

MARKETING BROCHURE • 20 PAGES

The Dweve technical platform

Follow a human question through governed knowledge, cognition, action, compute, deployment, and proof.

VISUAL BRIEFING

PRODUCT LEADERS, ARCHITECTS, AND EVALUATION TEAMS

ENGLISH



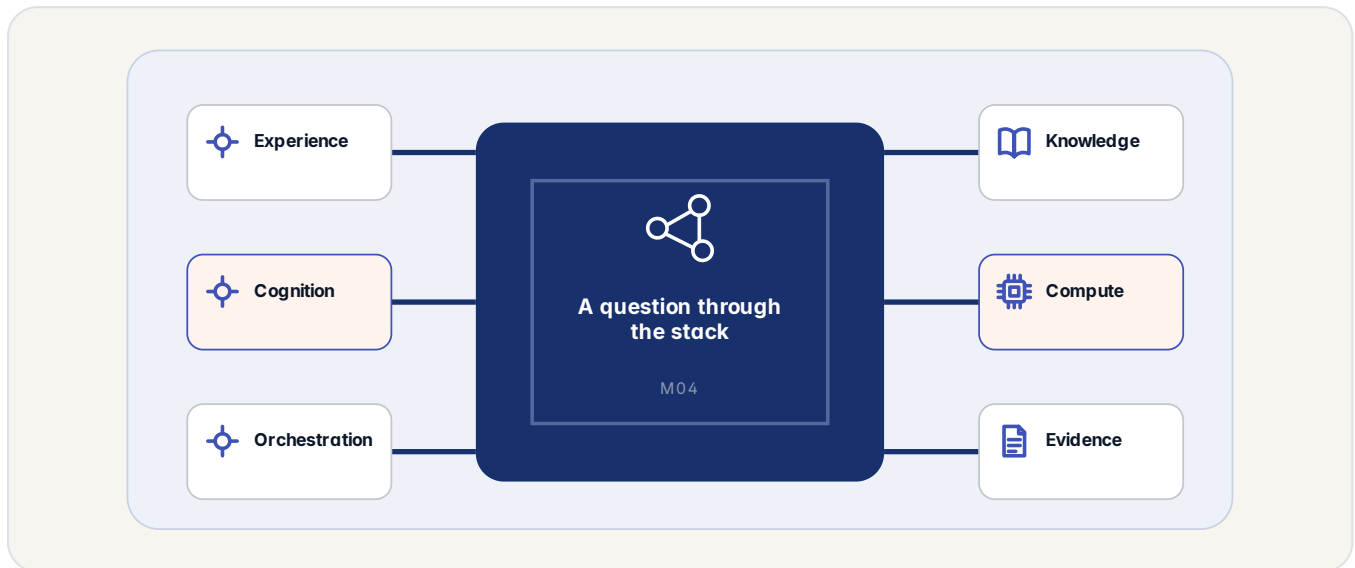
See the technical platform as a flow of responsibilities, boundaries, and artefacts.

M04

THE PROPOSITION

One connected work path

Deployment uses three operating postures: managed participation in Dweve's public Mesh, a licensed platform on hardware you control, or fully air-gapped operation with no outbound network. The operating responsibility changes without inventing a different product architecture.



How to read this visual: follow a question through the stack through the proposition.

MECHANISM

Read the promise against the operating reality

Keep the Fabric workflow familiar while moving from the Business Workspace on the public Mesh to licensed customer operation or a physically isolated perimeter. Location, product rights, network path, and operating responsibility change; sources, review, ownership, and approval remain recognizable.

Choose managed Fabric access, a licensed deployment on hardware you control, or a fully air-gapped installation with no outbound dependency. The access boundary changes while the platform architecture remains portable, so moving to direct licensed operation does not require a new application architecture.

- Experience
- Cognition
- Orchestration
- Knowledge

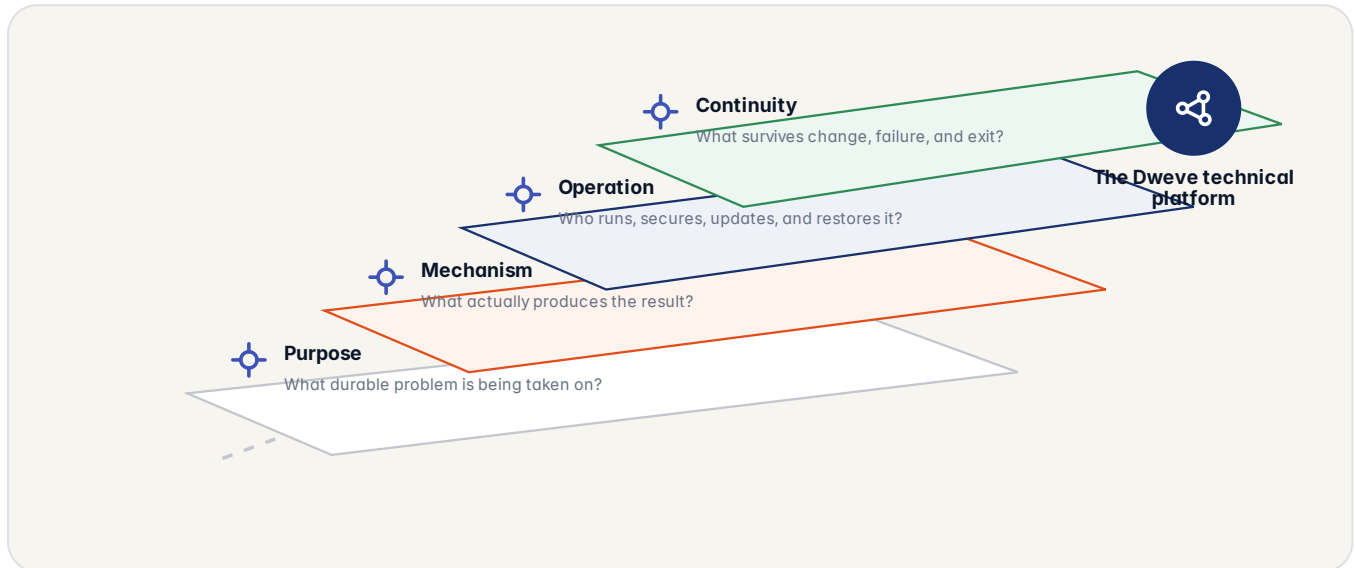
QUESTION TO CARRY

Which part of this proposition changes the outcome people receive today?

DEEPER VIEW

Read the proposition through four connected lenses

A serious decision connects purpose, mechanism, operation, and long-term responsibility.



How to read this visual: follow a question through the stack through deeper view.

DECISION

Connect the mechanism to the responsibility

Dweve connects the work surface, company record, coordination, development, cognition, knowledge, execution, and placement as one operating system. The eight commercial products carry distinct responsibilities, while Kera is the graph-native toolchain that prepares supported target execution beneath them. The work identity, control state, and evidence continue across each handoff.

Dweve maintains the agreed product, interfaces, service operation, and support boundary; the customer controls data, access, policy, reviewers, approvals, and deployment choice. The responsibility map is agreed before live work begins and stays with the operating record. Exportable work, sources, decisions, and evidence support transition and exit.

- Purpose
- Mechanism
- Operation
- Continuity

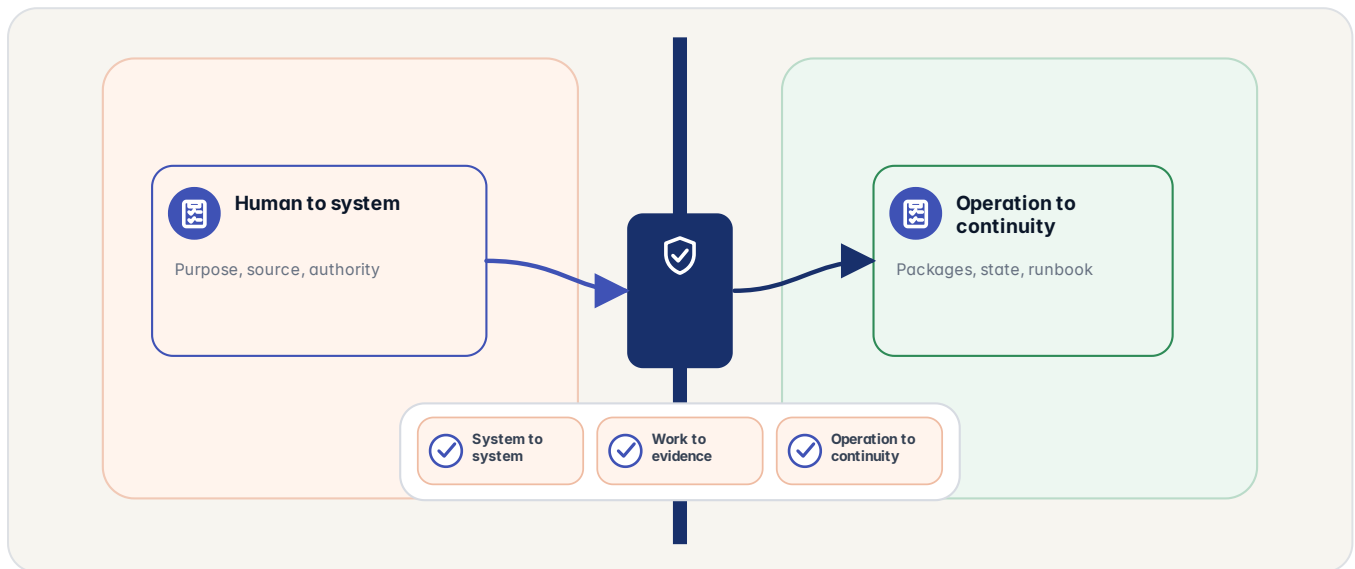
QUESTION TO CARRY

Can purpose, mechanism, operation, and continuity be reviewed together?

BOUNDARY MAP

Interfaces are promises between responsibilities

A clean boundary makes change, testing, replacement, and accountability easier to reason about.



How to read this visual: follow a question through the stack through boundary map.

MECHANISM

Inspect what must survive the hand-off

Fabric makes the workspace object graph the durable boundary. Goals, sources, participants, tools, approvals, events, and records belong to the work, not a provider conversation. Models and services connect through typed adapters, so a backend can change without moving state or losing execution history.

Typed inputs, versioned artifacts, stable formats, directed dependencies, and conformance checks define each boundary. Product surfaces depend on those contracts, not hidden shared state. A provider, engine, or service can be replaced one layer at a time and tested against observable behavior.

- Human to system
- System to system
- Work to evidence
- Operation to continuity

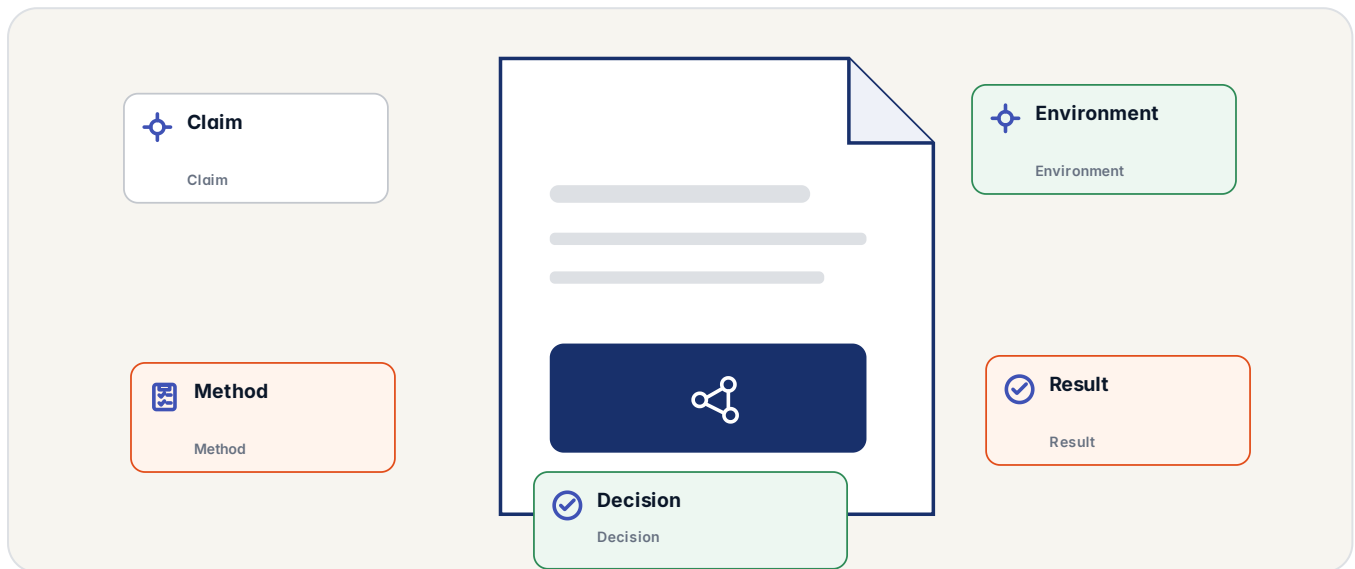
QUESTION TO CARRY

Which interface carries meaning, authority, failure, and evidence across the boundary?

EVIDENCE DESIGN

A claim becomes useful when the reviewer can test it

The brochure makes the claim visible. The proof still runs against the selected workload, environment, and acceptance criteria.



How to read this visual: follow a question through the stack through evidence design.

DECISION

Turn the claim into something reviewable

Three formats, Q8.8/Q16.16/Q31.32, selected per kernel for bit-identical results across CPU, ARM, and RISC-V. No denormals, no NaN surprises, no machine-dependent rounding. Replay a six-month-old inference and you get the same answer to the last bit. LRAT and Alethe certificates attach to every emit. The audit format is open: verify on your own tooling, keep the evidence in your own store.

Charter carries authority and policy beside the operating graph. Lattice evaluates explicit rules against supplied state and returns the applied path. AION checks deductions against supplied premises and can emit portable proof material. This does not establish source truth, model quality, fairness, or legal compliance; those require separate evidence and judgement.

- Claim
- Method
- Environment
- Result

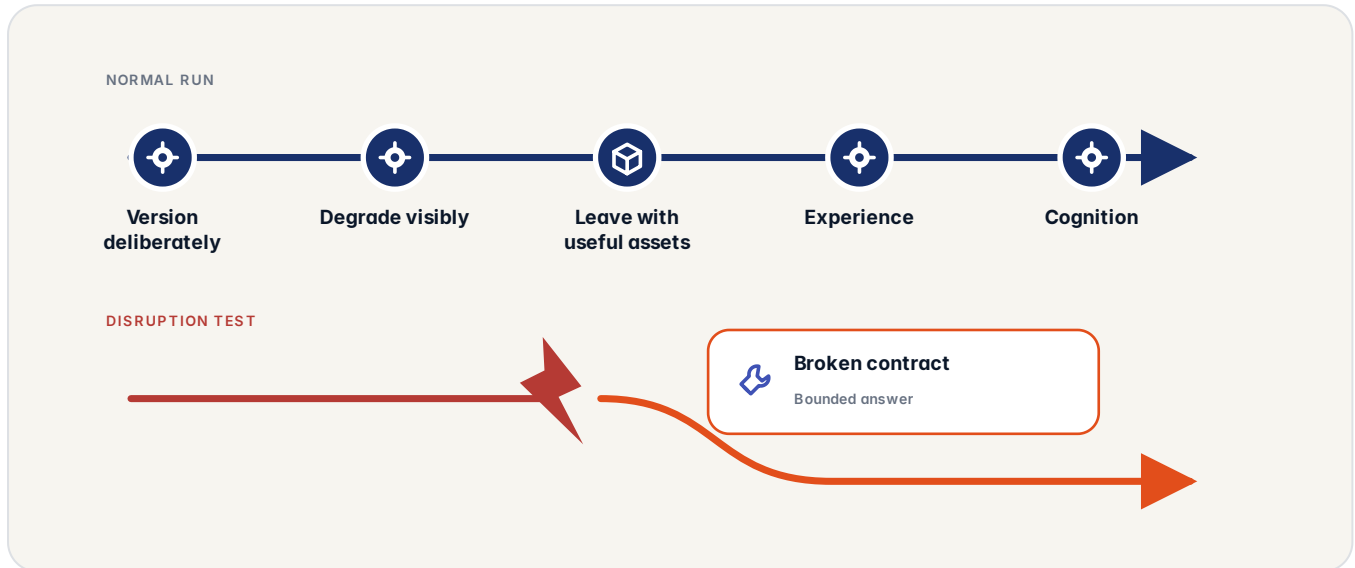
QUESTION TO CARRY

What would a reviewer need to reproduce or challenge the result?

CONTINUITY

Long-term fit includes change, failure, and exit

Evaluate the normal day and the difficult day before increasing organisational reliance.



How to read this visual: follow a question through the stack through continuity.

MECHANISM

Test the difficult day as well as the normal day

The visible open projects matter, but they are not the whole story. The wider engineering reality also includes the 60-crate dweve-* workspace carrying runtime, LLM, memory, storage, planning, evaluation, training, protocol, transport, metrics, security, and device support components. Read it like a core sample: each layer is a domain, each tile a project you can open.

Change a time, remove a wrong assumption, or add a missing preference. Fabric keeps your corrected result, the reason, and the earlier version. This thread or saved routine uses the better choice next time.

- Version deliberately
- Degrade visibly
- Leave with useful assets
- Experience

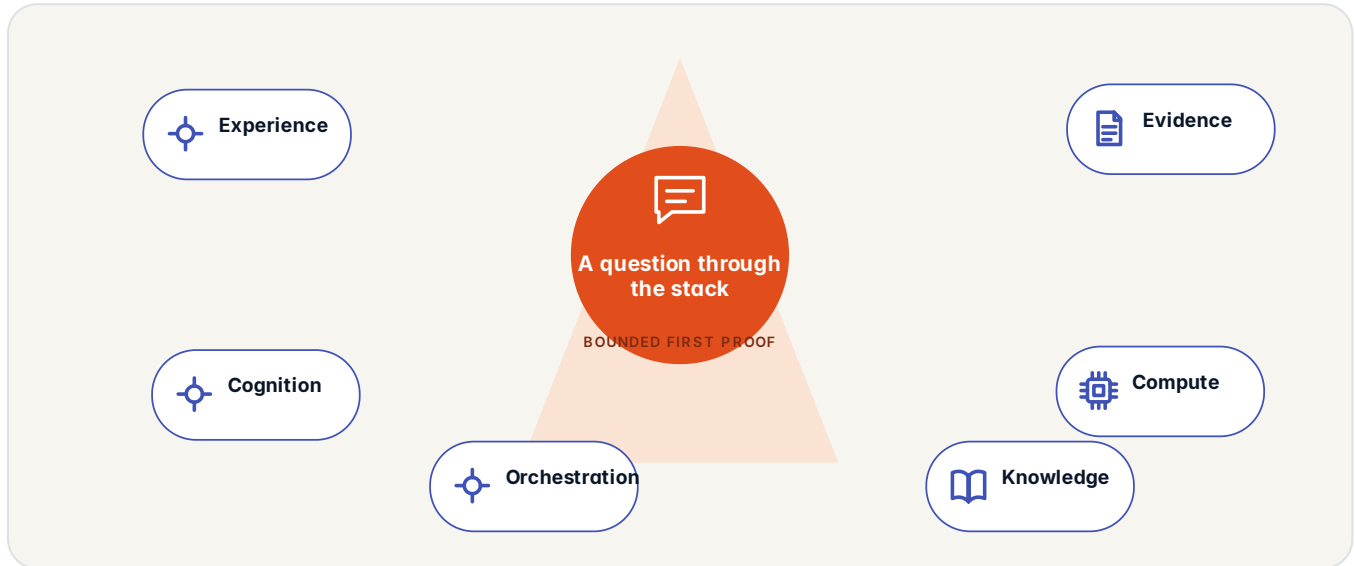
QUESTION TO CARRY

What must remain usable through change, failure, supplier loss, or exit?

USE AND FIT

Where this proposition can earn its place

Start with responsibilities that have a real owner, useful sources, and a consequence worth improving.



How to read this visual: follow a question through the stack through use and fit.

DECISION

Separate useful scope from attractive distraction

Choose real work with a named owner and a decision that matters. Baseline cycle time, queue age, repeat work, exceptions, and evidence effort; then run the same workflow in Fabric. Expand only after the agreed measures improve, reusing the same objects and contracts.

Every workflow that matters to the business ends with a decision and a paper trail. A chat tab can answer the question. Dweve finishes the work: owner assigned, sources attached, packet ready for the next reviewer.

- Experience
- Cognition
- Orchestration
- Knowledge

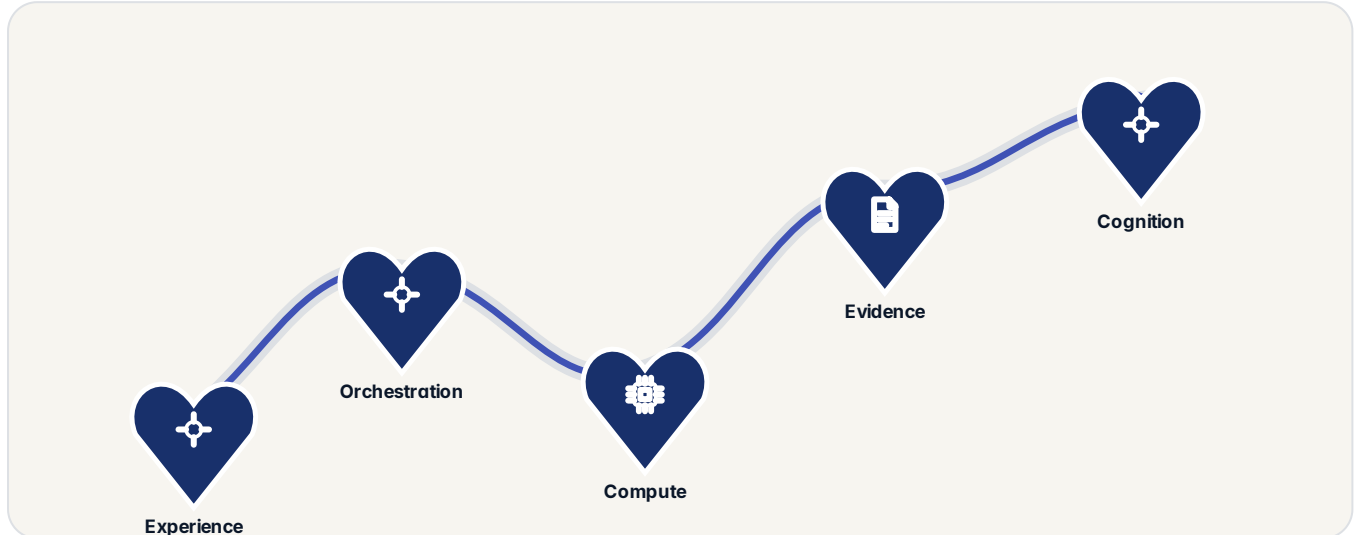
QUESTION TO CARRY

Where does this earn a place, and where should it deliberately not be used?

THE PATH

From first question to an informed decision

A focused engagement should make the next decision easier, not create a larger promise.



How to read this visual: follow a question through the stack through the path.

WHY THIS SHAPE WORKS

The workflow, operating boundary, evidence, and decision criteria are considered together from the start.

MECHANISM

Make every stage earn the next one

A useful system takes the messy part seriously: the documents, people, rules, and hand-offs that surround a decision. Dweve brings them into one review path so a team can move from a question to an answer that has an owner and a next step.

Fabric keeps intake, sources, checks, exceptions, ownership, and approval in one workflow. Teams see queue position, blockers, and the next responsible person. Each completed decision leaves a path that can be reused and measured.

- Experience
- Orchestration
- Compute
- Evidence

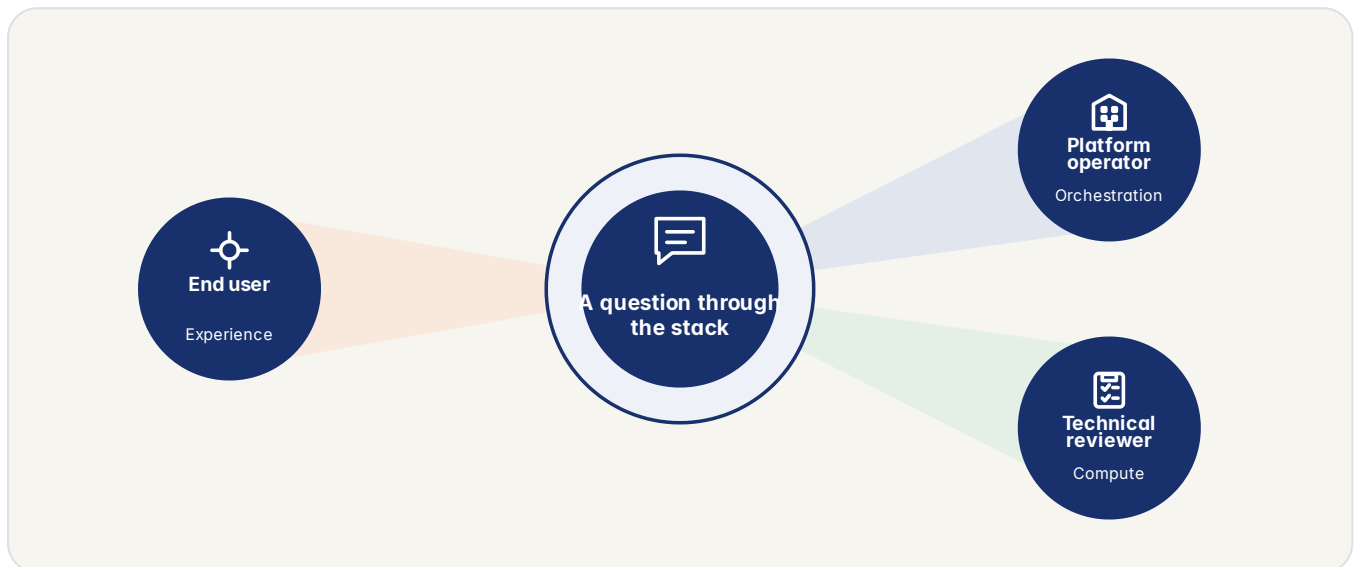
QUESTION TO CARRY

What evidence must exist before the scope is allowed to grow?

THE VALUE PICTURE

One outcome, seen from three responsibilities

Good AI work must make sense to the person using it, the team operating it, and the people reviewing it.



How to read this visual: follow a question through the stack through the value picture.

PRIMARY READER

Product leaders, architects, and evaluation teams

The first conversation.

DECISION SUPPORTED

See the technical platform as a flow of responsibilities, boundaries, and artefacts.

The next useful step.

DECISION

Read the same outcome from several responsibilities

Start in a European region, move the same workflow into your own environment, or isolate it completely. The operating model changes; the way people review and approve work does not.

The same intake, source, review, ownership, approval, and evidence path can operate through the Business Workspace, a private licensed deployment, or an isolated deployment. Data location, network access, and operating responsibility change at the perimeter. The decision record remains recognisable across all three.

- Experience
- Orchestration
- Compute
- Cognition

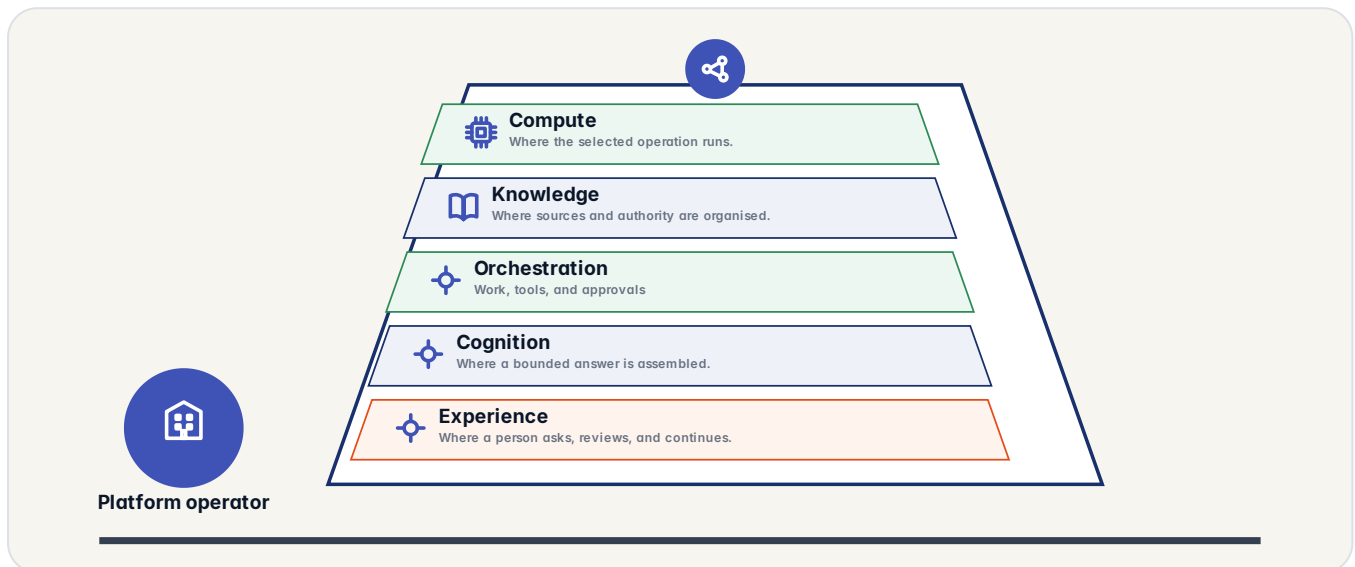
QUESTION TO CARRY

Can the same result satisfy its user, operator, and reviewer?

THE OPERATING MODEL

Control follows the work, from outcome to exit

A credible proposition connects the visible result to the less visible responsibilities underneath it.



How to read this visual: follow a question through the stack through the operating model.

MECHANISM

Name the controls behind the simple interface

Dweve maintains the product, contracts, and agreed support. The customer controls data, access, policy, reviewers, deployment, and exit. Exportable work and records keep operating history usable if the relationship changes.

When a policy, source, or workflow changes, Dweve follows the dependencies before the new version goes live. Affected work, required rechecks, and operating consequences are shown to the accountable owner. Rollout waits for explicit approval, while unaffected work keeps its recorded basis.

- Experience
- Cognition
- Orchestration
- Knowledge

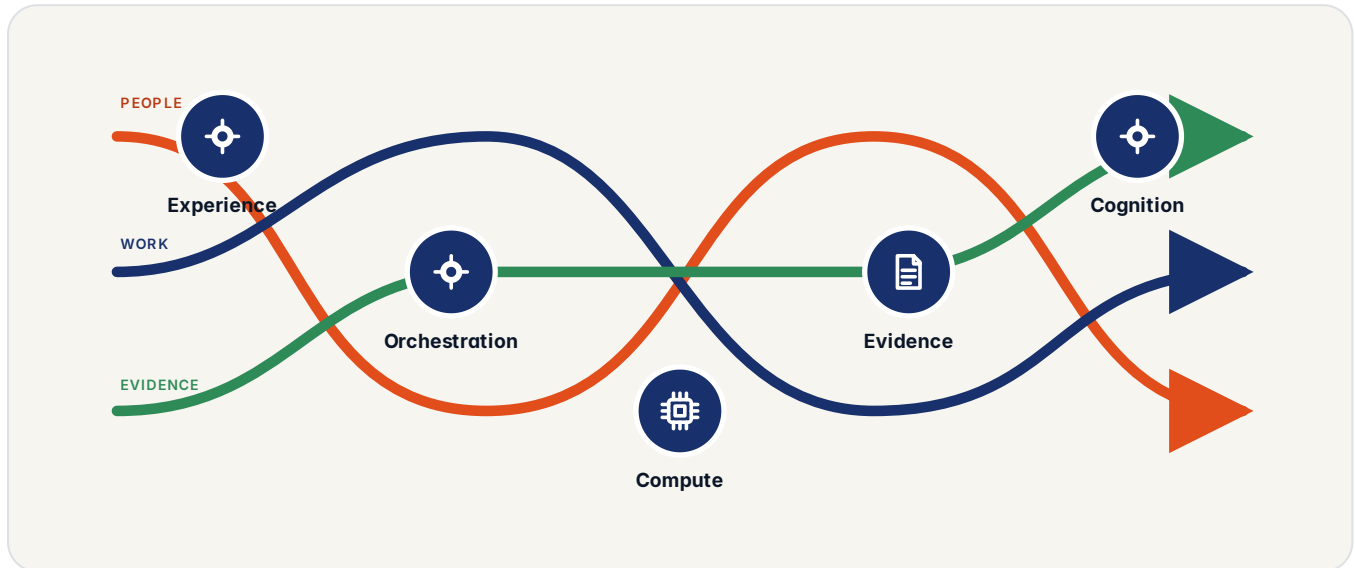
QUESTION TO CARRY

Who owns each control when the interface appears simple?

THE RESPONSIBILITY FLOW

People, platform, and evidence move together

The work becomes easier to govern when each stage leaves a clear hand-off.



How to read this visual: follow a question through the stack through the responsibility flow.

DECISION

Keep every hand-off attached to an owner

Dweve turns document-heavy work into a controlled flow from intake to approval. Sources, policy checks, exceptions, owners, actions, and evidence stay in one record, so operations can clear blockers and finance can measure the gain before scaling.

Set the baseline before the first live run, then compare the same workflow on the same definitions. Track cycle time, queue age, repeat work, exception handling, and evidence effort, with finance and operations agreeing how each measure is calculated. Expand only when the record shows an accepted improvement.

- Experience
- Orchestration
- Compute
- Evidence

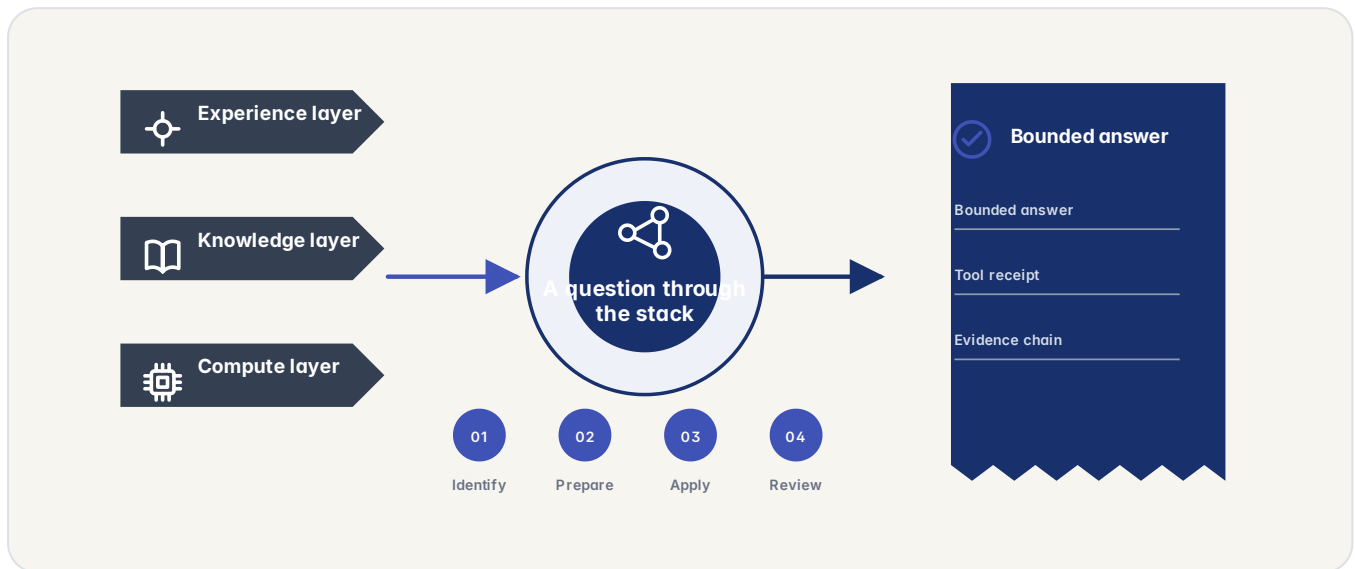
QUESTION TO CARRY

Is every hand-off visible, bounded, and assigned to an owner?

SOURCE TO RESULT

Every useful outcome begins with an authoritative basis

The work stays explainable when source identity, transformation, decision, and hand-off remain visible between systems.



How to read this visual: follow a question through the stack through source to result.

MECHANISM

Keep the basis visible as the work changes form

Spindle stores signed, versioned knowledge with source and transformation lineage. Corrections, withdrawals, policy changes, and conflicts are lifecycle events that identify affected atoms and exports. A signature establishes identity, integrity, and lineage, not truth; publication means that version passed its configured gates at that time.

Fabric creates the durable work object. Nexus or Aura coordinates governed actors; Loom resolves the cognitive graph; Spindle supplies versioned knowledge; Core and Kera execute on the selected target. The result returns to Fabric with route, source, policy, execution, and approval events in one trace.

- Identify
- Prepare
- Apply
- Review

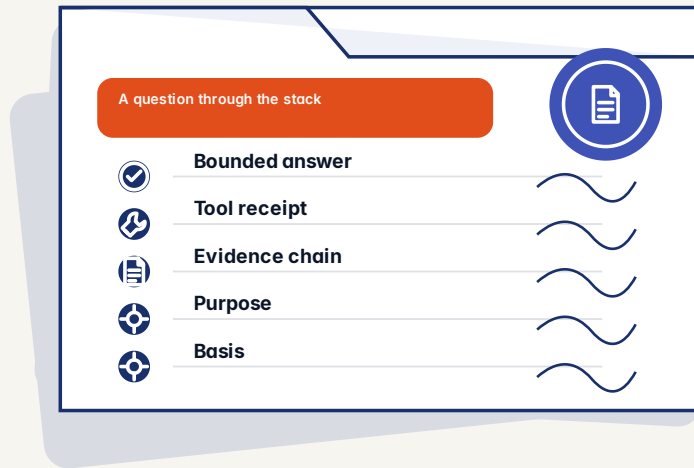
QUESTION TO CARRY

Can the result still be connected to the exact basis that shaped it?

EVIDENCE PACKET

The record should answer the next difficult question

Capture enough to explain the result, inspect authority, investigate failure, and make the next decision without rebuilding the story from memory.



How to read this visual: follow a question through the stack through evidence packet.

DECISION

Preserve the record needed by the next question

Dweve treats observability as part of the result. A useful system gives you the path, the constraints, and the artefact needed to inspect a decision without reconstructing the story from scattered logs.

A trace on every decision means a real question comes back with the source it used, the rule it applied, the owner who signed, and a sealed record you can replay. Here is the shape of that evidence. The example is illustrative until your own data is connected.

- Purpose
- Basis
- Authority
- Action

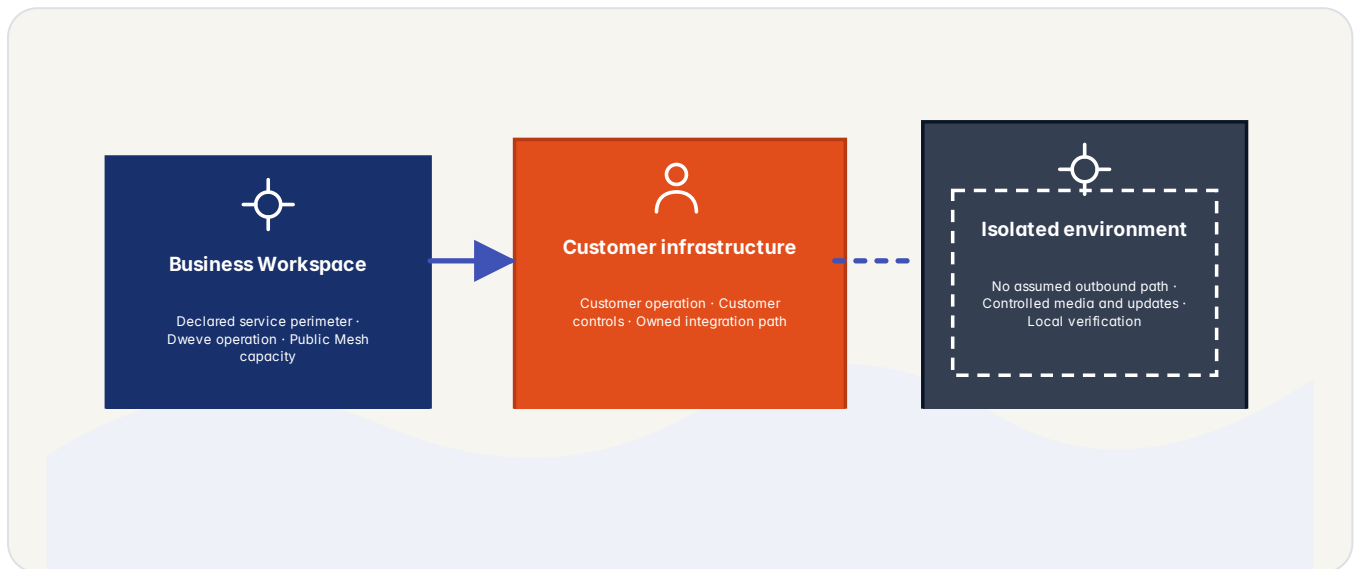
QUESTION TO CARRY

Does the retained record answer the next difficult question without reconstruction?

DEPLOYMENT CHOICE

Choose the operating posture around the work

Location is only one dimension. Decide who operates, who holds keys, which dependencies exist, how updates arrive, and what continuity requires.



How to read this visual: follow a question through the stack through deployment choice.

MECHANISM

Compare operating responsibility, not location alone

Sovereign deployment means the platform runs where you decide: managed through Fabric on the public Dweve Mesh, on your own hardware, or air-gapped. No re-platforming between tiers. And the rules you answer to are built in: GDPR, the EU AI Act, NIS2, and DORA are design constraints the platform satisfies by construction.

Move toward a licensed deployment when regulation, security posture, or procurement requires the platform to run entirely on your infrastructure.

- Begin through managed Fabric
- Run on customer infrastructure
- Close the environment
- Experience

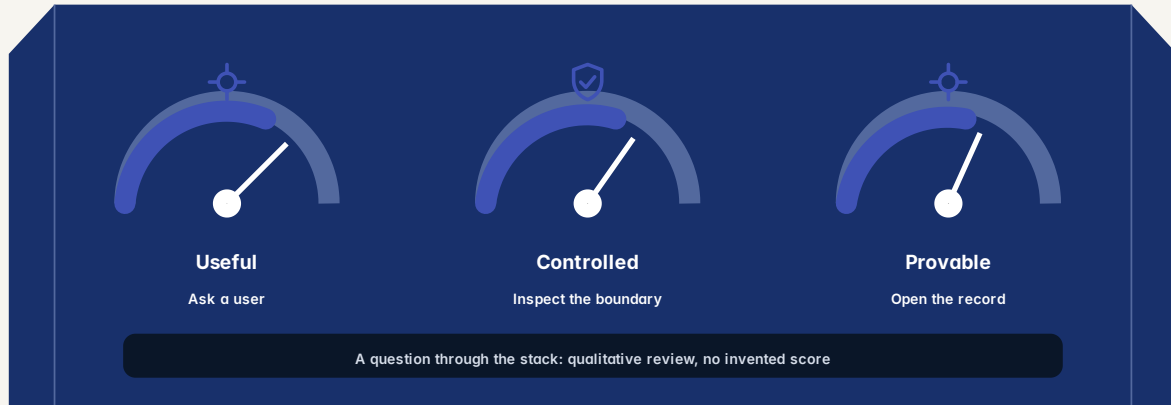
QUESTION TO CARRY

Who holds infrastructure, keys, updates, logs, recovery, and the exit path?

WHAT TO LOOK FOR

Three signals that the proof is earning trust

Progress is a useful outcome under understood control, supported by reviewable evidence.



How to read this visual: follow a question through the stack through what to look for.

DECISION

Use signals that can change the next decision

Check policy, risk, service level, and technical fit before treating existing hardware as eligible. Use that estate first, then add managed or dedicated capacity deliberately when measured demand or control needs require it. The placement decision records why each route was chosen.

Fabric holds the workflow; Nexus runs agent operations; Aura supports development; Loom composes cognition; Spindle maintains knowledge; Core and Kera execute; Mesh places workloads. Shared contracts reduce handovers across tools and vendors while one record carries control and cost evidence.

- Useful
- Controlled
- Provable
- Experience

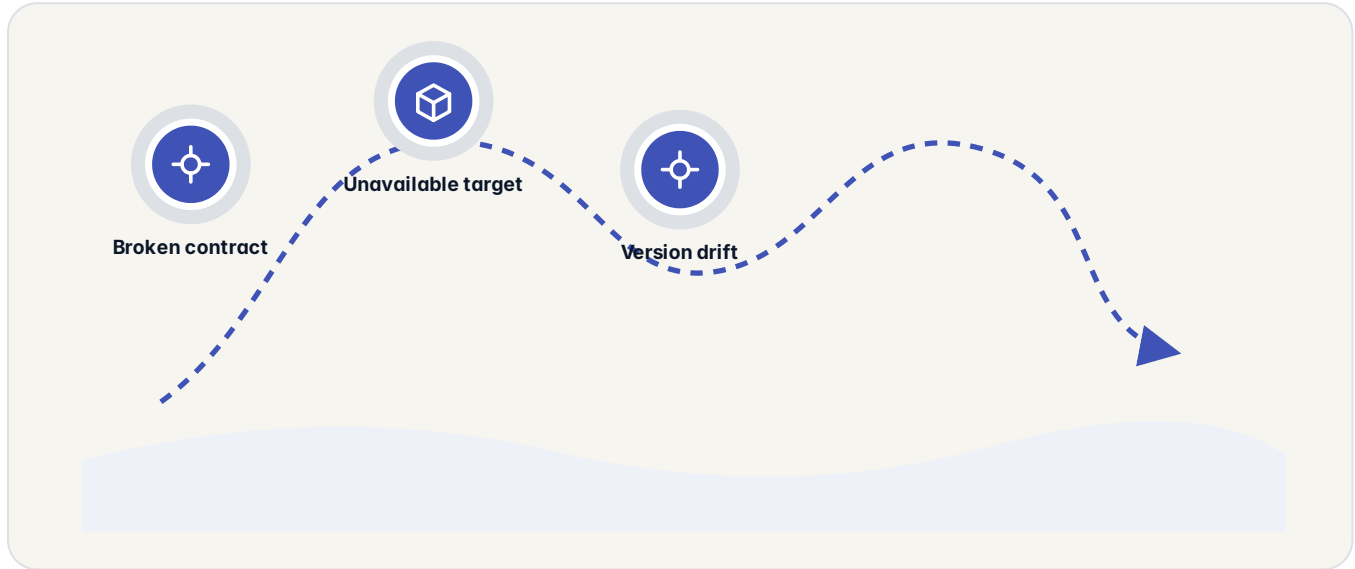
QUESTION TO CARRY

Which signals would change the next decision rather than merely impress the room?

HONEST LIMITS

Design failure before the first success demo

Visible failure is more useful than a polished result whose limits remain unknown.



How to read this visual: follow a question through the stack through honest limits.

MECHANISM

Design the failure before presenting the success

Mesh filters available, local, managed Mesh, and isolated capacity by policy and risk, then weighs cost and urgency among eligible routes. Operations gets a visible placement decision; finance can use suitable existing capacity first and reserve higher-control routes for work that requires them.

You work entirely in Fabric. Read files, ask follow-up questions, check sources, save a correction, and reuse a routine without moving the task between apps. Your folders and threads stay familiar from the first question to the finished result.

- Broken contract
- Unavailable target
- Version drift

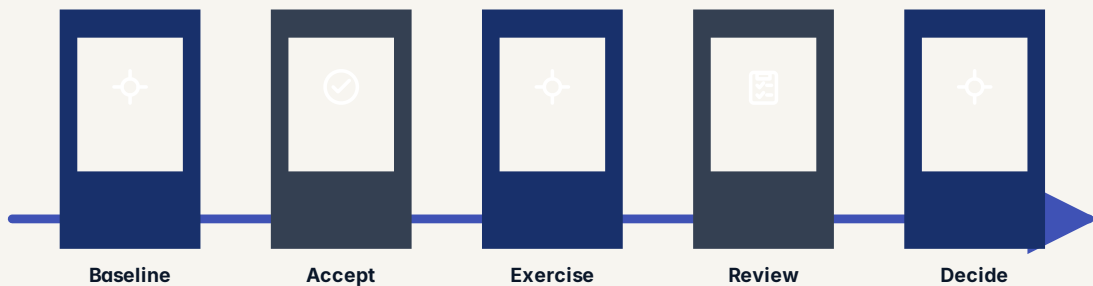
QUESTION TO CARRY

What happens when a source, permission, target, model, or supplier is unavailable?

ACCEPTANCE PATH

A proof should close with a decision, not applause

Agree the decision path before the demonstration so a strong or weak result can change what happens next.



PROCEED · NARROW · HOLD · REJECT

How to read this visual: follow a question through the stack through acceptance path.

DECISION

Close the proof with an owned decision

Dweve keeps the question, source material, open risk, accountable owner, and approval state in one decision record. Operations, finance, and control teams review the same basis instead of reconciling separate files after the meeting. The record closes only when the next action is clear.

Choose one live workflow with a known baseline, a named owner, and a decision that matters. Run it through the complete path and compare the agreed operating and finance measures. Add adjacent teams or workflows only after the owner accepts the improvement.

- Baseline
- Accept
- Exercise
- Review

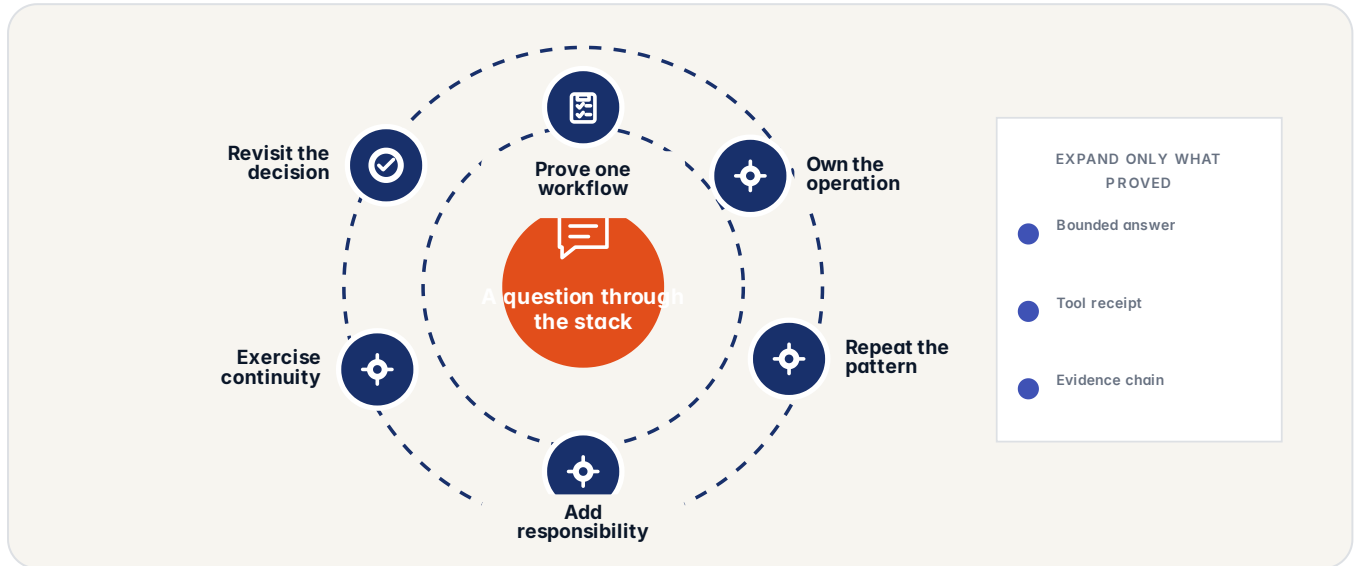
QUESTION TO CARRY

Are success, counter-metrics, failure cases, and stop conditions agreed first?

ADOPTION

Expansion is a sequence of owned choices

Move from one bounded result to a repeatable operating capability without making the scope vague.



How to read this visual: follow a question through the stack through adoption.

MECHANISM

Expand only what proved reusable

Most people open managed Fabric in a browser and start without an infrastructure project. Organisations that need direct product operation inside their own perimeter move to a licensed deployment. Workflow and evidence contracts remain portable, while product access, keys, patches, logs, and operating responsibility change.

Start with one workflow, one team, and a clear scope. Managed Fabric is ready on day one. Run the proof, then measure review speed and control.

- Prove one workflow
- Own the operation
- Repeat the pattern
- Add responsibility

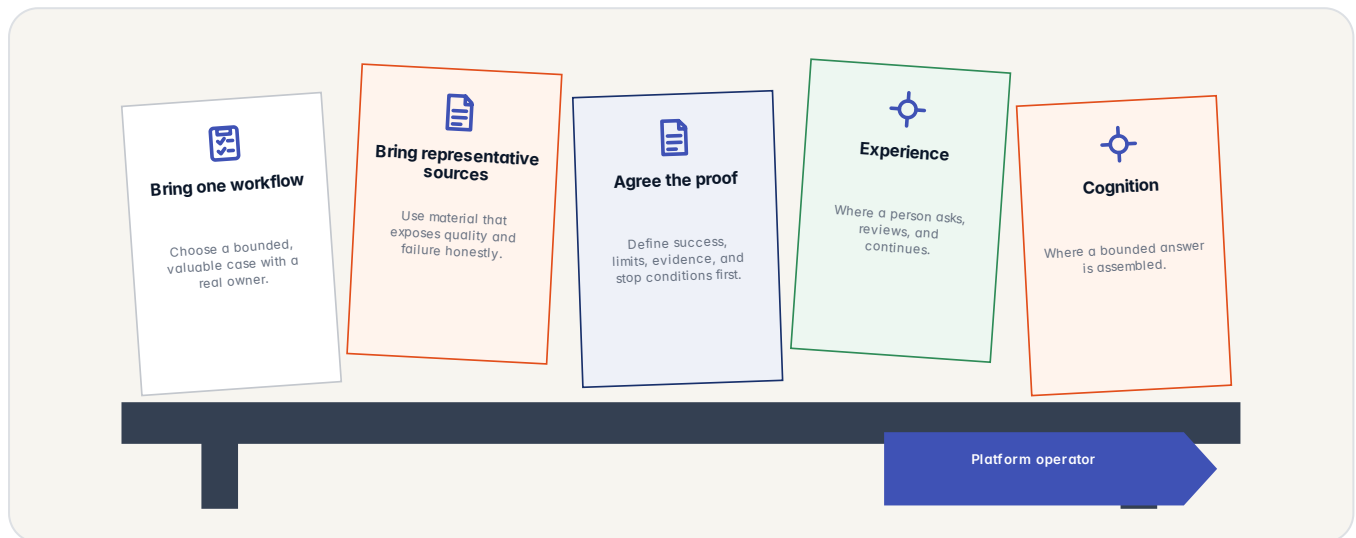
QUESTION TO CARRY

What can be reused safely, and what requires a new decision and proof?

WHAT TO PREPARE

Bring the ingredients for a useful first session

A small amount of real context produces a better conversation than a broad catalogue of hypothetical features.



How to read this visual: follow a question through the stack through what to prepare.

KEEP IT BOUNDED

Sensitive production data is not required for the first conversation. Bring representative material and the rules that define a useful result.

DECISION

Bring enough reality to make the session useful

Most AI apps keep the conversation inside one assistant. Need another opinion? Open another app. Need a different kind of AI? Upload the file again. Need to compare the answers? Copy them into a third place. Fabric keeps the task together. You can ask a question, add a document, invite another kind of AI, bring in a helper, save the useful process, and share the result with its source attached. The helper can change. The work does not start over.

When a task comes back every week, save the useful pattern instead of rebuilding it. Fabric can gather the same kind of material, prepare the same first pass, and leave the final choice with you.

- Bring one workflow
- Bring representative sources
- Agree the proof
- Experience

QUESTION TO CARRY

Which real sources, rules, owners, and examples will make the first session honest?

THE INVITATION

Continue with the question that matters

Scope a briefing, demonstration, or proof around the workflow and decision that matter to your organisation.

●
dweve.com/contact

✓

→

≡

ONE MAINTAINED PUBLIC ROUTE

Current contact choices stay on the website

CONTINUE ONLINE

dweve.com/contact

Current route and contact choices

START A CONVERSATION

Continue through the current contact page

Scope a workflow or ask a technical question, then use the contact page to begin.

START	dweve.com/contact Use the maintained public contact route
BRING	One bounded question Workflow, audience, decision, and desired evidence
CHOOSE	Briefing · demonstration · PoV Select the smallest useful next step
VERIFY	Current scope and terms Confirm availability, pricing, and responsibilities before commitment

MECHANISM

Turn the brochure into one bounded next step

The proof packet is created as the workflow moves, not rebuilt later. Source, rule, owner, and replay are product properties attached while the work happens, so an audit request is a retrieval, not a reconstruction.

The work moves forward. Owner assigned, sources cited, review state visible, and a proof packet assembled as the workflow runs.

- Start: dweve.com/contact
- Bring: One bounded question
- Choose: Briefing · demonstration · PoV
- Verify: Current scope and terms

QUESTION TO CARRY

What is the smallest useful next conversation Dweve should prepare?