

PRODUCT HANDBOOK · P01

The Fabric product handbook

The work-centred surface where people, models, agents, teams, tools, sources, flows, and decisions stay attached to one workspace object.

WORK OBJECT

PARTICIPANT PORTS

SOURCE-LINKED ANSWERS

TENANT AUDIT CHAIN



Fabric is the workspace and stable work object. Models, agents, people, teams, tools, and sources participate without becoming the work itself.

P01

CONTENTS

Fabric, from first question to durable work

This handbook follows one policy question through the browser surface, source projection, participant choice, saved flow, shared answer, API call, audit chain, and operational proof.

PRIMARY READERS

Product, architecture, delivery and operations teams

READING DEPTH

Professional + technical

DECISION THIS SUPPORTS

A product-specific handbook for teams evaluating Fabric as the place where questions become inspectable, reusable work.

01

The work object

03–05

Why a workspace is not a chat transcript or model session

02

Compose the work

06–10

Participants, context, sources, flows, budgets, and permissions

03

Expose the record

11–15

API parity, sharing, audit, failure, and deployment

04

Prove the product

16–17

Acceptance corpus, production record, sources, and next action

05

Subject index

18–19

An alphabetical finder for concepts, controls, products, and decisions.

HOW TO USE IT

Replace the policy question with one real Fabric workflow. Keep the boundaries and evidence questions, but validate current interfaces, limits, retention, pricing, deployment, and service terms against the live product description and agreement.

SOURCE DISCIPLINE

Dweve-specific facts are taken directly from the English website text audited 2026-07-20; web-navigation wording is humanised for print. Evaluation guidance is editorial, and scenarios and derived visuals are illustrative. Unsupported synthesis is replaced by canonical copy or omitted.

THE WORK OBJECT

Fabric is a workspace, not a model wrapper

A model can produce an answer. Fabric owns the work object that gives the answer a purpose, participants, sources, decisions, sharing state, and history.

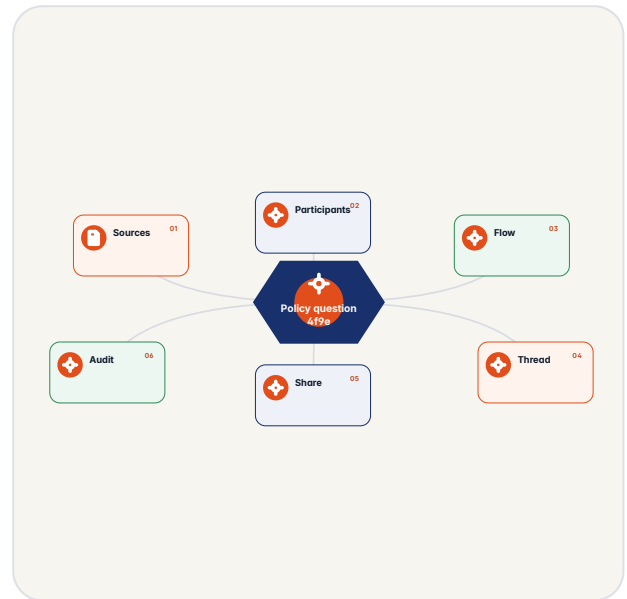
An analyst asks: “Which clauses in the new procurement policy change our approval route?” In Fabric, the question is not left as a disposable prompt. It becomes work with workspace identity, purpose, selected documents, a source-linked answer, follow-up thread, participants, and a shareable result. The model may change between attempts. The work object remains the thing the team can inspect.

The stable centre

The website describes one work object shared by the dashboard, REST, WebSocket, and flows. That object can carry people, models, agents, teams, tools, governed objects, knowledge references, budgets, and a tenant audit chain without treating any one participant as the owner of the work.

FABRIC TEST

If a colleague receives only a screenshot of the answer, the conclusion travelled but the work did not.



The workspace surface keeps the question, selected context, participants, and resulting record visible as one object.

WORK OBJECT • SOURCE-LINKED RESULT

The answer should survive the screenshot test

Fabric turns a question into work when the next person can inspect what was asked, what was included, what answered, and what remains uncertain.

MESSAGE	FABRIC WORK
A flattened answer Fast to forward, hard to verify - text without purpose - provider session hidden - sources detached - no reusable next step	A connected result Readable and inspectable - stable work identity - selected source subset - participant and decision state - thread, flow, share, and audit references

Fabric does not need to expose every internal detail to preserve the material context of the work.

A shared Fabric answer can carry the selected reply, relevant document or page references, follow-up thread, and intended access scope. The receiver does not need the rest of the workspace. The point is a source-linked slice that can be checked without recreating the analyst's path from memory.

Keep three boundaries visible

- The workspace boundary: who may see or change the work object?
- The evidence boundary: which sources, participants, decisions, and limitations support this answer?
- The sharing boundary: which part may travel to this person, team, or application?

USER-FACING QUESTION

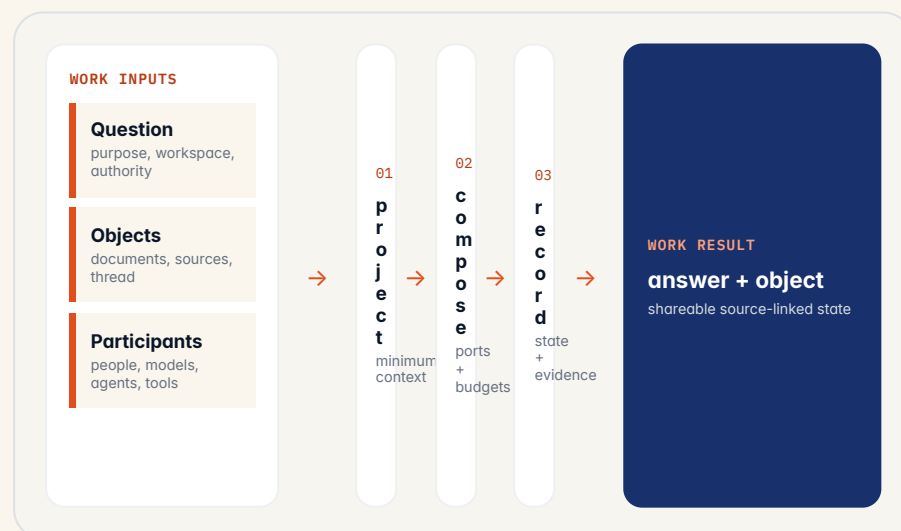
Can the recipient ask "where did this come from?" and reach the selected source or thread without receiving unrelated private work?

01

COMPOSE THE WORK

Keep the work object stable while the participants change

Fabric's participant ports distinguish people, models, agents, teams, and tools. That distinction lets a team change help without changing what the work means.



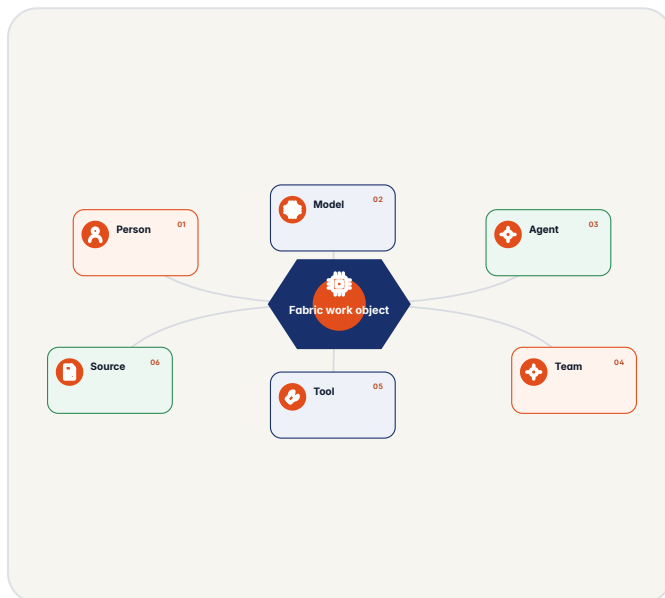
The model is selected inside the work. The work object is not selected by the model.

COMPOSITION • PEOPLE, MODELS, AGENTS, TEAMS, TOOLS

Participant ports are not a provider directory

The same work can include a human question, a model response, an agent task, a team review, and a tool result without collapsing those roles.

Fabric's participant ports are useful because they answer different questions. A person may own the purpose. A team may own review. A model may propose language. An agent may execute a bounded task. A tool may retrieve or calculate. The work object records who or what participated and what authority each had.



Ports show who or what contributed, with the work object retaining capability, authority, and result state.

Port questions

- What capability is this participant allowed to exercise?
- Which input did it receive and which output did it return?
- Did it recommend, transform, retrieve, calculate, communicate, or act?
- Who can challenge, revoke, replace, or approve it?
- What evidence and limitation should remain attached to the contribution?

PROVIDER BOUNDARY

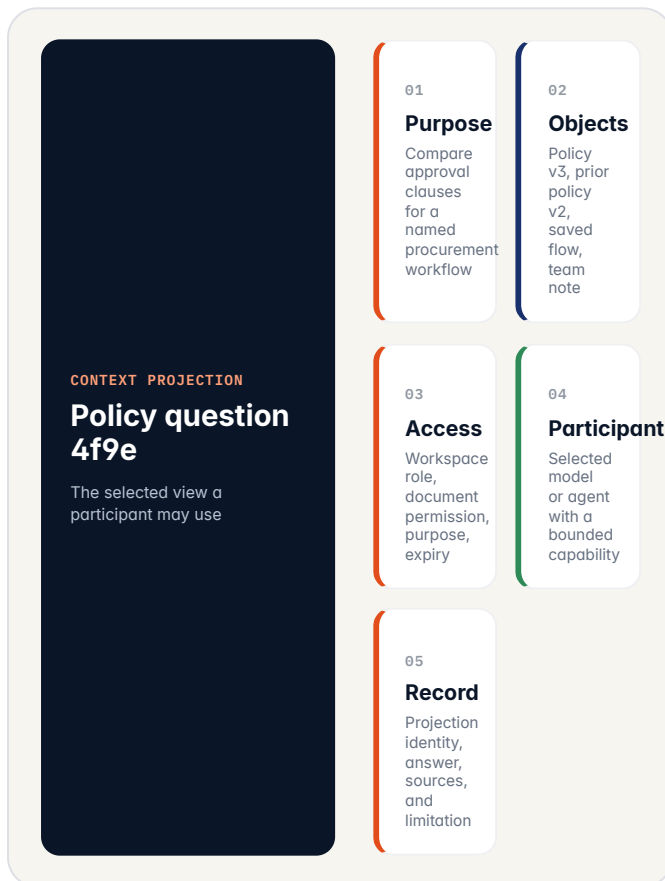
Swapping a model can change an answer. It should not change the identity, purpose, permissions, or audit meaning of the work object.

COMPOSITION • SELECTED OBJECTS

Context projection is a privacy and quality control

A workspace can contain a lot. A participant should receive the minimum authorised context needed for its task, with references to the rest rather than an uncontrolled dump.

The procurement question may need policy pages, a saved approval flow, and one team decision. It does not automatically need every document in the workspace, another project's thread, or a broad customer export. Fabric's work object can carry selected objects and references while keeping workspace context outside the participant's projection.



The work object can remember what was projected without exposing the whole workspace to every participant.

Projection acceptance

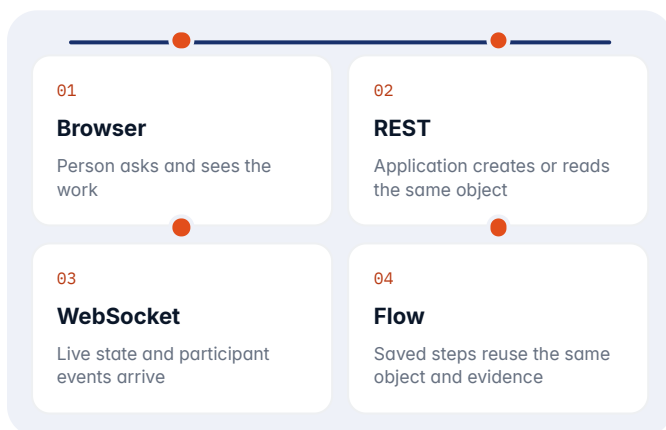
- Remove one source and verify the participant cannot infer it from a hidden broad context.
- Revoke a document after projection and define whether the run may complete or must stop.
- Add a private thread and verify the shared answer does not include it by default.
- Compare the source-linked answer with the exact selected source subset.
- Retain the projection and access decision so a reviewer can explain why this participant saw this context.

COMPOSITION • INTERFACE PARITY

The browser, REST, WebSocket, and flow are surfaces over one object

A dashboard interaction, API request, stream, and saved flow can feel different while producing one accountable Fabric state model.

The product description describes the same work surface through dashboard, REST, WebSocket, and flows, with workspace-scoped keys and versioned contracts writing to one tenant audit chain. The implementation must make the semantic relationship explicit: a request created by the API should be visible in the workspace; a browser action should have a stable API or event interpretation; a flow should not create a second hidden work model.



Transport changes; work identity, state, permissions, and evidence do not silently fork.

Parity exercise

Create the same policy question through the browser and workspace-scoped API. Add the same source, select the same approved participant, produce a result, revoke sharing, and inspect the audit chain. Compare work identity, source subset, participant record, result state, share state, and timestamps.

TRANSPORT IS NOT OUTCOME

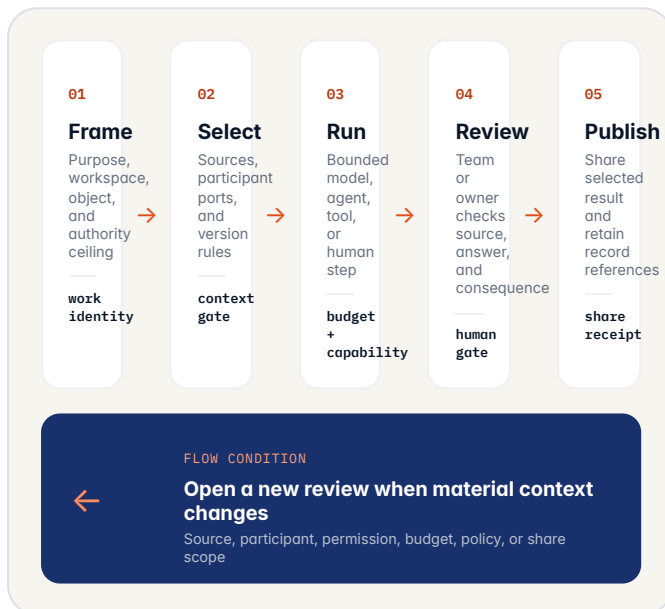
A successful HTTP request can create an awaiting-review work state. A disconnected stream can occur after the model responded. The caller must read Fabric state rather than infer business outcome from transport.

COMPOSITION • FLOW GRAPHS AND BUDGETS

Saved flows turn good work into a repeatable shape

A flow is useful when it makes a bounded sequence reusable while keeping each participant, source, approval, budget, and result visible.

The procurement question can become a saved flow: identify the policy version, retrieve selected clauses, ask a bounded comparison, route the result to a procurement reviewer, and publish only the approved source-linked note. The flow should not hide a broad autonomous loop behind a friendly name.



A saved flow is a visible graph of work, not a prompt template with hidden state.

Flow controls

- version the flow and make active versus historical versions explicit
- give every step an owner, capability, input, output, timeout, budget, and failure state
- prevent a retry from duplicating a share, external action, or audit event
- let a human return, amend, or stop the flow without editing history
- retain the flow identity and selected participant versions in the work record

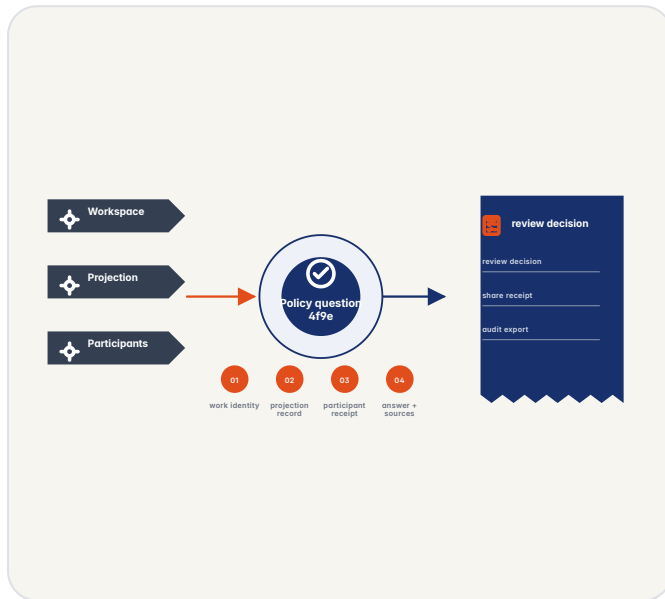
FLOW QUESTION

Can a colleague understand what the saved flow will do before it is run, and can the owner prove what it actually did afterward?

EXPOSE THE RECORD

One tenant audit chain is useful only if the work stays joined

The audit chain should connect work, participants, source projections, decisions, sharing, and changes without becoming an unreadable event dump.



The trace turns a workspace question into a source-linked answer, a review decision, and a portable record.

EXPOSE THE RECORD • ACCESS

Permissions describe capability, not just membership

A viewer, editor, administrator, model, agent, or tool should not inherit every operation merely because it can see the workspace.

Fabric's product description describes permissions and roles as part of the workspace surface. A practical model separates reading knowledge, editing knowledge, managing workflows, managing budgets, reading the audit chain, inviting participants, sharing work, and administering workspace policy. A role label is only useful when it maps to explicit capabilities and evidence.

CAPABILITY	EXAMPLE OWNER	EVIDENCE TO RETAIN
Read selected sources	viewer / model port	object, purpose, expiry
Edit a saved flow	editor / flow owner	version, diff, approver
Invite a participant	workspace admin	actor, capability, scope
Run a paid or bounded model	authorised caller	budget, provider, result
Share an answer	work owner	recipient, subset, receipt
Read audit chain	auditor / owner	query, range, export

The capability matrix is a product contract; exact role names and limits must match implementation.

Access tests

- A private source must not leak through a generated answer or shared link.
- A model port must not invite another participant or change workspace policy unless explicitly authorised.
- A revoked editor must not mutate a saved flow through an old token or stream.
- A shared answer must preserve its intended expiry and recipient boundary.
- An audit reader must not gain authority to alter the work object.

SECURITY BOUNDARY

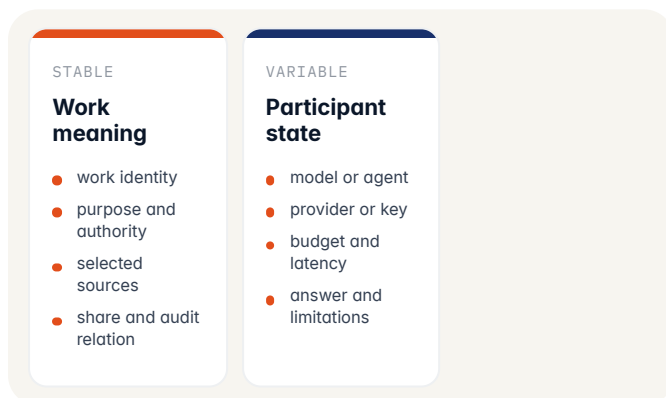
Workspace visibility, participant capability, source permission, and audit access are different controls. Passing one does not imply the others.

MODEL, AGENT, TOOL, AND BUDGET SELECTION

Provider changes should not fracture the work

Fabric can select Loom, another approved model, an agent, or a tool. The work object should preserve why the participant was selected and what changed when it was replaced.

The product description describes a model-agnostic interface, Loom as the default model route, external provider keys, agent and workflow surfaces, and one tenant audit chain. A product proof should not assume that “same interface” means same result. It should make provider, model or agent version, key boundary, budget, prompt or flow version, source projection, and result differences visible.



Fabric should keep the stable work record while making participant variation explicit.

Provider-swap exercise

Run policy question 4f9e with the default route and one approved alternate. Compare source retrieval, answer, citations, participant manifest, token or budget state, latency, refusal, and share result. The exercise should make a changed answer visible and keep the same work identity and review requirement.

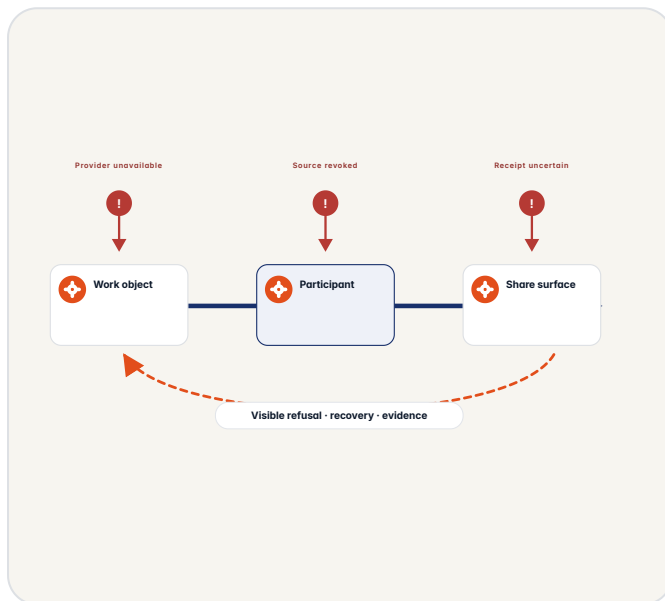
NO SILENT DOWNGRADE

When a selected participant is unavailable, Fabric must expose refusal, fallback, or degraded state. It should not silently substitute a provider and present the result as though nothing changed.

EXPOSE THE RECORD • RECOVERY

A workspace must fail without losing the work

Provider errors, exhausted budgets, revoked sources, failed flow nodes, and disconnected clients should produce inspectable states rather than a missing conversation.



A failed participant or surface returns to the work owner as a visible state instead of erasing the work.

Recovery questions

- Can the user see which attempt failed and whether any answer or share effect occurred?
- Can an administrator restore the work object without restoring revoked source access?
- Can the same flow retry idempotently without duplicating an external action or audit event?
- Can the workspace export the work, source references, participant manifests, and chain state?
- Can a clean deployment verify the record without the original provider session?

02

PROVE FABRIC

Make the workspace trustworthy as a service

The product becomes dependable when its object, ports, source projections, interfaces, permissions, failure states, and evidence survive production conditions.

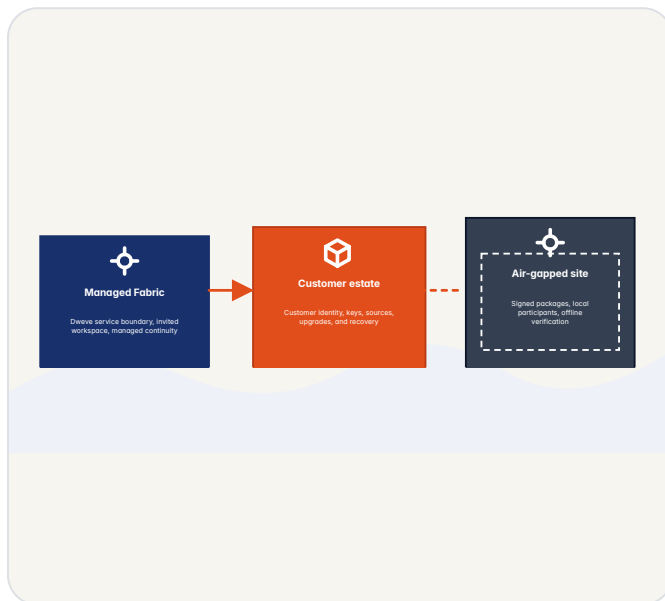
FABRIC PROOF · 4F9E		From request to durable work record	
01	WORKSPACE IDENTITY which tenant and work object are in scope	Fabric	bound
02	SOURCE PROJECTION which documents and permissions were used	workspace	scoped
03	PARTICIPANT MANIFEST which models, agents, people, or tools took part	port owner	versioned
04	FLOW AND BUDGET what sequence and resource ceiling ran	work owner	recorded
05	REVIEW AND SHARE who accepted and what subset travelled	team owner	signed
06	AUDIT EXPORT the work can be checked outside the live session	workspace admin	portable

A Fabric proof is a work-object proof, not a model benchmark detached from workspace state.

PROVE FABRIC • POSTURE

Deployment changes the workspace responsibility boundary

Managed Fabric, licensed operation, and air-gapped use can share object and API semantics while carrying different identity, key, data, update, support, and recovery duties.



The deployment shape changes the operating boundary; the Fabric work and evidence contract must remain testable in each posture.

The deployment page describes managed, licensed, and air-gapped responsibilities. Fabric evaluation should begin with a bounded workflow and synthetic or authorised documents. The team then proves identity, source access, participant controls, key handling, API parity, audit export, update, support, backup, restore, and exit in the selected posture.

NO POSTURE SHORTCUT

"EU" or "air-gapped" is not a complete deployment description. Record where data, keys, model/provider calls, audit records, updates, support, and recovery actually travel.

THE ACCEPTANCE CORPUS

Run the workspace against work that can go wrong

A Fabric proof includes ordinary questions, source and permission changes, participant failure, provider substitution, flow retry, sharing mistakes, audit export, and restore.

CORPUS SLICE	INCLUDE	EXPECTED OBSERVATION
Normal work	question, selected policy, model, source-linked answer	object, thread, sources, and share close
Source change	document superseded or withdrawn after the run	historical source remains resolvable
Permission change	source revoked or recipient removed	new access and old record stay distinct
Participant fault	model unavailable, agent timeout, tool refusal	fallback or failure is explicit
Provider swap	approved model or key changes	manifest and answer difference are visible
Flow retry	timeout after possible external/share effect	idempotency and reconciliation hold
Abuse	cross-workspace query, poisoned document, broad export	denied, scoped, and audited
Recovery	backup, clean restore, offline audit verification	work and evidence remain coherent

The corpus is part of the product decision. A successful chat path alone is not acceptance.

Measure the work object, not only latency

- work identity and state parity across browser, API, stream, and flow
- source projection, participant, budget, permission, and share alignment
- provider/model change, refusal, fallback, and limitation visibility
- audit-chain completeness, export, retrieval, and independent checking
- access denial, incident record, restore outcome, and exit package

Record the raw cases, expected object transitions, failed assertions, answer differences, reviewer decisions, and remaining gaps. A product owner needs evidence that explains whether Fabric is reliable for this work, not a chat quality score detached from the workspace boundary.

PRODUCTION RECORD

The organisation decides whether Fabric becomes a dependency

A product proof ends with a recorded adoption decision, not with a polished workspace tour.

The product owner, workspace administrator, engineering lead, security owner, knowledge owner, operational reviewer, procurement/legal owner, and affected users may each own a different condition. They decide the workflow, workspace and data scope, participant and provider boundary, permissions, API surfaces, budgets, support, retention, export, posture, recovery, and exit state.

FABRIC PRODUCTION RECORD	Policy question 4f9e proof decision
WORK	One source-linked policy question and saved flow BOUNDED
WORKSPACE	Named tenant, team, documents, participants, and users SCOPED
CONTRACT	Dashboard + REST + WebSocket + flow state parity LINKED
AUTHORITY	Roles, ports, sharing, budget, and audit access REVIEWED
OPERATION	Support, limits, update, restore, export, incident OWNED
EXIT	Work, source, chain, and participant export/transition DATED
DECISION	Go / narrow / hold / stop SIGNED
The record should remain understandable after the implementation team and first workspace administrators have moved on.	

The smallest credible next step

Run the policy question through browser and API surfaces with one selected source, one participant swap, one review, one share, and one failure/recovery exercise. Make the user, workspace admin, security owner, source owner, and recipient inspect the same work record. If their questions require reconstruction, narrow the proof until they can answer them.

PRIMARY ACCEPTANCE CRITERION

Fabric preserves one work identity and exposes its selected sources, participants, decisions, share state, result, failure state, and audit references across the chosen interfaces and deployment posture.

That proves a bounded workspace responsibility. It is not permission to generalise from one policy question to every team, data class, provider, agent, or business process.

FIND A SUBJECT

Subject index

Page references point to the substantive explanation, worked example, control, boundary, or decision record, not merely to a passing label.

A Acceptance criteria 7, 16–17	A Accountability 8	A Adoption 17
A Air-gapped deployment 15	A Appeal and challenge 6	A Audit 3–4, 6, 8–17
A Authority 5–6, 9, 11–12, 17	B Benchmarking 14	B Boundaries 4, 6, 11–12, 15–17
C Capabilities 6–7, 9, 11	C Continuity 15	C Contracts 11, 15, 17
C Cost and budgeting 9, 11–12, 14, 16–17	D Dependencies 17	D Deployment 13, 15, 17
E Evidence 4–6, 8, 11, 13–16	E Exit 15–17	F Fabric 3–8, 10–17
F Failure 9, 13–14, 16–17	I Identity 3–4, 6–10, 12, 14–17	I Incident response 16–17
I Inputs 6, 9	L Ledger 17	L Limits and exclusions 6–7, 11, 16–17
L Loom 12	M Manifests 12, 14, 16	M Models and solvers 3, 5–9, 11–12, 14–16
O Operations 11, 15, 17	O Outputs 6, 9	O Ownership 3, 9, 11, 13–14, 16–17
P Policies 3, 7–12, 16–17	P Privacy 7	P Procurement 3, 7, 9, 17
P Proof 12, 14, 16–17	R Recovery 13, 15–17	R Responsibility 15, 17

SUBJECT INDEX CONTINUED

Subject index · R–W

Terms are grouped alphabetically. Consecutive page references are collapsed into ranges.

R**Retention**

17

S**Security**

11, 17

S**Sources**

3–17

S**Sovereignty**

6–7, 11–12

S**State**

3–6, 8–9, 12–14, 16–17

T**Testing**

3–4

T**Trace**

10

V**Verification**

4, 7, 13, 15–16

V**Versioning**

9, 11–12

W**Workflows**

7, 12, 15, 17

CLOSING PRODUCT RECORD

Start with the work your team keeps losing

Choose one question that grows into real work, and prove Fabric can keep its context, participants, sources, decision, and share state together.

Write the workspace and work identity, purpose, selected objects, participant ports, context projection, model/agent/tool manifest, flow and budget, reviewer, share subset, API surface, audit chain, deployment posture, recovery, retention, export, and stop conditions.

Then choose the hard case: a revoked source, replaced model, expired share, duplicate flow retry, unavailable provider, private thread, over-broad export, or restored workspace with a changed permission. A handbook is useful when it makes the failure concrete enough to run.

WORK

One bounded question and result

Do not begin with every workspace

CONTEXT

Selected sources, permissions, and participant ports

Minimum authorised view

SURFACE

Browser, API, stream, and saved flow

One semantic object

EVIDENCE

Answer, source, decision, share, and audit references

No screenshot-only handoff

DECISION

Go, narrow, hold, or stop

Signed with exit and review triggers

THE FABRIC STANDARD

An answer is useful. Work becomes dependable when the workspace can show what was asked, what was included, who or what participated, what was decided, what was shared, and how the record can be checked later.