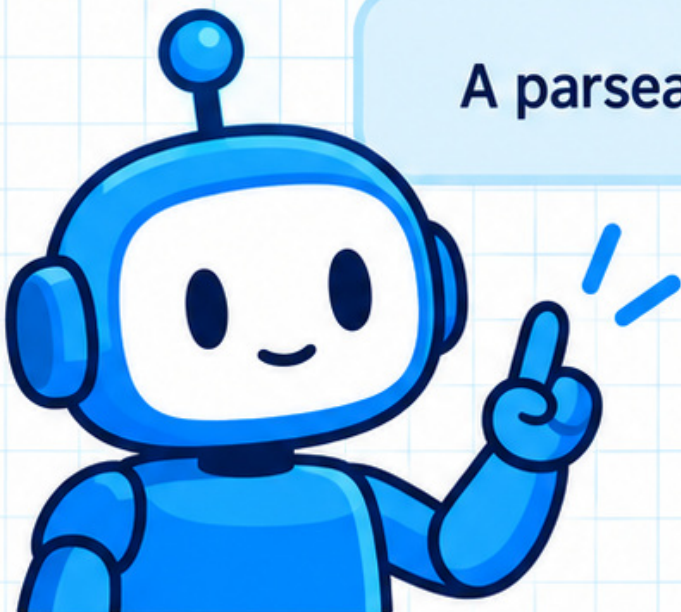
does the result
correctly represent
the input and task?



Authorization
is another
application
check after
these.



A parseable object can fail either later check.

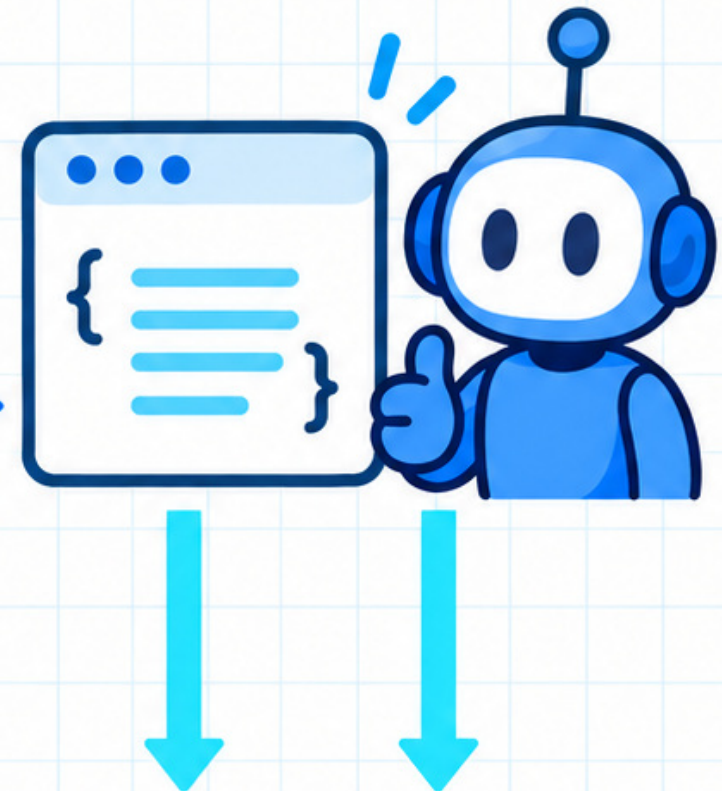


WHAT STRUCTURED GENERATION HELPS WITH

Supported provider features can constrain output to a declared schema.



Check the chosen model, endpoint, and supported schema subset.



They do not prove that extracted facts are correct.



They also do not replace handling refusals, incomplete responses, or API errors.



INSPECT THE RESPONSE ENVELOPE FIRST



Did the API request fail or time out?
Is the response complete?
Does it contain a refusal?
Is the expected output present?

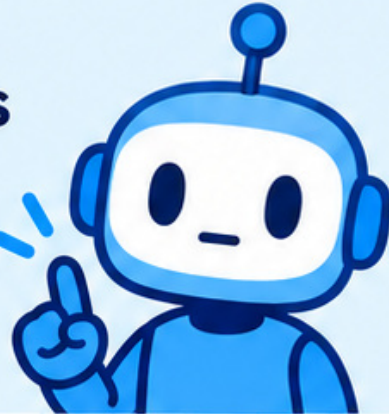


Do not treat every response as your business object.



GIVE REFUSAL ITS OWN ROUTE

A provider refusal is a response state, not your intent field.



Return the application's appropriate refusal message or route.



Record the category without exposing unnecessary sensitive text.



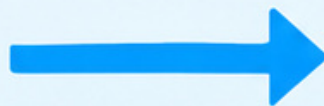
Do not keep rewriting the request to bypass the refusal.



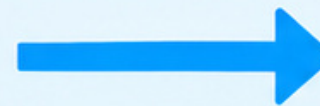
Refusal signal → defined refusal handler → user response; no automatic retry loop.



Refusal signal



Defined refusal handler



User response

INCOMPLETE IS NOT ALMOST VALID

A token limit, interruption, or connection failure may leave partial content.

Illustrative incomplete JSON

```
{  
  "intent": "order_status",  
  "order_id": "104"
```

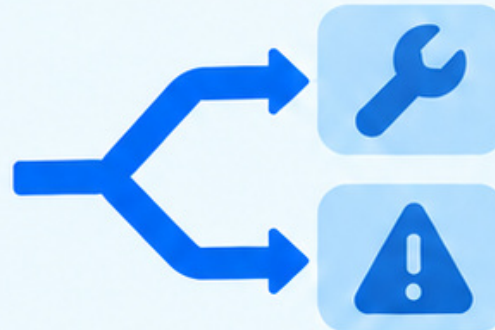
Do not execute downstream behavior from a JSON prefix.



Keep the result unaccepted.



Apply the declared bounded recovery or unavailable response.



Completion must be explicit.



VALIDATE AGAIN AT YOUR BOUNDARY

Pydantic can check the application's fields, types, enums, and configured constraints.



Application schema validator

- fields ✓
- types ✓
- enums ✓
- configured constraints ✓



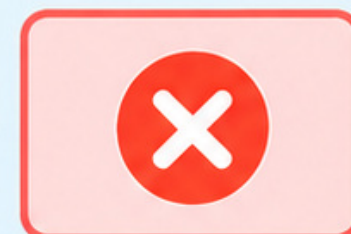
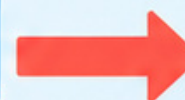
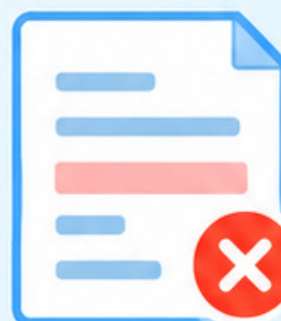
Choose strictness and extra-field behavior deliberately.

strictness

extra-field behavior



Reject unexpected fields rather than silently trusting them.



Validation must target the schema version the consumer understands.



Application schema validator

Version



MEANING NEEDS ITS OWN CHECKS



The object can pass the schema and still be wrong.



Compare extracted facts with the supplied input when possible.



Use task evaluations for ambiguous intent decisions.

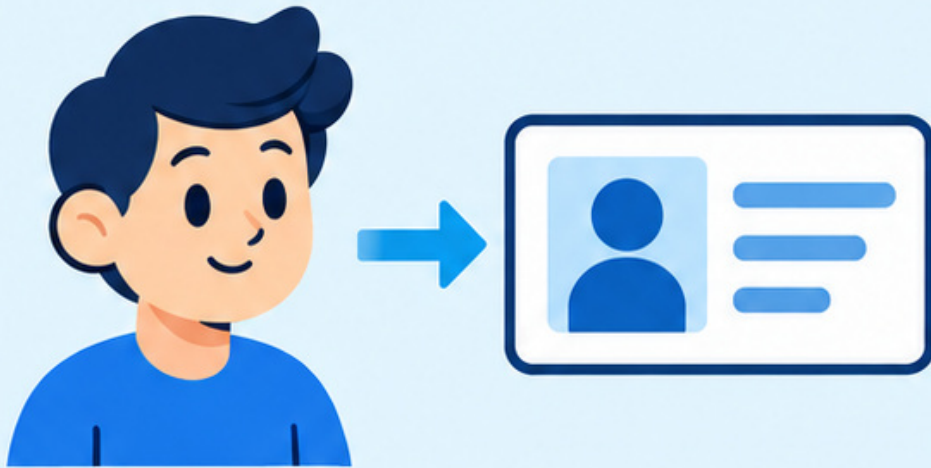


A confidence-like number is not proof of correctness.



AUTHORIZE AFTER EXTRACTION

Resolve the caller's identity in the application.



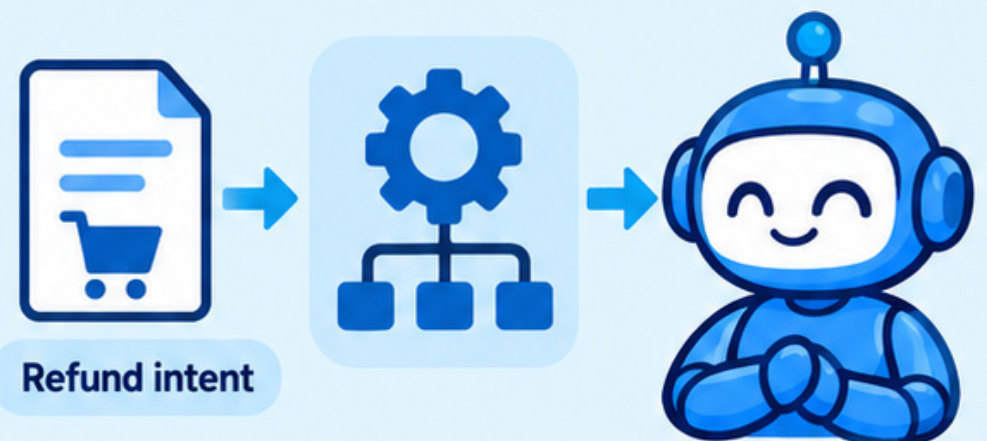
Check that the caller may access the extracted order.



A valid ID and a correct intent do not grant permission.



Refund intent only selects a workflow; business rules and any required approval govern execution.



NULL SHOULD HAVE A MEANING



Illustrative message:

Where is my order?



Result:

```
intent = order_status;  
order_id = null.
```



SEPARATE CASE: UNKNOWN INTENT



Unknown intent can request clarification.



The application asks for the missing identifier or uses an authorized lookup flow.

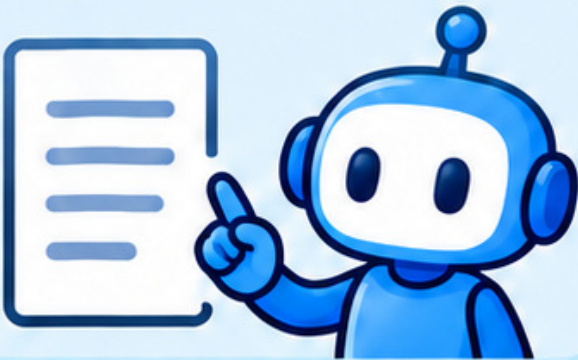


Missing data must not silently become a guessed value.

BOUND RECOVERY WORK

Illustrative policy: at most one repair attempt for recoverable format failure, within the request budget.

Keep the original input and contract.



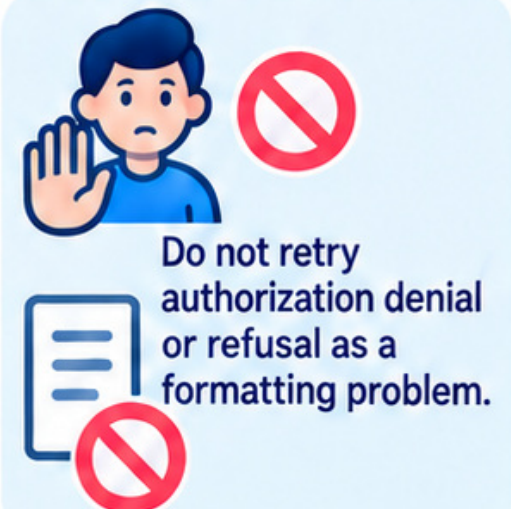
Revalidate the complete new result.



If it still fails, return the defined fallback.



ONE BOUNDED REPAIR, THEN REVALIDATE



TEST EVERY APPLICATION BRANCH



Create fixtures for:

- Valid and supported output.
- Wrong field type or extra field.
- Correct shape but wrong extracted ID.
- Refusal, incomplete output, and transport error.
- Missing ID and denied access.

Each fixture needs a specific expected route.

SCHEMA BRANCHES



Valid shape.



Wrong field type
or extra field.

MEANING BRANCHES



Correct shape but
wrong extracted ID.

ENVELOPE BRANCHES



Refusal, incomplete
output, and transport
error.

WORKFLOW BRANCHES



Missing ID and
denied access.

MEASURE THE WHOLE ROUTE

Track schema-pass rate and task correctness separately.



schema-pass rate



task correctness

Report refusals, incomplete results, failures, repair attempts, and fallback rate.



refusals



incomplete results



failures



repair attempts



fallback rate

Include all eligible attempts in declared denominators.



all eligible attempts

Check latency and cost for the full request, including retries.



A parser success rate is not task success.



parser success rate



task success



YOUR TURN: CHOOSE THE BRANCH



Illustrative cases:

- A. JSON has intent = order_status and order_id = 1042 as a number.
- B. Valid strings name another customer's order.
- C. The response ends halfway through JSON.
- D. order_id is null and intent is order_status.

Assume strict string validation. What happens next?

- A** JSON has intent = order_status and order_id = 1042 as a number.

```
{  
  "intent": "order_status",  
  "order_id": 1042  
}
```



- B** Valid strings name another customer's order.

```
{  
  "intent": "order_status",  
  "order_id": "1050"  
}
```



- C** The response ends halfway through JSON.

```
{  
  "intent": "order_status",  
  "order_id": "1042"
```



- D** order_id is null and intent is order_status.

```
{  
  "intent": "order_status",  
  "order_id": null  
}
```



ACCEPT RESULTS IN STAGES

**A**

A → schema failure; use the declared recovery route.

**B**

B → authorization denial; no protected fields.

**C**

C → incomplete-output handler; no downstream action.

**D**

D → ask for the missing ID or use authorized lookup.



Shape, meaning,
and permission
are different checks.



Next: choose
models with
a task scorecard.