

Ourbit Vendor-Neutral AI - Day 14: Design a Vendor-Neutral AI Workflow

Day 14 - Design a Vendor-Neutral AI Workflow

Programme: Vendor-Neutral AI - 15-Day Certification Programme

Purpose: Integrate prompting, context, retrieval, tools, evaluation and human controls into a portable solution architecture.

Learning outcomes

- Translate a problem into requirements and success measures.
- Decide whether prompting, retrieval, tools or automation are actually required.
- Design evidence, permission, validation and human-control boundaries.
- Separate portable business logic from vendor-specific implementation.
- Plan fallback and migration paths.

1. Architecture starts with requirements

Define the user, problem, inputs, desired outcome, unacceptable failures and success measures before selecting a model or platform. This prevents a product's current feature set from becoming the architecture by accident.

2. Use the simplest sufficient design

Some tasks need only a clear instruction. Others need controlled retrieval because answers must come from approved knowledge. Tools are needed only when the system must interact with external capabilities. Automation is justified when repeated workflow steps can be made reliable enough to save meaningful effort. Each added layer creates additional failure modes and controls.

3. Define the control plane

Specify authoritative sources, permitted data, tool permissions, validation rules, human approval gates, logs, fallback behavior and stop conditions. These controls are part of the solution, not documentation added after it works.

4. Separate core logic from vendor adapters

Where practical, keep programme rules, business rules, data schemas, prompts/instructions, evaluation criteria and state definitions under your control. A provider-specific model call, storage connector or user interface can then be treated as an adapter. Portability is not the absence of vendors; it is avoiding unnecessary ownership of your core logic by one vendor.

5. Design for failure and change

Define what happens when the model is unavailable, retrieval returns nothing, a tool fails, evidence conflicts or a provider must be replaced. A portable system should degrade safely and preserve enough state and documentation for another implementation to continue.

Worked example - knowledge assistant

An organization wants an assistant for approved internal procedures. The portable core defines document identity, approval status, retrieval filters, answer rules, source traceability, access policy and evaluation tests. One model provider performs generation today. If that provider changes, the controlled knowledge, rules and tests remain reusable.

Guided practice

Create a one-page architecture for an AI knowledge workflow. Include user/task, inputs, authoritative sources, retrieval if required, generation step, tools if required, validation, human gate, output, audit record, fallback and one portability boundary.

Knowledge check

- Why define requirements before choosing a vendor?
- When is retrieval justified?
- What belongs in the control plane?
- What should remain portable where feasible?
- Why design fallback before launch?

Applied evidence task

Design an AI-assisted service workflow that could migrate to a different model provider. Identify which components are Ourbit-controlled, which are replaceable adapters, how evidence is verified, what actions require approval and how the workflow behaves during provider failure.

Day 14 takeaway

Vendor neutrality is an architecture discipline: own the requirements, logic, data meaning, controls and tests; use external providers as replaceable implementation components where practical.

Assessment boundary

This learning evidence belongs only to vendor-neutral-ai-15-day.

Learning comes before credentials. Completing this reading alone does not issue a certificate. Programme assessment and practical requirements apply.