

PRODUCT HANDBOOK · P03

The Nexus product handbook

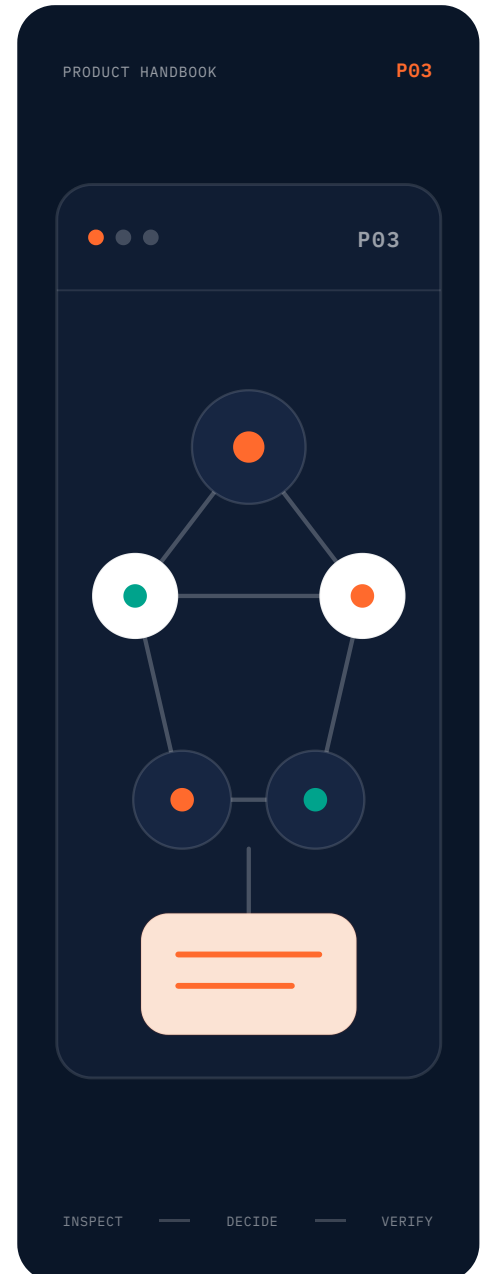
An executable organisation formed around one objective, with owned tasks, typed collaboration, memory by purpose, enforced authority, bounded recovery, and a work record that survives the team.

OBJECTIVE-FIRST TEAMS

TASK OWNERSHIP

AUTHORITY GATES

WORK RECEIPTS



Nexus forms a temporary organisation around the objective. Agents, models, tools, memory, policies, workflows, and people contribute without becoming the authority.

P03

CONTENTS

Nexus, from objective to accountable outcome

This handbook follows a service incident from its first objective through team formation, task graphs, collaboration, memory, approvals, recovery, observable work, deployment, and production proof.

PRIMARY READERS

Product, architecture, delivery and operations teams

READING DEPTH

Professional + technical

DECISION THIS SUPPORTS

A product-specific handbook for teams deciding whether Nexus can turn a multi-agent process into one accountable operation without inventing authority.

01

Form the organisation

03–07

Objective, team shape, roles, tasks, and operating memory

02

Run the work

08–14

Collaboration structures, authority, review, failure, and recovery

03

Keep the receipt

15–22

Evidence, replay, stack boundaries, deployment, and acceptance

04

Subject index

23–24

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

HOW TO USE IT

Replace the incident in this edition with one process that currently crosses people, systems, rules, and approvals. Keep the questions, then validate current interfaces, runtime limits, service terms, pricing, retention, and deployment against the live product and agreement.

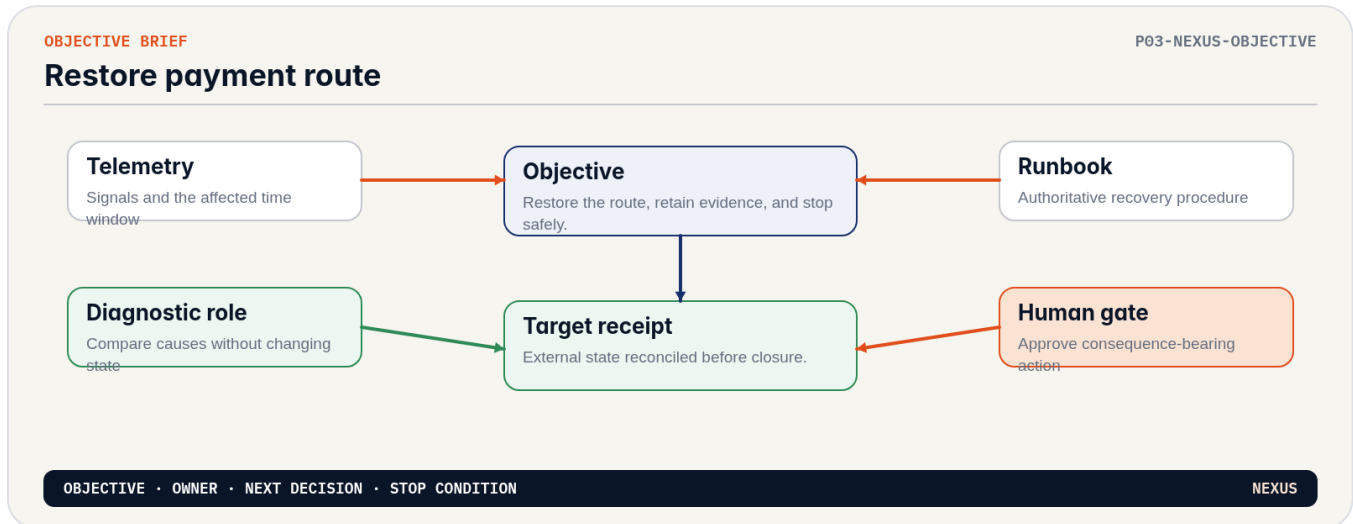
SOURCE DISCIPLINE

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.

FORM THE ORGANISATION • OBJECTIVE

Nexus begins with an objective, not a collection of bots

A model can answer and an agent can act. Nexus adds the missing operating question: who is responsible for bringing the whole objective to a controlled end?



The organisation changes shape during the incident. Accountability does not disappear between handovers.

Consider a service incident that requires containment, diagnosis, recovery, and a customer update. A prompt loop can ask several helpers for suggestions, but suggestions do not make an operation. Nexus turns the incident into one objective with acceptance conditions, source authority, action limits, named human authority, and a record of what eventually happened.

The objective is the durable centre

The team is allowed to change around the work. A failing specialist can be replaced, a reviewer can be added, and a blocked dependency can be escalated. The objective, its boundaries, and the responsibility for its outcome remain visible while those changes happen.

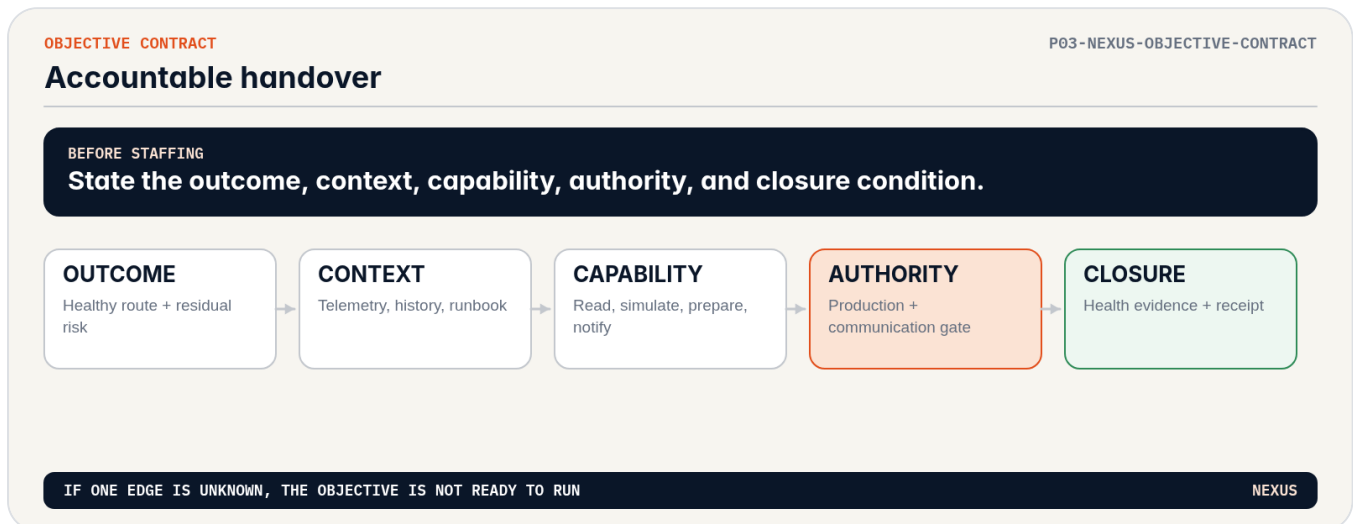
NEXUS TEST

If the operating lead cannot state the objective, current state, next decision, owner, and stop condition in one briefing, the organisation is still only activity.

FORM THE ORGANISATION • OBJECTIVE CONTRACT

An objective becomes executable when its edges are explicit

“Investigate the incident” is a theme. A runnable objective states the intended outcome, evidence available, permitted action, human gate, and conditions for a safe stop.



An objective is an operating contract. It gives the organisation somewhere precise to stop.

Nexus needs more than a headline. The incident objective should say what “restored” means, which systems and tenants are in scope, what telemetry and runbooks are authoritative, which actions may be prepared, which actions need approval, and what must be communicated. Without those edges, a capable agent can quietly turn an investigation into an unapproved change.

Before staffing begins

- What observable result would allow the owner to close the objective?
- Which source or system is authoritative when telemetry and a runbook disagree?
- Which proposed actions are reversible, and which change an external system?
- What must be true before a person is asked to approve?
- What evidence is retained if the objective is narrowed or fails?

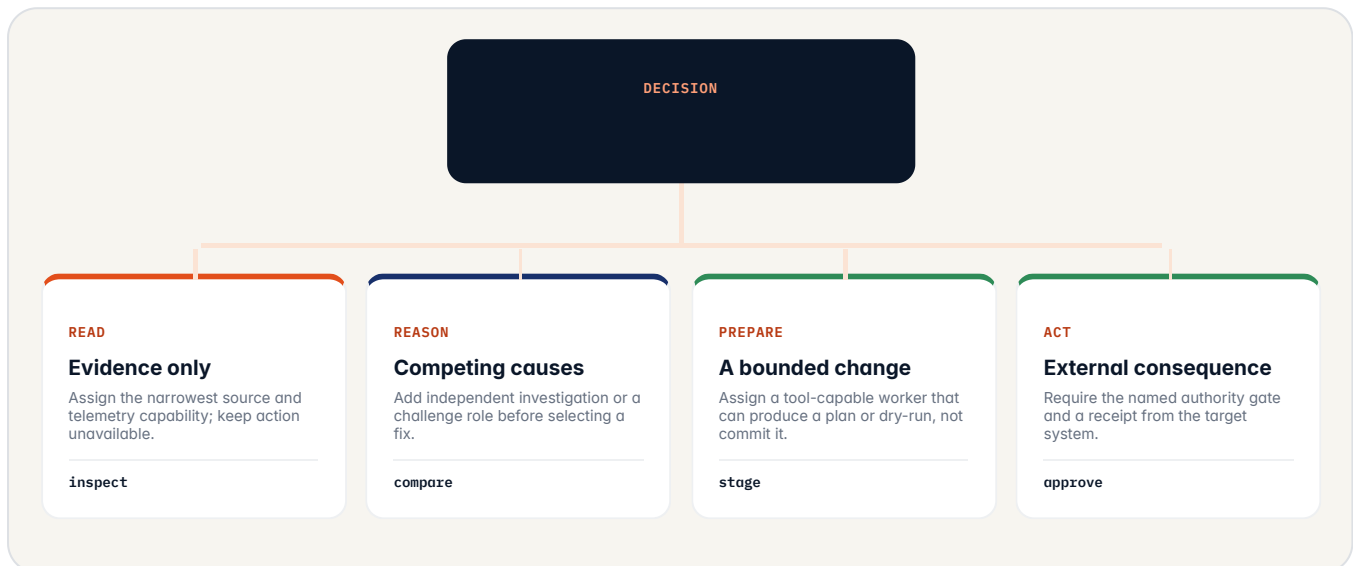
BOUNDARY OVER FLUENCY

A persuasive explanation cannot compensate for an objective whose success condition and authority ceiling were never defined.

FORM THE ORGANISATION · STAFFING

The smallest useful team is formed from the problem

Nexus does not keep a fixed cast of pretend employees behind every request. It selects roles and capabilities that the current objective actually needs.



Nexus chooses an organisation around the next responsibility, not around the number of available agents.

A small incident may need one diagnostic worker and a human lead. A payment outage may need a telemetry reader, a causal investigator, a change-preparation tool agent, a communications drafter, and a reviewer. A security-sensitive action may require a second independent challenge. The team is not a product catalogue; it is a temporary operating structure with named responsibility.

Staffing questions

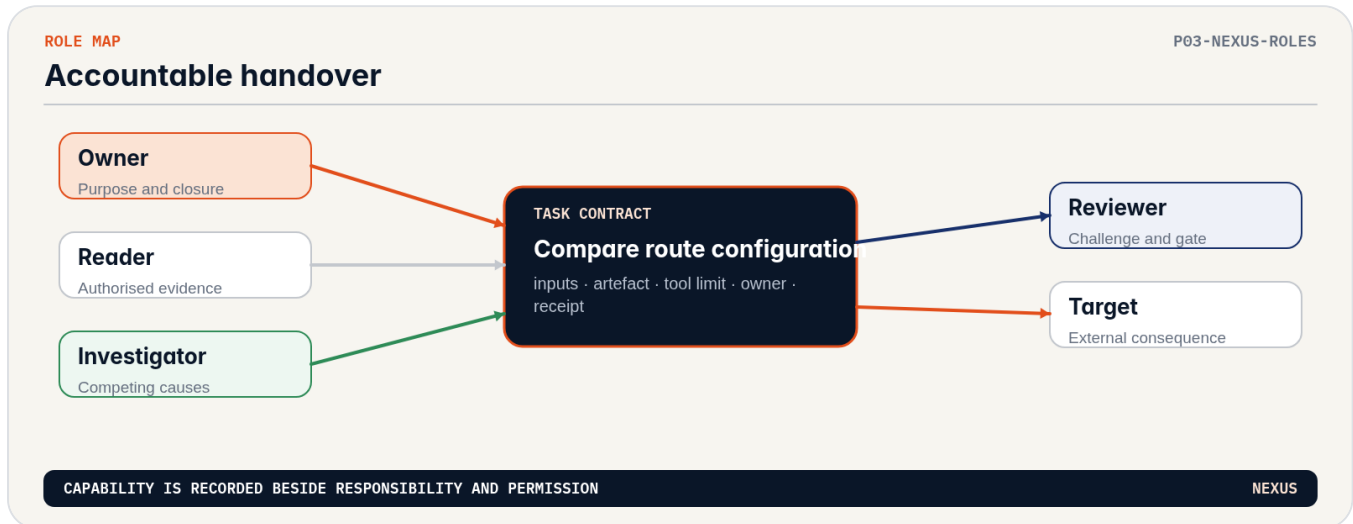
- Does each proposed role have a purpose and an expected contribution?
- Can the objective be completed with fewer participants?
- Which role owns the task if two specialists can perform it?
- What capability is needed, and what permission is deliberately withheld?
- What event would justify expanding or dissolving the team?

A useful staffing decision is inspectable. It names the candidate set, the match, the reason for the match, current load, and any fallback. It does not need an opaque “best agent” story when the organisation can show the capability and authority that made the assignment valid.

FORM THE ORGANISATION · ROLES

Roles are operating contracts, not assistant personalities

A role exists when the organisation can say what it receives, what it may produce, which tools it may use, how it is challenged, and where its responsibility ends.



The role is the boundary between a participant's capability and the organisation's responsibility.

Calling one worker "the analyst" and another "the manager" does not create an organisation. Nexus treats roles as runtime structure: purpose, responsibility, capability, resource limit, handover contract, policy checks, and escalation path. A reviewer can inspect without having permission to execute. A tool agent can prepare a change without being allowed to commit it. A supervisor can see the operation without owning every subtask.

Role acceptance

- Give a diagnostic role a misleading or incomplete source and verify that it flags the gap.
- Give a reviewer the ability to check but not execute, then verify the tool boundary.
- Replace one participant while preserving the role, task state, and expected artefact.
- Ask who owns a disagreement and verify that the answer is not "the last model to speak."
- Retain the role assignment and authority decision in the work record.

A ROLE IS REAL WHEN IT CAN BE CHALLENGED

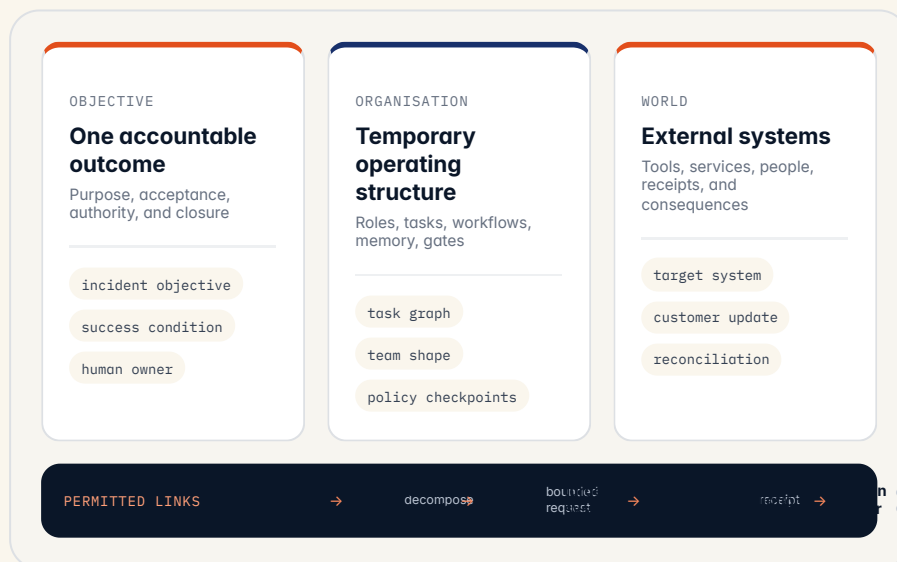
If a role cannot be refused, reassigned, audited, or stopped, it is probably a persona wrapped around an unowned action.

01

RUN THE WORK

Make tasks, teams, and handovers durable

Nexus keeps the task whole while participants change. The organisation is useful because state, ownership, and authority survive the movement between specialists.

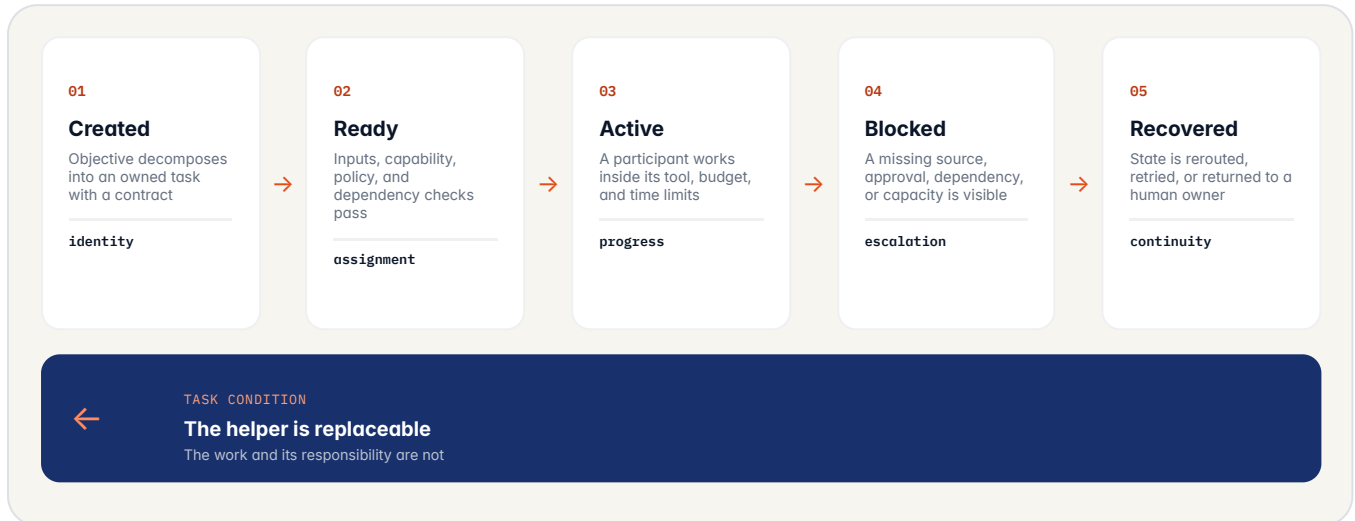


The organisation is between objective and world: it coordinates work without pretending to own the consequence.

RUN THE WORK • TASK STATE

A task is the durable unit of execution

Participants may be replaced. A task should not be silently restarted, duplicated, or reduced to a transcript when the helper changes.



A state machine makes a handover observable. A transcript makes the next participant guess.

The payment incident has a task to compare route configuration with the last healthy deployment. That task needs an owner, inputs, expected artefact, dependency state, tool allowance, deadline, retry budget, and escalation path. If the first specialist times out, the next specialist receives the task state and context that matters. It does not begin from a vague “please investigate” message.

Task contract checks

- Is the expected artefact concrete enough for another role to validate?
- Are dependencies and gates different from prose in a prompt?
- Can an observer tell ready, active, blocked, failed, and closed apart?
- Does retry preserve the original task identity and bound the number of attempts?
- When a task is reassigned, which state and evidence travel with it?

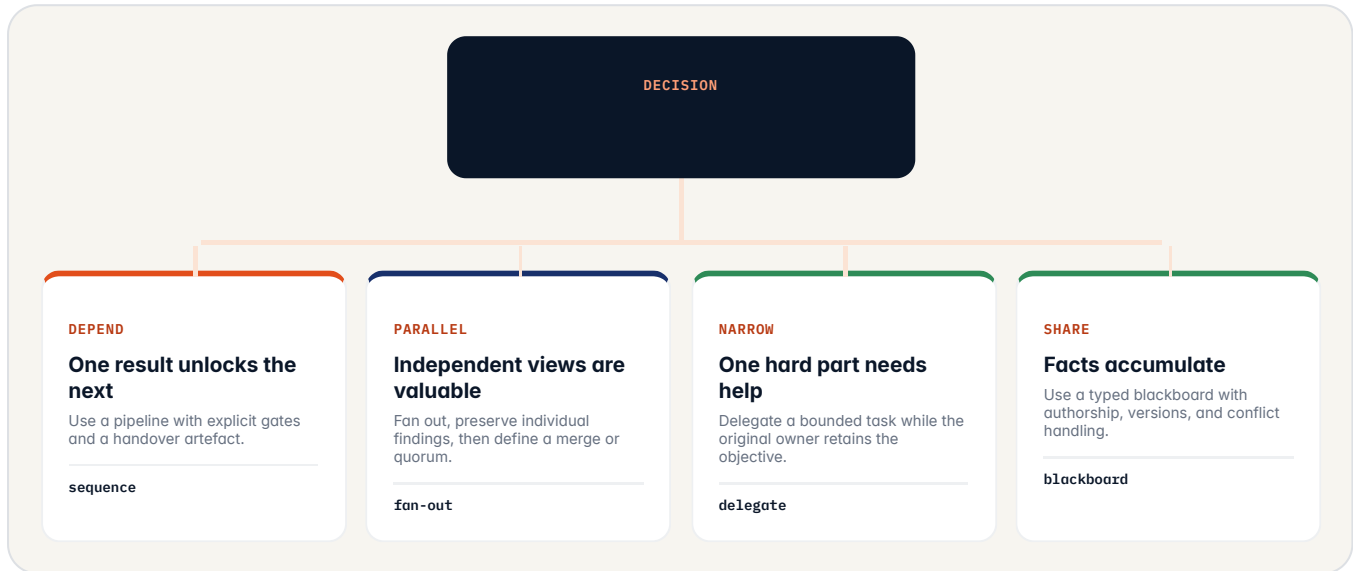
NO SILENT RESTART

If the same action can run twice because a timeout was mistaken for a failure before the target answered, the task contract is incomplete.

RUN THE WORK • WORKFLOW TOPOLOGY

Collaboration is a typed shape of work

Nexus can sequence, fan out, delegate, share structured findings, debate, and require a quorum. The choice should follow the dependency and risk of the objective.



A workflow type tells the runtime what it may do with state. It does not decide the customer's policy.

A pipeline is appropriate when one task feeds the next. Fan-out helps when several specialists can investigate independently. Delegation keeps a bounded subtask connected to the owner. A blackboard holds structured intermediate facts. Debate and quorum make competing interpretations visible before a merge. These are not chat-room moods; they are execution structures with entry conditions, handover points, merge rules, and completion states.

Structure before scale

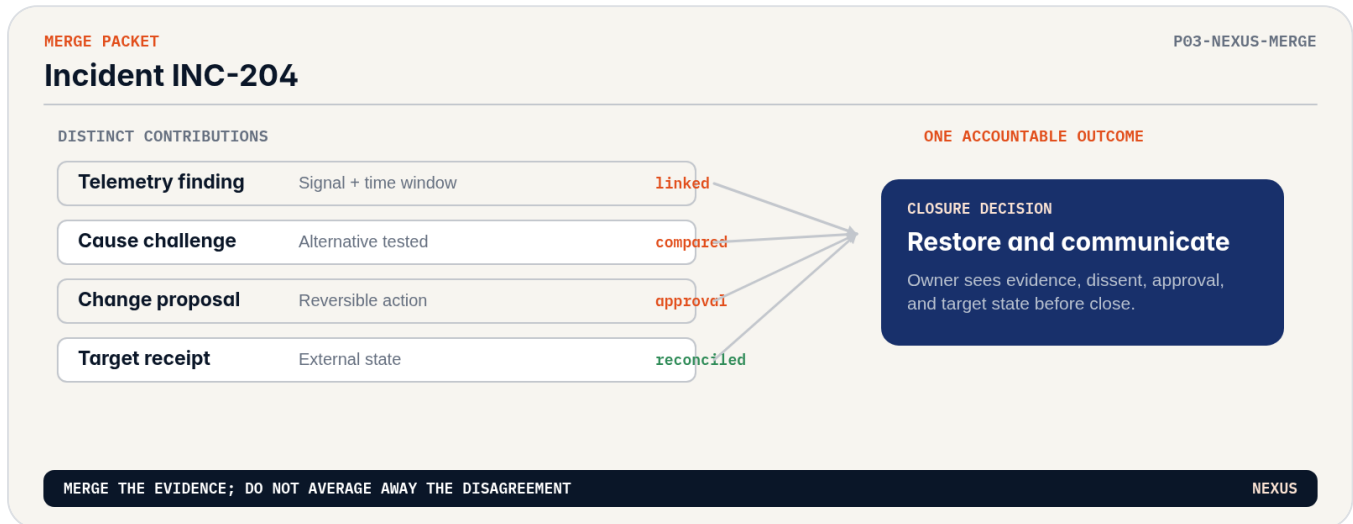
- Define entry conditions before inviting more workers.
- Give each branch a completion and failure meaning.
- Keep dissent as a typed contribution instead of averaging it into a smooth paragraph.
- Make the merge owner and merge criteria explicit.
- Reject cycles before the workflow reaches a live side effect.

The operational advantage is not that more agents speak at once. It is that the team can be monitored, constrained, tested, replayed, and explained as a known structure.

RUN THE WORK • CONTRIBUTIONS

The merge point is where accountability becomes visible

A customer should receive one coherent outcome, while the organisation preserves the distinct evidence and disagreement that made the outcome possible.



The final answer is compact. The work record keeps the contributions that make it defensible.

A merge is not a vote by default

The incident lead may accept one finding, reject another, request more evidence, or decide that the team cannot safely identify the cause. Nexus should preserve why the merge happened and who had authority to make it. A quorum is useful only when the objective defines what enough agreement means and what happens when it is not reached.

KEEP DISSENT ATTACHED

A contribution that lost the merge can still explain the residual risk. Deleting it makes the closed result look more certain than the operation was.

RUN THE WORK • MEMORY

Organisational memory has four different jobs

A task needs current state. A team may need previous cases, durable knowledge, and reusable procedures. Nexus keeps those purposes separate so context follows the role that needs it.

MEMORY PURPOSE	CARRIES	SHOULD NOT BECOME	ACCEPTANCE QUESTION
Working	Active task, constraints, intermediate findings, open decision	A permanent record of everything seen	Can a replacement continue without receiving unrelated work?
Episodic	Earlier cases, outcomes, recovery choices, and lessons	An unqualified precedent	Does the team distinguish similarity from authority?
Semantic	Facts, documents, relationships, and governed knowledge	A source of permission by itself	Is the source version and authority still visible?
Procedural	Reusable steps, handovers, checks, and gates	A fixed team forced onto every objective	Can the procedure be adapted and still show its owner?

Memory is useful when its purpose and scope are explicit. Recall does not grant authority.

The investigator working on the incident needs the current alert, dependency state, and open hypotheses. They may also need the last similar outage, the approved service runbook, and the procedure for requesting a rollback. Those are not one endless conversation. Each memory class has a scope, owner, retention decision, and access boundary.

Memory boundary exercises

- Give the diagnostic role a prior incident that is similar but not authoritative; verify that it remains a hypothesis.
- Withdraw a runbook version and verify which current and historical tasks may still reference it.
- Ask a replacement specialist to continue with working state but not private unrelated conversations.
- Promote one successful procedure and retain the owner, effective version, and review trigger.
- Delete or expire a memory class and verify the work record states what was lost.

THE ORGANISATION REMEMBERS HOW IT WORKS

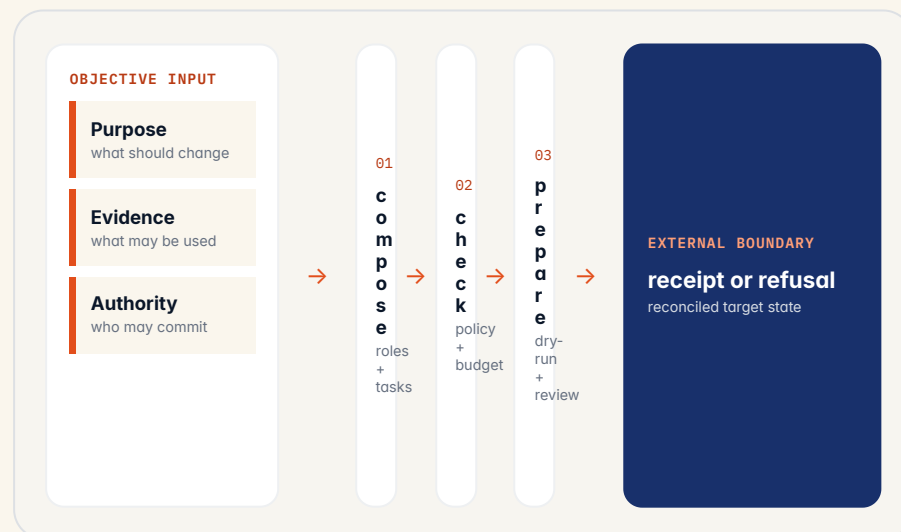
The team may end while the objective, result, memory decision, and operating record remain. That is reuse with provenance, not a cloned workforce.

02

AUTHORITY AND RESILIENCE

Let capability help without letting it self-authorise

Nexus coordinates work inside the authority set by the customer, policy owner, and human approver. It must make action limits, review, failure, and recovery observable.



Nexus may organise the route to an action. The action's authority remains outside the organisation's capability graph.

AUTHORITY AND RESILIENCE • RUNTIME GATES

Capability does not equal permission

The tool that can press the button is not automatically the role allowed to press it. Nexus evaluates policy, budget, capability, approval, and invariants at runtime.

BOUNDARY	NEXUS CAN ESTABLISH	STILL OWNED ELSEWHERE
Identity	Which participant, role, and task are active	Customer identity policy and account ownership
Capability	Which tool, source, model, or action the role may request	The target system's final authentication and effect
Policy	Which gate, budget, invariant, or approval must pass	The organisation's lawful or business policy meaning
Review	What briefing and evidence a person receives	The person's decision and consequence acceptance
Receipt	Which response and state are attached to the task	Whether the external world has fully reconciled

Runtime enforcement makes the boundary concrete. It does not transfer the customer's authority into Nexus.

A tool agent may read an incident dashboard and prepare a rollback request. It may not receive permission to mutate production merely because the requested change looks obvious. A reviewer may evaluate the evidence without receiving the tool capability. The incident lead may approve the change while the platform still refuses it because the target state, budget, or maintenance window is invalid.

The invariant watch

- No duplicate assignment of the same side-effecting task.
- No uncovered task type silently dropped because no qualified owner exists.
- No overcommitment or starvation hidden behind a healthy-looking queue.
- No approval represented by a model message or a tool return.
- No external effect treated as complete before the target receipt is reconciled.

POLICY IS NOT PROMPT DECORATION

A system instruction can request restraint. A runtime gate can reject the action, record why, and leave the task in a state that another owner can inspect.

AUTHORITY AND RESILIENCE • HUMAN GATE

A reviewer needs a briefing, not a private model meeting

The approval surface should make the proposed action, evidence, policy, expected effect, reversibility, and handover point clear before asking for a decision.

TRANSCRIPT	NEXUS BRIEFING
A pile of activity The reviewer reconstructs the decision - many messages - sources mixed with guesses - approval buried in context - effect and reversibility unclear	A decision packet The reviewer sees the boundary - proposed action stated directly - relevant evidence and policy - named handover and authority - expected effect and rollback path

The organisation can keep details available while presenting the human gate at the level of the consequence.

For the payment incident, the reviewer does not need every internal message. They need to know what change Nexus proposes, which telemetry and configuration support it, which policy routes the decision to them, what could change, whether rollback is available, and where responsibility moves after approval. A concise briefing is more accountable than a long transcript that hides the decisive facts.

Approval questions

- What exactly would change if I approve this?
- Which source, observation, or policy made this the proposed next step?
- What did the organisation not establish?
- What can be reversed, and who owns reversal?
- What receipt will close the action or reopen the task?

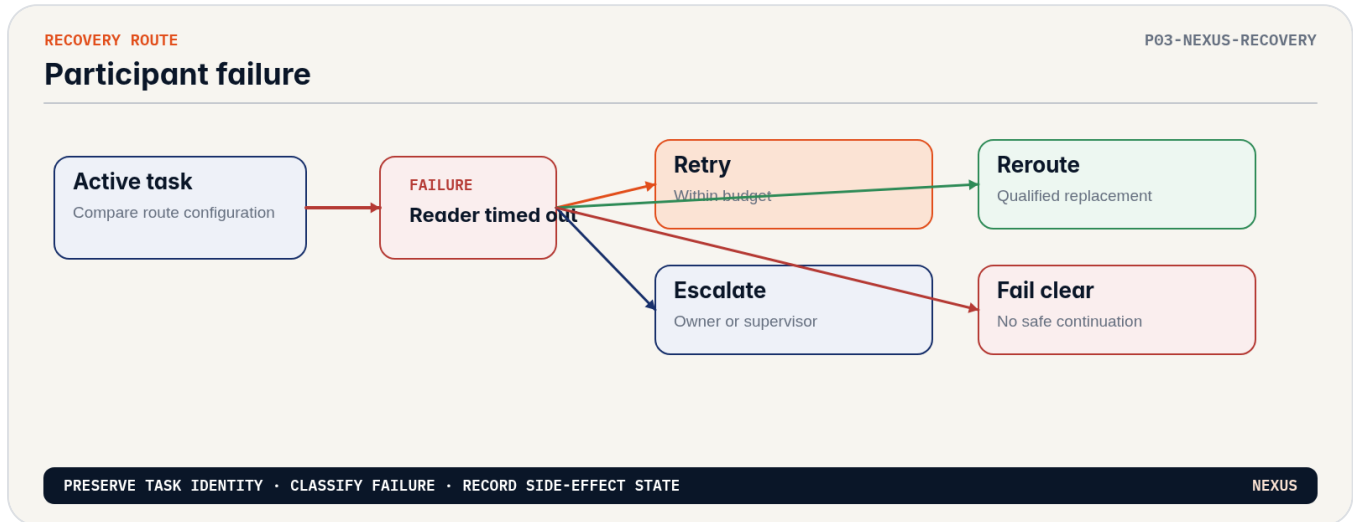
HUMAN AUTHORITY STAYS HUMAN

Nexus may prepare, compare, and explain. It does not turn an agent's confidence into an approval or conceal that an approval is still missing.

AUTHORITY AND RESILIENCE • RECOVERY

A failed helper should not erase the task

Services time out, participants become unhealthy, reviewers go offline, and resources run short. Recovery should be chosen from bounded responses, with the task state intact.



The response is a state transition, not an improvised sentence about what may have happened.

Suppose the telemetry reader loses access after identifying the affected route. Nexus can retry within a limit, assign the remaining work to another qualified participant, escalate to a supervisor or person, or fail clearly. The next participant receives the useful state and the original failure classification. It does not get a new task that quietly forgets the evidence already gathered or repeats a side effect.

Recovery record

- Original failure: what failed, when, and how it was classified.
- Recovery choice: retry, reroute, escalate, or controlled failure.
- Replacement state: who received the work and what context travelled.
- Eventual result: how the task and objective finally ended.
- Side-effect check: whether an external action was attempted, accepted, or left uncertain.

NO FAILURE DISGUISE

A recovery that returns fluent text while dropping the missing source, approval, or target receipt is not resilience. It is a new uncertainty hidden in the old task.

AUTHORITY AND RESILIENCE • RECORD

The work record shows observable operations, not hidden thoughts

A dependable receipt answers which roles, sources, policies, tools, approvals, failures, and outcomes mattered. It does not pretend private chain-of-thought is a reliable audit artefact.

NEXUS WORK RECORD • INC-204		What a reviewer can reconstruct	
01	OBJECTIVE AND STAFFING why this organisation and these roles existed	Nexus	formed
02	TASK TRANSITIONS who owned each responsibility and when state changed	scheduler	ordered
03	SOURCES AND TOOLS what was retrieved, called, returned, or denied	participants	linked
04	POLICY AND APPROVALS which rules allowed or stopped the route	governance	gated
05	RECOVERY DECISIONS how failures and replacements were handled	resilience	retained
06	OUTCOME AND RECEIPT what the operation changed and how it closed	target + owner	reconciled

The record is a receipt of work and authority. It is not a performance of certainty.

Record only what can carry responsibility

Keep the objective, team formation, role and capability assignments, task transitions, artefact contracts, collaboration messages that matter to the merge, policy and human gates, tool requests and results, recovery choices, final merge, dissolution, and work receipt. The record can be compact for ordinary reading and detailed on demand.

- Can a reviewer identify the source path without reading internal chatter?
- Can the affected owner see which action changed their system?
- Can operations distinguish a missing receipt from a failed action?
- Can a later team tell which evidence was current at the time?
- Can the record show where Nexus stopped and a customer system or human took over?

KEEP THE RECEIPT • REPLAY

Replay the boundary, not an imaginary unchanged world

Nexus can make internal ordering, routing, policy, retry, and accounting decisions reproducible while recording external calls and their versions at the boundary.

REPLAY AREA	PIN OR CAPTURE	MEANING OF A DIFFERENCE
Organisation state	task graph, role set, policy, ordering, retries	local runtime drift or an explicit new decision
Participant call	endpoint, version, input, output, position	external result changed or cannot be reproduced
Source context	document/version, authority, effective date	knowledge changed or lineage is incomplete
Human gate	briefing, decision, actor, timestamp, reason	authority chose a different consequence
Target receipt	request, response, resulting state, reconciliation	the world accepted, rejected, or left the effect uncertain

Replay is strongest when it tells the reviewer exactly which boundary moved.

The organisation's local state can be pinned: task ordering, workflow structure, policy version, role assignments, scoring, retry decisions, and lifecycle transitions. The world outside can change: a service may return a different status, a source may be withdrawn, a human may choose another option, or a deployment may have moved. A useful replay records those differences instead of claiming the external world stayed still.

Replay tests

- Re-run a read-only incident route with fixed task and policy state.
- Change one source version and verify the divergence points to context rather than staffing.
- Replay a timeout and verify the retry budget prevents duplicate external work.
- Compare a live tool result with a captured result without implying that capture equals a new action.
- Export enough state for an independent reviewer to understand what was pinned and what was observed.

REPRODUCIBILITY HAS A BOUNDARY

A deterministic local route can still end in a different customer decision when the external system or human authority is different. That difference belongs in the receipt.

03

PROVE AND OPERATE

Make the organisation earn reliance

Nexus is ready for a real process when its team formation, task state, authority, recovery, evidence, deployment, and exit conditions can all be tested together.

NEXUS PROOF · OBJECTIVE TO OUTCOME		The operation closes only when the receipt closes	
01	OBJECTIVE purpose and success condition are owned	business owner	bound
02	ORGANISATION roles and collaboration shape fit the work	Nexus	formed
03	AUTHORITY policies, budgets, and human gates hold	governance	checked
04	RECOVERY failure does not become an unowned action	operations	tested
05	RECEIPT outcome and external state reconcile	target owner	closed

The proof follows responsibility through the system rather than counting agent messages.

PROVE AND OPERATE • STACK

Nexus sits between the objective and the systems that matter

Fabric gives people the work surface. Loom can provide cognition. Spindle governs knowledge. Mesh supplies distributed execution. Nexus composes the organisation that turns those capabilities into an operation.



Nexus coordinates capabilities across the bridge between an objective and the systems that matter, without absorbing their contracts.

The boundary is practical. If Loom returns a cognitive result, Nexus decides which task owns it, who may challenge it, and whether it is enough to move forward. If Aura prepares a repository change, Nexus retains the task and review state while the repository and CI remain authoritative for the actual change. If Spindle supplies a source atom, Nexus records its use without becoming the source owner.

DELETION TEST

Remove Nexus from the design and name the missing organisation state: objective ownership, role formation, task recovery, authority gates, collaboration structure, and closed receipt. Those responsibilities should not disappear into a generic assistant.

PROVE AND OPERATE • ACCEPTANCE CORPUS

Acceptance begins with ordinary work and ends at the boundary cases

A convincing demo shows a team completing a happy path. A production decision also needs disagreement, missing authority, failed helpers, duplicate prevention, source drift, and an external receipt.

CORPUS SLICE	RUN	EXPECTED OBSERVATION
Small objective	One read-only question with one qualified role	The smallest team forms and closes without unnecessary participants
Multi-role objective	Diagnosis, preparation, review, and communication	Task ownership, handovers, and merge remain visible
Disagreement	Two independent findings conflict	Dissent stays attached and the configured authority decides
Authority boundary	A tool can prepare but not commit	The action stops for approval and records the gate
Helper failure	Timeout, unhealthy agent, missing reviewer	Retry, reroute, escalation, or named failure preserves task state
Source drift	Runbook or telemetry version changes	Lineage and replay identify the changed context
Duplicate risk	Retry after an uncertain target response	Idempotency or reconciliation prevents a second side effect
Closure	Target returns a receipt and owner reviews it	External state and Nexus outcome reconcile before close

The acceptance corpus tests responsibility under pressure, not only whether several agents can produce text.

Measure the organisation

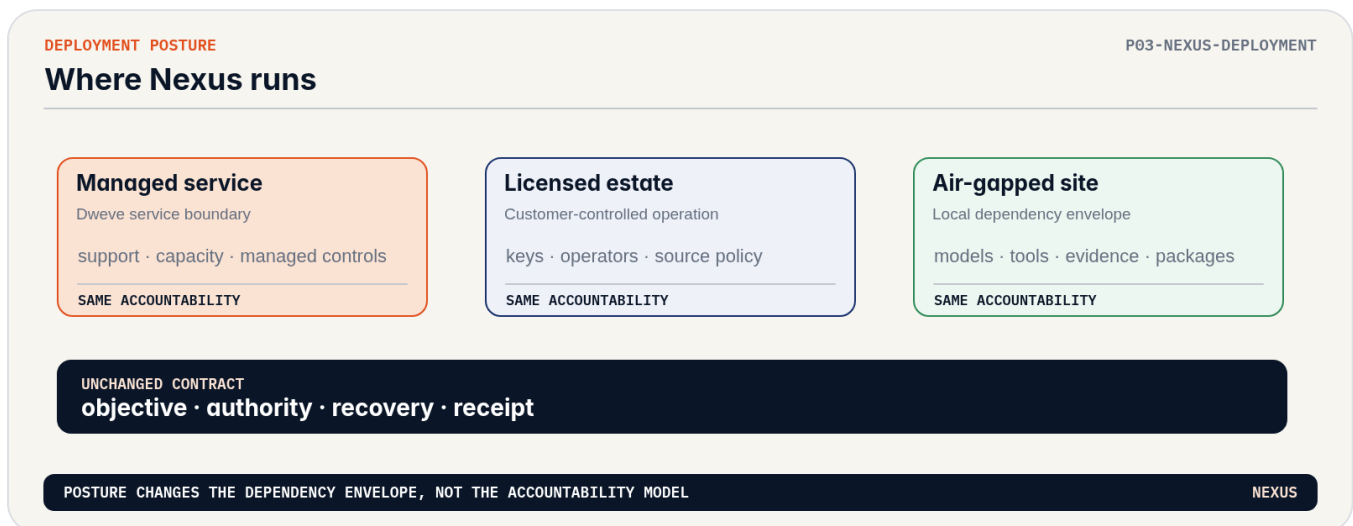
- Time to form an appropriate team, not simply number of workers invoked.
- Task latency, queue health, retry, reroute, blocked state, and escalation.
- Policy violations, approval waits, duplicate prevention, and action reversibility.
- Evidence completeness, dissent retention, source lineage, and target reconciliation.
- Resource use, service level, restore time, export quality, and operator workload.

Record raw objectives, candidate roles, assignments, task transitions, tool calls, approvals, receipts, failed assertions, reviewer decisions, and remaining gaps. A production proof should end in go, narrow, hold, or stop with an owner and review trigger.

PROVE AND OPERATE · DEPLOYMENT

Deployment changes where Nexus runs, not what makes it accountable

Managed service, licensed direct operation, and air-gapped deployment change the infrastructure and support boundary. They should not change task ownership, policy gates, recovery semantics, or the work record.



The posture changes the operator and dependency envelope. The objective, authority, and receipt model remain the same.

A connected deployment may use external models and services under explicit rules. A private deployment may keep models, tools, sources, and records inside customer-controlled infrastructure. An air-gapped operation may require local capability and a different support path. Those differences matter for egress, updates, capacity, incident response, retention, and export; they do not turn an agent into a business owner.

Deployment questions

- Which party owns identity, keys, capacity, patching, incident response, and data retention?
- Which participants and tools may cross the deployment boundary?
- How are models, role definitions, policies, source versions, and workflows promoted?
- Can the work record and task state be exported when the agreement or environment ends?
- What is the safe degraded mode when the managed or local dependency is unavailable?

POSTURE IS A RESPONSIBILITY MAP

A private installation can reduce egress. It cannot remove the customer's obligation to name who may approve a production action.

PROVE AND OPERATE • PRODUCTION RECORD

The production decision is a record, not a demo score

Adopt Nexus for one bounded process when the business owner, platform operator, source owner, security owner, and human approver can inspect the same operation from their different responsibilities.

NEXUS PRODUCTION RECORD		Service-incident organisation proof decision
OBJECTIVE	Contain, diagnose, restore, communicate, and retain lessons	BOUNDED
TEAM	Named roles, capabilities, collaboration shape, and fallback	FORMED
AUTHORITY	Policies, budgets, human gates, and target permissions	REVIEWED
RESILIENCE	Retry, reroute, escalation, failure, and duplicate protection	TESTED
EVIDENCE	Sources, tasks, actions, approvals, dissent, and receipts	LINKED
OPERATION	Capacity, incident, retention, export, support, and restore	OWNED
DECISION	Go / narrow / hold / stop with review trigger	SIGNED

The record should remain understandable after the original agents, models, operators, and integration team have changed.

The product owner decides the process boundary and success condition. The operations owner decides recovery and service limits. The source owner decides which material is authoritative. The security owner decides identity, access, egress, and tool posture. The human approver decides where consequence requires a person. The platform owner decides capacity, release, support, export, and exit. One product cannot silently merge these duties.

The smallest credible next step

Bring one incident or operational process to Nexus with a real objective, current source set, smallest useful team, explicit task contracts, one human gate, one failure injection, and a target receipt. Make the process owner, operator, security owner, source owner, and approver inspect the same packet. If each person must reconstruct a different truth, narrow the proof.

PRIMARY ACCEPTANCE CRITERION

Nexus can show why the organisation formed, who owned each task, which authority gate applied, how failure was handled, what changed outside the runtime, and why the objective was or was not closed.

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 3, 6–7, 11, 13, 20, 22	A Accountability 3, 6–7, 10, 14, 19, 21	A Air-gapped deployment 21
A Appeal and challenge 5, 9–10, 19	A Audit 16	A Aura 19
A Authority 3–7, 9–22	B Boundaries 3–4, 6, 11–14, 17, 19–22	C Capabilities 4–6, 8, 12–13, 16, 19, 21–22
C Configuration 8, 14	C Continuity 8	C Contracts 4, 6, 8
C Cost and budgeting 8, 12–13, 17	D Dependencies 3, 8–9, 11, 21	D Deployment 8, 17–18, 21
D Determinism 17	E Evidence 4–5, 7–8, 10, 12–16, 18, 20–22	E Exit 18, 22
F Fabric 19	F Failure 8–9, 12, 15, 18, 20, 22	G Governance 16, 18
H Human authority 3, 14, 17	I Identity 8, 13, 21–22	I Incident response 3–5, 7–8, 10–11, 13–14, 17, 21–22
I Inputs 12, 17	I Integration 22	L Ledger 22
L Limits and exclusions 3, 8, 12, 22	L Lineage 17, 20	L Loom 19
M Mesh 19	M Models and solvers 3, 6, 13–14, 21	N Nexus 3–7, 9–22
O Operations 3, 6, 10, 16, 18–22	O Outputs 17	O Ownership 3–4, 6–9, 11–13, 15–16, 18–22

SUBJECT INDEX CONTINUED

Subject index · P–W

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

P Performance 16	P Policies 6–9, 12–14, 16–18, 20–22	P Proof 18, 20, 22
P Provenance 11	R Recovery 3, 11–12, 15–16, 18–22	R Replay 17, 20
R Reproducibility 17	R Responsibility 3, 5–6, 8, 14, 16, 18–22	R Retention 11, 20–22
R Risk 4, 9–10, 20	S Security 5, 22	S Service levels 20
S Sources 3–6, 8, 11, 13–17, 19–22	S Sovereignty 4, 7, 10, 14, 17, 21	S Spindle 19
S State 3, 5–13, 15–21	T Testing 3, 19	T Tool authority 13
V Verification 6, 11, 17, 21	V Versioning 11, 17, 20	W Workflows 9, 17

CLOSING PRODUCT RECORD

Start with the process nobody wants to reconstruct by hand

Choose one objective that currently crosses people, systems, rules, and approvals. Make the organisation's work visible before asking it to scale.

Write the objective, acceptance condition, source authority, smallest useful team, role contracts, task graph, collaboration shape, memory scopes, policy gates, approval points, resource limits, recovery choices, record fields, deployment posture, retention, export, and stop conditions.

Then choose the hard case: a conflicting finding, unavailable participant, missing reviewer, stale source, duplicate-risk retry, denied tool, uncertain target response, resource limit, restore, or export. Nexus becomes useful when the difficult handover is concrete enough to run.

OBJECTIVE

One accountable outcome with an owner

The team forms around it

ORGANISATION

Roles, tasks, workflows, memory, and gates

Not a fixed cast of bots

AUTHORITY

Capability, permission, approval, and receipt

They are different boundaries

RESILIENCE

Retry, reroute, escalate, or fail clearly

The task keeps its identity

DECISION

Go, narrow, hold, or stop

Signed with review triggers

THE NEXUS STANDARD

A dependable executable organisation returns one accountable outcome while keeping roles, tasks, sources, policies, dissent, failures, approvals, and external receipts inspectable.