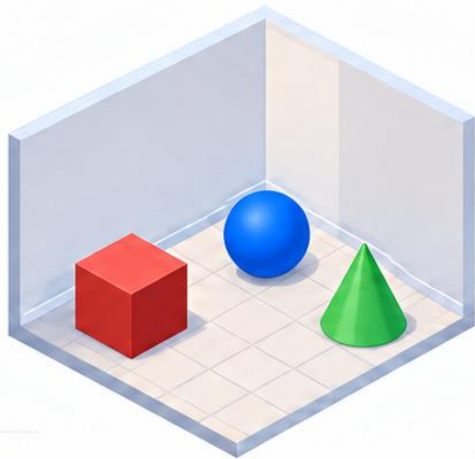
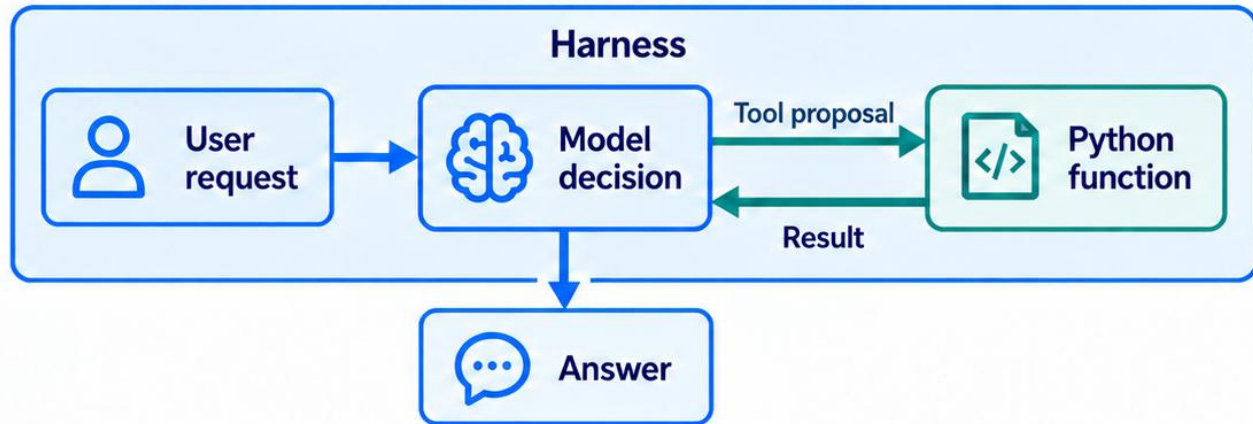


Your first Agent Playground agent



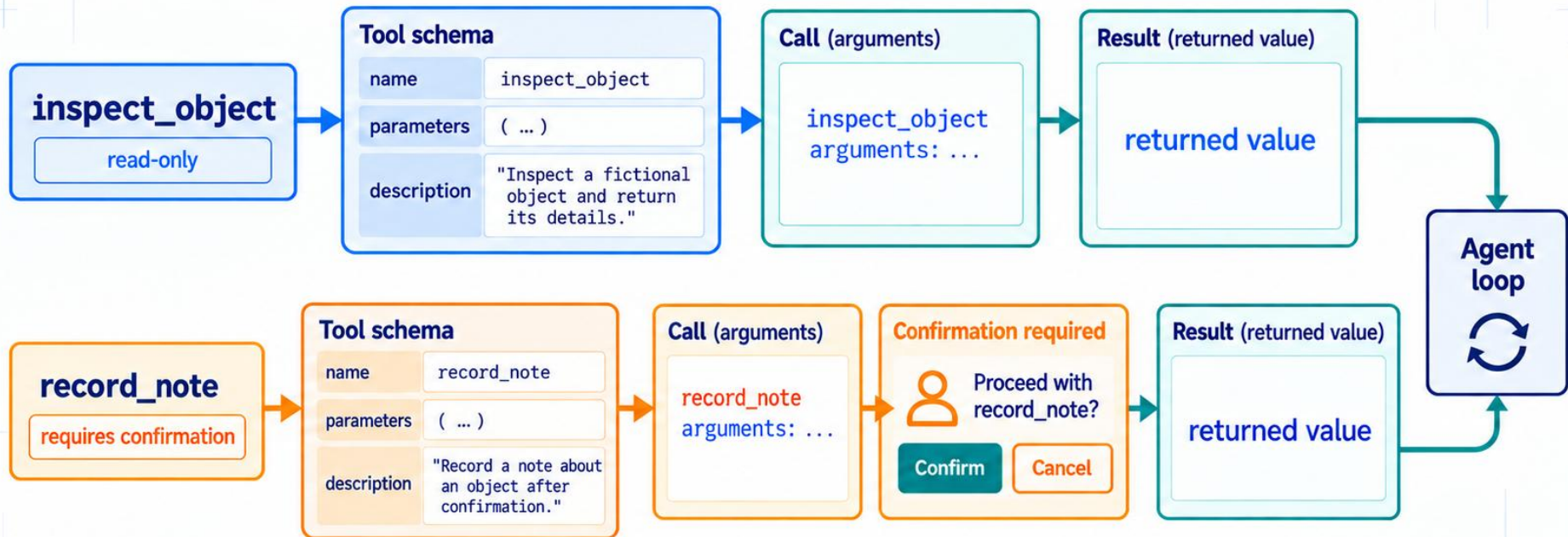
Fictional room catalogue
Teaching fixture

```
Agent(ModelPolicy(model))
```



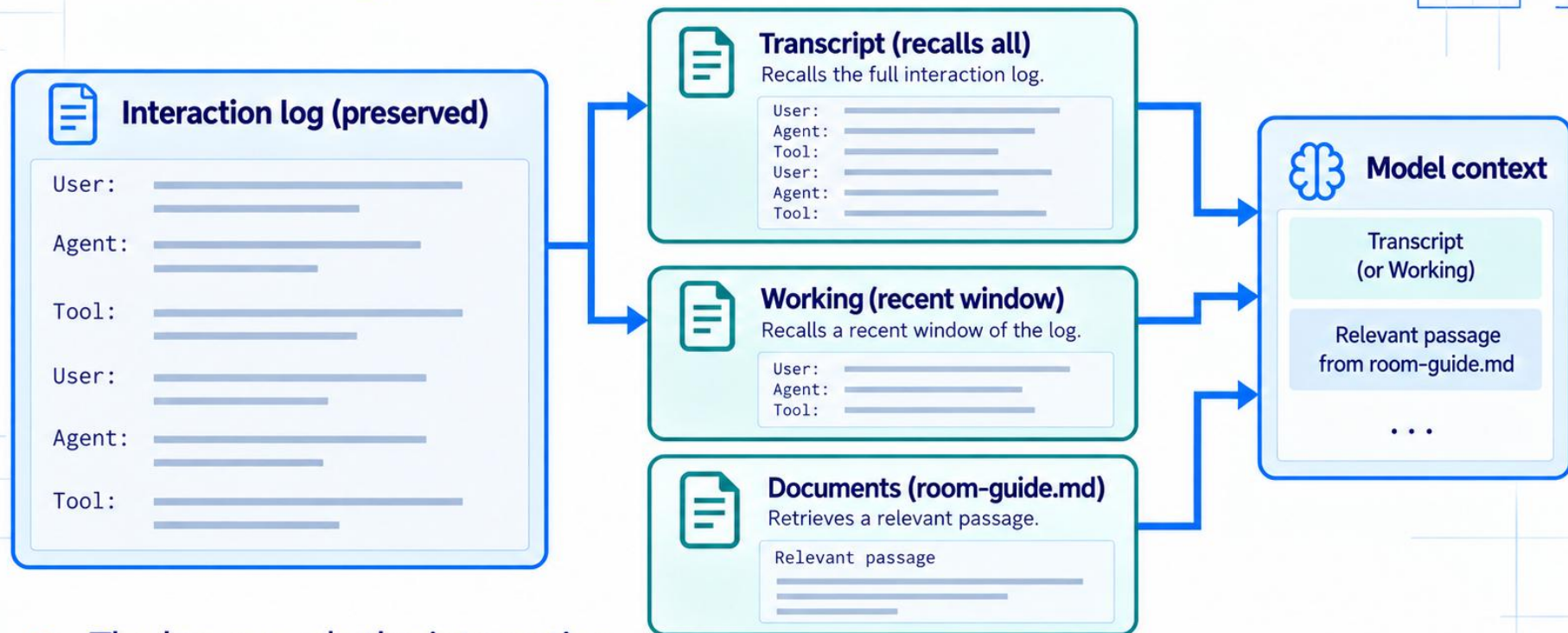
- Create the agent directly in the notebook.
- The LLM chooses calls and answers. The harness runs the loop.

Functions become tools



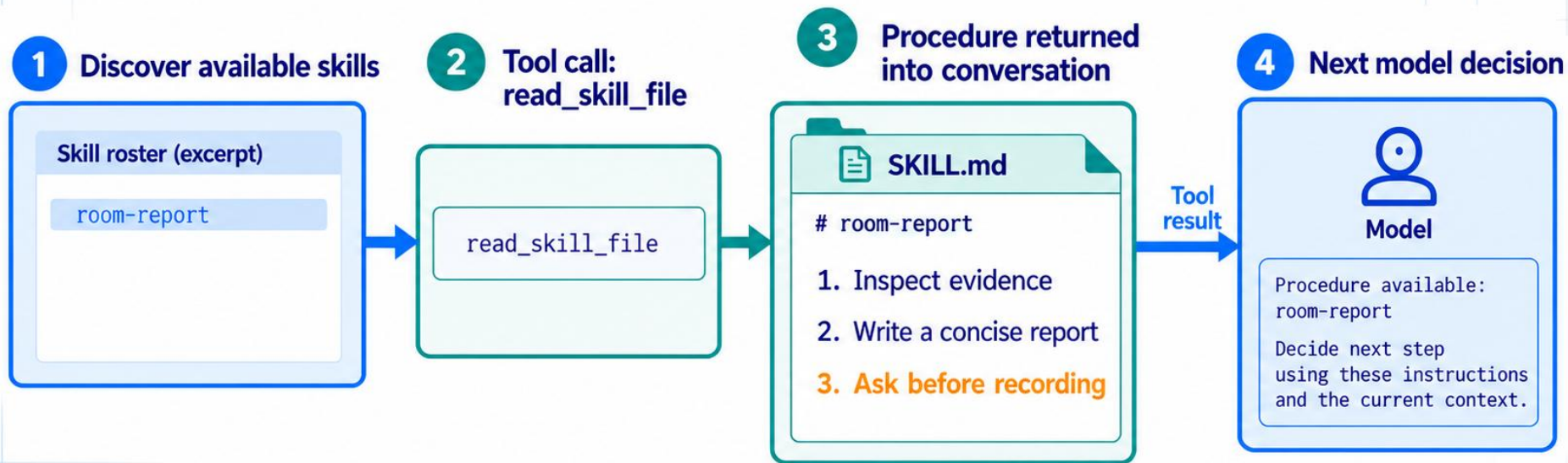
- A signature and docstring describe a callable.
- Inspect the call, its arguments and the returned result.

Memory supplies context



- The log records the interaction.
- Memory selects what reaches the next decision.

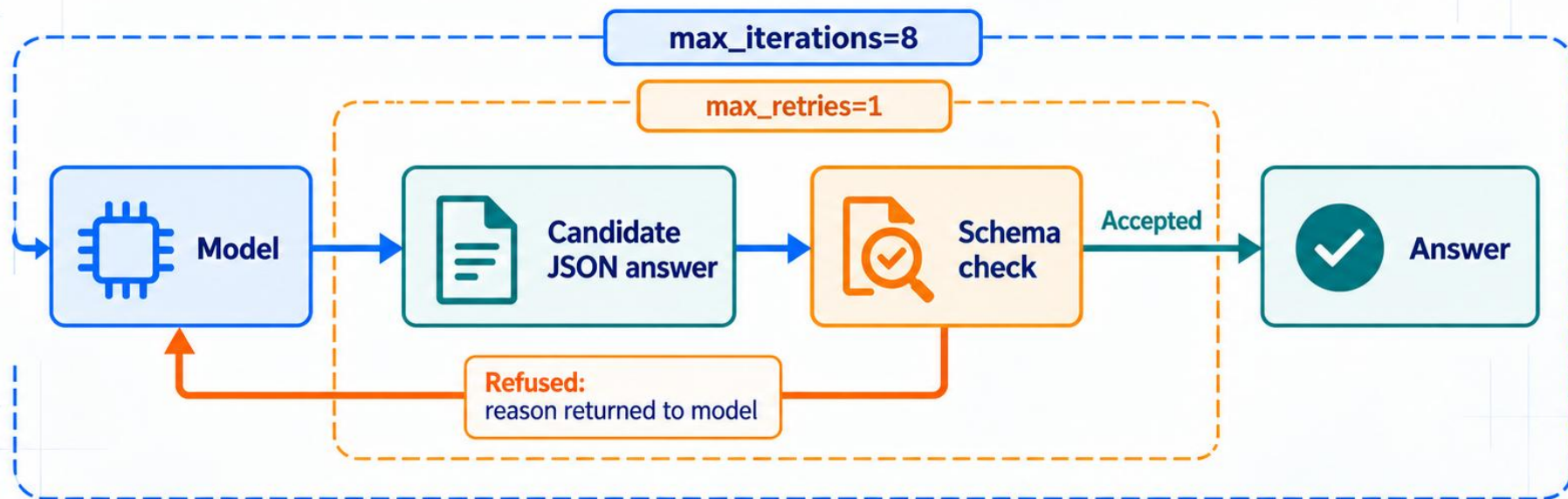
Skills supply instructions on demand



- The roster names an available procedure.

- Reading the skill brings its instructions into the conversation.

Verification and limits



- Schema checks the answer's structure.

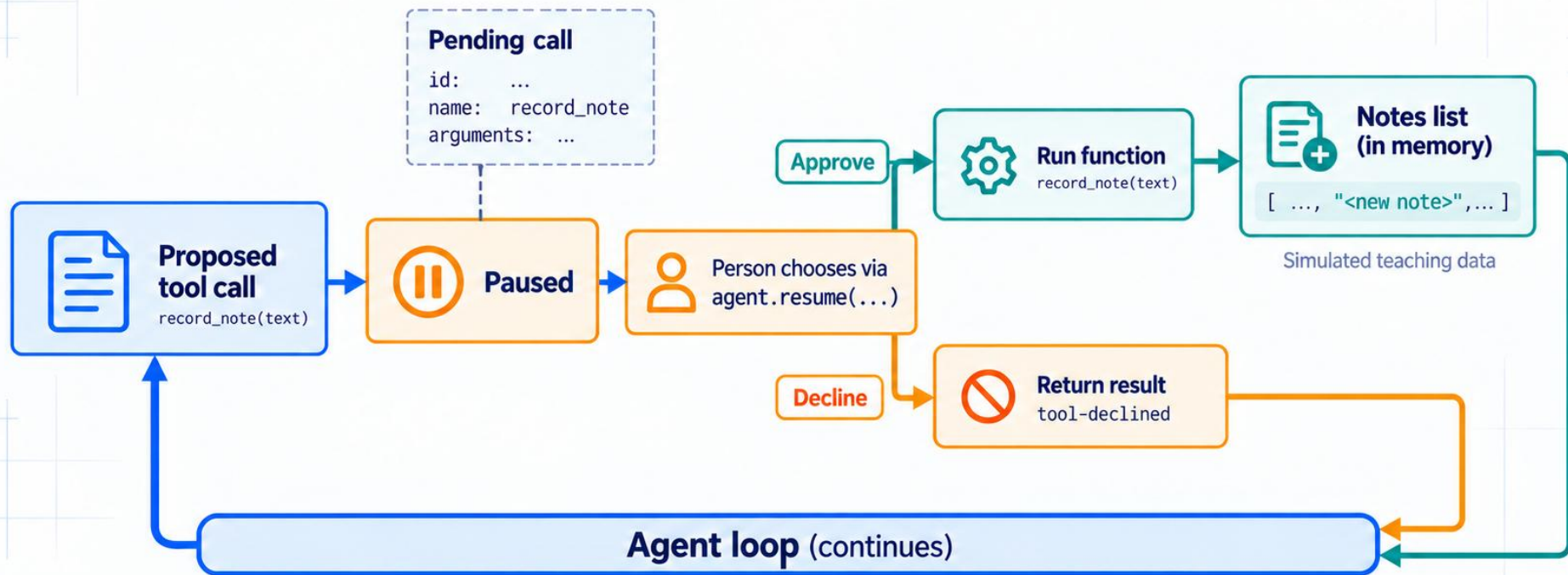
- Refusals can trigger a bounded retry.

- The turn limit bounds the loop.



Structure alone does not prove the facts.

A note waits for approval

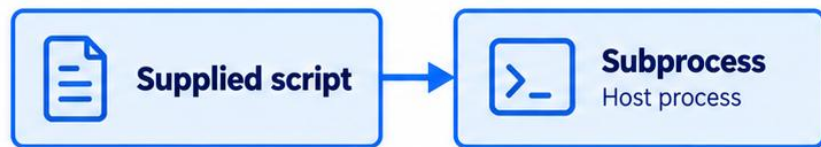


- Inspect the pending call before choosing.
- Approve or decline through `agent.resume(...)`.
- Check the notes list to see whether the effect occurred.

Further harness connections



Supplied script and harness



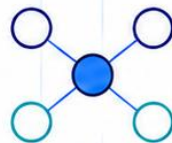
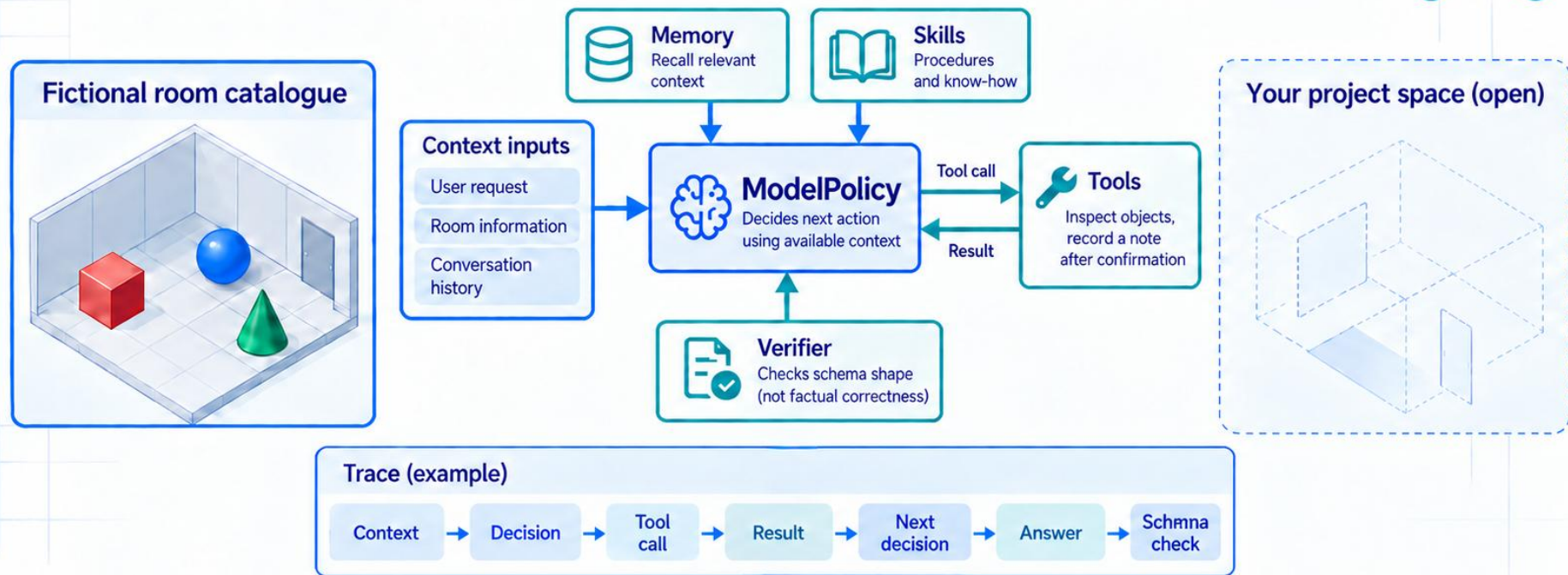
Delegation with the same harness



- Reward scores steps; the harness collects an Episode.
- QLearning.learn updates a table; ModelPolicy weights stay unchanged.
- Supplied scripts and delegation use the same harness.

ModelPolicy weights stay unchanged.

A small agent you can explain



- Your work: one small tool and an explicit agent configuration.
- Your evidence: a short **trace** and one explained outcome.
- Original project domains and integration remain open.