

FOUNDATION AND DECISION GUIDE · F07

Proof, trace, determinism, and replay

A precise evidence model for review, appeal, incidents, historical reconstruction, and independent verification.

DECISION

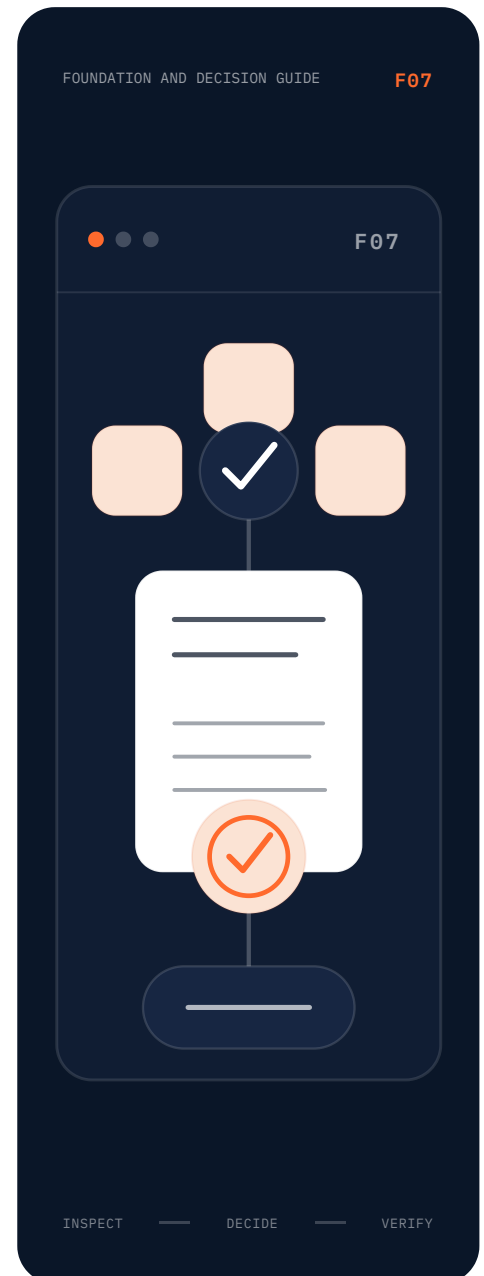
SCOPE

EVIDENCE

OWNERSHIP

PROOF

EXIT



A complete guide to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

F07

CONTENTS

From the first question to a defensible decision

This 45-page guide turns 15 owned topics into an explanation, evidence requirement, proof challenge, and closing decision record.

PRIMARY READERS

Assurance, audit and evidence teams

READING DEPTH

Specialist

DECISION THIS SUPPORTS

Determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

01

Read this first

03–05

Audience, reader decision, canonical boundary, terms, and guide-specific factual spotlight

02

Understand the decision and its language

06–...

Why an answer is not enough · Evidence vocabulary · The anatomy of a decision record · Evidence created during work · Deterministic computation

03

Define the mechanism, controls, and evidence

...

Version and source pinning · Hash chains, Merkle structures, and signatures · Decision reasoning versus system events · Replay pipeline · Offline and third-party verification

04

Prove adoption, operation, continuity, and exit

...–43

Human authority and appeal · Privacy and redaction · Three worked cases · What proof does not prove · Acceptance tests

05

Sources and next action

44–43

verification records and closing decision package

06

Subject index

44–45

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

TARGET LENGTH

11,000–15,000 words.. Page count varies with the number and depth of topics rather than forcing every subject into the same shell.

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.

READ THIS FIRST • READER DECISION

Who should use this guide, and what decision it supports

risk, audit, compliance, model governance, engineering, and oversight teams.

Determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

The worked scenario is an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. It is illustrative and deliberately bounded. Replace it with the organisation's actual workflow, systems, data, sources, policies, roles, infrastructure, commercial conditions and consequences while preserving the evidence discipline.

The guide's concrete output is a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict. Use the chapter explanation pages to align language, the evidence pages to create acceptance records, and any challenge pages to test whether the boundary survives missing assumptions and failure. End in a named decision, not an attractive summary.

TRACE

Decision path

Current guide lens; verify source and scope.

LEDGER

System events

Current guide lens; verify source and scope.

TRANSCRIPT

Execution boundary

Current guide lens; verify source and scope.

REPLAY

Scoped comparison

Current guide lens; verify source and scope.

PRIMARY RISK

Calling every log a trace or every signed record a proof creates false assurance. Historical material can also become unverifiable when schemas, keys, source versions, runtimes or rights disappear.

READ THIS FIRST • DISCIPLINE

How to read claims, evidence, responsibility, and status

The website is the only factual source for this edition. The guide adds explanation, architecture and proof requirements without inventing current product, regulatory, assurance, commercial, or customer-result facts.

Six evidence layers

A published description states a current proposition. Delivered software and documentation show what is available in a selected version. Customer configuration shows how it is used. An observed run shows one result. Independent assurance has scope, method and date. A written agreement creates commercial commitments and remedies.

Responsibility stays explicit

The relevant owners include domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier. The supplier can provide product, evidence and commitments; customer owners still decide purpose, source and policy authority, professional or domain judgement, acceptable risk, infrastructure, staffing, affected-person treatment and production reliance.

Status and strong statements

Measured claims stay tied to workload, hardware and method. Research remains research. Alignment is not certification. "Open" is verified per component and licence. Pricing is re-audited at release. Roadmap and availability are not assumed. Conflicts or ambiguity become diligence items.

LAYER	EVIDENCE	CANNOT ESTABLISH ALONE
Website	website copy + captured date	Delivery, fit, contract
Delivery	Manifest + docs + artefacts	Customer result
Configuration	Data/roles/policy/posture	Operating outcome
Run	Raw output + target + packet	General performance
Assurance	Scoped test/report + findings	Universal security/compliance
Agreement	Rights + duties + remedy	Technical behaviour without test

Keep the layers separate in every chapter and decision record.

READING RULE

Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal.

READ THIS FIRST • CANONICAL SPOTLIGHT

The evidence vocabulary must stay separated

One packet can link several evidence types without pretending that each proves the same proposition.

A source record identifies origin and version. A rule record identifies the policy or constraint applied. A decision trace links material steps and state. An event ledger records typed system history. An execution transcript records what a constrained runtime did. A receipt records authority or external effect. A certificate or proof object supports a declared proposition. A hash, Merkle root or signature supports integrity and trust-policy checks.

Replay starts after retrieval and verification. It reconstructs the original material, then recomputes or simulates only the supported scope. A verifier can work without the supplier when schemas, trust material, referenced artefacts and executable dependencies remain available. The verdict must say exact, equivalent within a declared tolerance, divergent, incomplete or unverifiable.

A person-facing explanation is another view of the same decision record: result, material fact or source/rule, responsible human or mandate, and next action. It may redact sensitive detail, but its core facts must agree with the reviewer view, target state and retained packet.

AREA	CURRENT GUIDE RECORD	VERIFY / DECIDE
Provenance	Where material came from	Identity, authority, version, transform
Trace	How a decision progressed	Steps, states, versions, links
Ledger	What the system recorded	Typed ordered events and integrity
Transcript	What execution did	Capabilities, calls, result, resource state
Receipt	Who approved or what target applied	Authority or external correlation
Replay verdict	What can be reproduced now	Scope, comparison, limits, divergence

Current source-edition summary. Recheck publication-sensitive facts and delivered scope at the decision date.

GUIDE-SPECIFIC PROOF

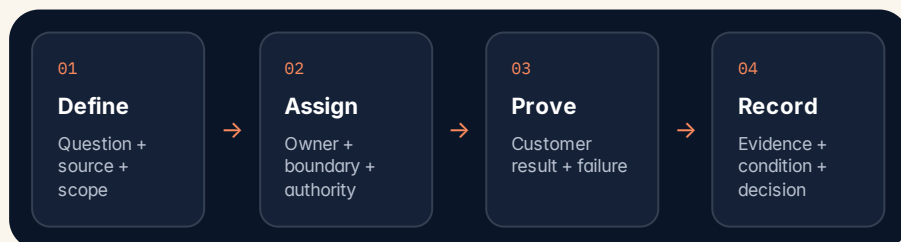
Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes.

01

PROOF, TRACE, DETERMINISM, AND REPLAY

Understand the decision and its language

Define the real question, current state, responsibilities, source claims, and the artefact this guide must produce.



The same evidence discipline applies to every chapter in the section.

CHAPTER 01 • EXPLAIN

Why an answer is not enough

the operational, supervisory, incident, and appeal gaps left by a score or chat log.

This chapter owns a specific part of the reader's decision: the operational, supervisory, incident, and appeal gaps left by a score or chat log.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Why an answer is not enough", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

The operational

Define the exact proposition and current authoritative source.

LENS 2

Supervisory

Name the responsible person, system, dependency, and version.

LENS 3

Incident

Specify the evidence and how an independent reviewer checks it.

LENS 4

And appeal gaps left by a score or chat log

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 01 • APPLY

Turn “Why an answer is not enough” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, the operational, supervisory, incident, and appeal gaps left by a score or chat log., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for why an answer is not enough; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 01 • CHALLENGE

Stress-test the boundary behind “Why an answer is not enough”

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: Calling every log a trace or every signed record a proof creates false assurance. Historical material can also become unverifiable when schemas, keys, source versions, runtimes or rights disappear.

For this chapter, remove or alter the strongest assumption in “the operational, supervisory, incident, and appeal gaps left by a score or chat log.”. Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

STRESS LANE	INJECTION	EXPECTED EVIDENCE
Authority	Role absent, revoked, conflicted or bypassed	Wait/refuse/escalate + receipt
Dependency	Source, service, package, model or key unavailable	Degraded state + manifest + recovery
Integrity	Alter, delete, reorder, substitute	Verification failure + quarantine
Operation	Load, outage, clock, storage or update fault	Alert + safe state + restore
Target	Timeout, duplicate, rejection or conflicting state	Idempotency + reconciliation
Exit	Supplier access removed or contract ends	Export + local verification + continuity result

Select applicable lanes and add topic-specific consequences before the run.

STOP RULE

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 02 • EXPLAIN

Evidence vocabulary

define source, rule, model, owner, action, event, trace, receipt, transcript, certificate, signature, root, verifier, and replay.

This chapter owns a specific part of the reader's decision: define source, rule, model, owner, action, event, trace, receipt, transcript, certificate, signature, root, verifier, and replay.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Evidence vocabulary", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Define source

Define the exact proposition and current authoritative source.

LENS 2

Rule

Name the responsible person, system, dependency, and version.

LENS 3

Model

Specify the evidence and how an independent reviewer checks it.

LENS 4

Owner

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 02 • APPLY

Turn “Evidence vocabulary” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, define source, rule, model, owner, action, event, trace, receipt, transcript, certificate, signature, root, verifier, and replay., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for evidence vocabulary; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 03 • EXPLAIN

The anatomy of a decision record

fields, identities, versions, timestamps, state, approvals, and links.

This chapter owns a specific part of the reader's decision: fields, identities, versions, timestamps, state, approvals, and links.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "The anatomy of a decision record", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Fields

Define the exact proposition and current authoritative source.

LENS 2

Identities

Name the responsible person, system, dependency, and version.

LENS 3

Versions

Specify the evidence and how an independent reviewer checks it.

LENS 4

Timestamps

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 03 • APPLY

Turn “The anatomy of a decision record” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, fields, identities, versions, timestamps, state, approvals, and links., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant customer configuration, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for the anatomy of a decision record; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 04 • EXPLAIN

Evidence created during work

show when records attach and why after-the-fact reconstruction is weaker.

This chapter owns a specific part of the reader's decision: show when records attach and why after-the-fact reconstruction is weaker.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Evidence created during work", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

When records attach

Define the exact proposition and current authoritative source.

LENS 2

Why after-the-fact reconstruction is weaker

Name the responsible person, system, dependency, and version.

LENS 3

Scope and limits

Specify the evidence and how an independent reviewer checks it.

LENS 4

Owner and evidence

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 04 • APPLY

Turn “Evidence created during work” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, show when records attach and why after-the-fact reconstruction is weaker., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant observed operation, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for evidence created during work; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 04 • CHALLENGE

Stress-test the boundary behind "Evidence created during work"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: Calling every log a trace or every signed record a proof creates false assurance. Historical material can also become unverifiable when schemas, keys, source versions, runtimes or rights disappear.

For this chapter, remove or alter the strongest assumption in "show when records attach and why after-the-fact reconstruction is weaker.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

STRESS LANE	INJECTION	EXPECTED EVIDENCE
Authority	Role absent, revoked, conflicted or bypassed	Wait/refuse/escalate + receipt
Dependency	Source, service, package, model or key unavailable	Degraded state + manifest + recovery
Integrity	Alter, delete, reorder, substitute	Verification failure + quarantine
Operation	Load, outage, clock, storage or update fault	Alert + safe state + restore
Target	Timeout, duplicate, rejection or conflicting state	Idempotency + reconciliation
Exit	Supplier access removed or contract ends	Export + local verification + continuity result

Select applicable lanes and add topic-specific consequences before the run.

STOP RULE

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 05 • EXPLAIN

Deterministic computation

same inputs, artefacts, versions, and supported target; explain the guarantee and its preconditions.

This chapter owns a specific part of the reader's decision: same inputs, artefacts, versions, and supported target; explain the guarantee and its preconditions.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Deterministic computation", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Same inputs

Define the exact proposition and current authoritative source.

LENS 2

Artefacts

Name the responsible person, system, dependency, and version.

LENS 3

Versions

Specify the evidence and how an independent reviewer checks it.

LENS 4

And supported target

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 05 • APPLY

Turn “Deterministic computation” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, same inputs, artefacts, versions, and supported target; explain the guarantee and its preconditions., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant contract and assurance, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for deterministic computation; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

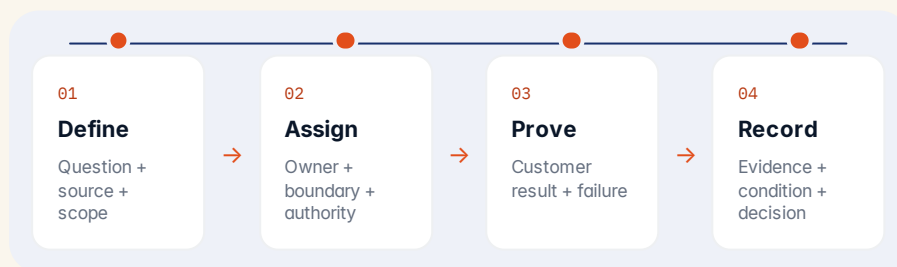
Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

02

PROOF, TRACE, DETERMINISM, AND REPLAY

Define the mechanism, controls, and evidence

Turn architecture and policy language into versions, owners, boundaries, tests, records, failures, and explicit limitations.



The same evidence discipline applies to every chapter in the section.

CHAPTER 06 • EXPLAIN

Version and source pinning

policies, data, models, code, numeric profiles, tools, and deployment context.

This chapter owns a specific part of the reader's decision: policies, data, models, code, numeric profiles, tools, and deployment context.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Version and source pinning", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Policies

Define the exact proposition and current authoritative source.

LENS 2

Data

Name the responsible person, system, dependency, and version.

LENS 3

Models

Specify the evidence and how an independent reviewer checks it.

LENS 4

Code

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 06 • APPLY

Turn “Version and source pinning” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, policies, data, models, code, numeric profiles, tools, and deployment context., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for version and source pinning; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 07 • EXPLAIN

Hash chains, Merkle structures, and signatures

explain tamper evidence and portable verification without implying that integrity proves correctness.

This chapter owns a specific part of the reader's decision: explain tamper evidence and portable verification without implying that integrity proves correctness.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Hash chains, Merkle structures, and signatures", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Tamper evidence

Define the exact proposition and current authoritative source.

LENS 2

Portable verification without implying that integrity proves correctness

Name the responsible person, system, dependency, and version.

LENS 3

Scope and limits

Specify the evidence and how an independent reviewer checks it.

LENS 4

Owner and evidence

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 07 • APPLY

Turn “Hash chains, Merkle structures, and signatures” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, explain tamper evidence and portable verification without implying that integrity proves correctness., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for hash chains, merkle structures, and signatures; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 07 • CHALLENGE

Stress-test the boundary behind “Hash chains, Merkle structures, and signatures”

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: Calling every log a trace or every signed record a proof creates false assurance. Historical material can also become unverifiable when schemas, keys, source versions, runtimes or rights disappear.

For this chapter, remove or alter the strongest assumption in “explain tamper evidence and portable verification without implying that integrity proves correctness.”. Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

STRESS LANE	INJECTION	EXPECTED EVIDENCE
Authority	Role absent, revoked, conflicted or bypassed	Wait/refuse/escalate + receipt
Dependency	Source, service, package, model or key unavailable	Degraded state + manifest + recovery
Integrity	Alter, delete, reorder, substitute	Verification failure + quarantine
Operation	Load, outage, clock, storage or update fault	Alert + safe state + restore
Target	Timeout, duplicate, rejection or conflicting state	Idempotency + reconciliation
Exit	Supplier access removed or contract ends	Export + local verification + continuity result

Select applicable lanes and add topic-specific consequences before the run.

STOP RULE

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 08 • EXPLAIN

Decision reasoning versus system events

separate AION certificates, Knot/Ledger event provenance, Selvedge execution transcripts, and product work records.

This chapter owns a specific part of the reader's decision: separate AION certificates, Knot/Ledger event provenance, Selvedge execution transcripts, and product work records.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Decision reasoning versus system events", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Separate AION certificates

Define the exact proposition and current authoritative source.

LENS 2

Knot/Ledger event provenance

Name the responsible person, system, dependency, and version.

LENS 3

Selvedge execution transcripts

Specify the evidence and how an independent reviewer checks it.

LENS 4

And product work records

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 08 • APPLY

Turn “Decision reasoning versus system events” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, separate AION certificates, Knot/Ledger event provenance, Selvedge execution transcripts, and product work records., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant customer configuration, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for decision reasoning versus system events; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 09 • EXPLAIN

Replay pipeline

retrieve, validate, reconstruct, execute or verify, compare, and record the review result.

This chapter owns a specific part of the reader's decision: retrieve, validate, reconstruct, execute or verify, compare, and record the review result.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Replay pipeline", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Retrieve

Define the exact proposition and current authoritative source.

LENS 2

Validate

Name the responsible person, system, dependency, and version.

LENS 3

Reconstruct

Specify the evidence and how an independent reviewer checks it.

LENS 4

Execute or verify

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 09 • APPLY

Turn “Replay pipeline” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, retrieve, validate, reconstruct, execute or verify, compare, and record the review result., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant observed operation, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for replay pipeline; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 10 • EXPLAIN

Offline and third-party verification

public keys, standard formats, isolated archives, and supplier-independent checking as supported by the site.

This chapter owns a specific part of the reader's decision: public keys, standard formats, isolated archives, and supplier-independent checking as supported by the site.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Offline and third-party verification", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Public keys

Define the exact proposition and current authoritative source.

LENS 2

Standard formats

Name the responsible person, system, dependency, and version.

LENS 3

Isolated archives

Specify the evidence and how an independent reviewer checks it.

LENS 4

And supplier-independent checking as supported by the site

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 10 • APPLY

Turn “Offline and third-party verification” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, public keys, standard formats, isolated archives, and supplier-independent checking as supported by the site., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant contract and assurance, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for offline and third-party verification; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 10 • CHALLENGE

Stress-test the boundary behind "Offline and third-party verification"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: Calling every log a trace or every signed record a proof creates false assurance. Historical material can also become unverifiable when schemas, keys, source versions, runtimes or rights disappear.

For this chapter, remove or alter the strongest assumption in "public keys, standard formats, isolated archives, and supplier-independent checking as supported by the site.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

STRESS LANE	INJECTION	EXPECTED EVIDENCE
Authority	Role absent, revoked, conflicted or bypassed	Wait/refuse/escalate + receipt
Dependency	Source, service, package, model or key unavailable	Degraded state + manifest + recovery
Integrity	Alter, delete, reorder, substitute	Verification failure + quarantine
Operation	Load, outage, clock, storage or update fault	Alert + safe state + restore
Target	Timeout, duplicate, rejection or conflicting state	Idempotency + reconciliation
Exit	Supplier access removed or contract ends	Export + local verification + continuity result

Select applicable lanes and add topic-specific consequences before the run.

STOP RULE

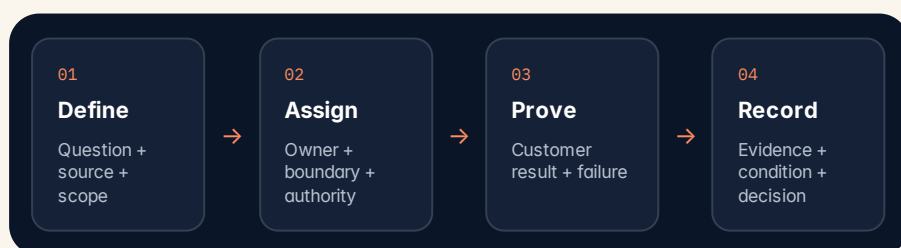
If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

03

PROOF, TRACE, DETERMINISM, AND REPLAY

Prove adoption, operation, continuity, and exit

Use customer evidence, challenge, recovery, diligence, commercial terms, and a signed decision before expanding reliance.



The same evidence discipline applies to every chapter in the section.

CHAPTER 11 • EXPLAIN

Human authority and appeal

reviewer identity, approval, override, refusal, affected-person explanation, and escalation.

This chapter owns a specific part of the reader's decision: reviewer identity, approval, override, refusal, affected-person explanation, and escalation.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Human authority and appeal", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Reviewer identity

Define the exact proposition and current authoritative source.

LENS 2

Approval

Name the responsible person, system, dependency, and version.

LENS 3

Override

Specify the evidence and how an independent reviewer checks it.

LENS 4

Refusal

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 11 • APPLY

Turn “Human authority and appeal” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, reviewer identity, approval, override, refusal, affected-person explanation, and escalation., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for human authority and appeal; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 12 • EXPLAIN

Privacy and redaction

what evidence may contain, supported masking, access control, and what a portable packet must not leak.

This chapter owns a specific part of the reader's decision: what evidence may contain, supported masking, access control, and what a portable packet must not leak.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Privacy and redaction", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

What evidence may contain

Define the exact proposition and current authoritative source.

LENS 2

Supported masking

Name the responsible person, system, dependency, and version.

LENS 3

Access control

Specify the evidence and how an independent reviewer checks it.

LENS 4

And what a portable packet must not leak

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 12 • APPLY

Turn “Privacy and redaction” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, what evidence may contain, supported masking, access control, and what a portable packet must not leak., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for privacy and redaction; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 13 • EXPLAIN

Three worked cases

regulated decision, agent incident, and cross-backend numeric replay.

This chapter owns a specific part of the reader's decision: regulated decision, agent incident, and cross-backend numeric replay.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Three worked cases", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Regulated decision

Define the exact proposition and current authoritative source.

LENS 2

Agent incident

Name the responsible person, system, dependency, and version.

LENS 3

And cross-backend numeric replay

Specify the evidence and how an independent reviewer checks it.

LENS 4

Scope and limits

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 13 • APPLY

Turn “Three worked cases” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, regulated decision, agent incident, and cross-backend numeric replay., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant customer configuration, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for three worked cases; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 13 • CHALLENGE

Stress-test the boundary behind “Three worked cases”

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: Calling every log a trace or every signed record a proof creates false assurance. Historical material can also become unverifiable when schemas, keys, source versions, runtimes or rights disappear.

For this chapter, remove or alter the strongest assumption in “regulated decision, agent incident, and cross-backend numeric replay.”. Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

STRESS LANE	INJECTION	EXPECTED EVIDENCE
Authority	Role absent, revoked, conflicted or bypassed	Wait/refuse/escalate + receipt
Dependency	Source, service, package, model or key unavailable	Degraded state + manifest + recovery
Integrity	Alter, delete, reorder, substitute	Verification failure + quarantine
Operation	Load, outage, clock, storage or update fault	Alert + safe state + restore
Target	Timeout, duplicate, rejection or conflicting state	Idempotency + reconciliation
Exit	Supplier access removed or contract ends	Export + local verification + continuity result

Select applicable lanes and add topic-specific consequences before the run.

STOP RULE

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 14 • EXPLAIN

What proof does not prove

bad source, bad policy, inappropriate goal, missing context, biased process, and lawful-basis questions remain possible.

This chapter owns a specific part of the reader's decision: bad source, bad policy, inappropriate goal, missing context, biased process, and lawful-basis questions remain possible.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "What proof does not prove", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

Bad source

Define the exact proposition and current authoritative source.

LENS 2

Bad policy

Name the responsible person, system, dependency, and version.

LENS 3

Inappropriate goal

Specify the evidence and how an independent reviewer checks it.

LENS 4

Missing context

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 14 • APPLY

Turn “What proof does not prove” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, bad source, bad policy, inappropriate goal, missing context, biased process, and lawful-basis questions remain possible., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant observed operation, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operations owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for what proof does not prove; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

CHAPTER 15 • EXPLAIN

Acceptance tests

a practical checklist for a customer to test trace completeness, cross-backend replay, offline verification, and reviewer usability.

This chapter owns a specific part of the reader's decision: a practical checklist for a customer to test trace completeness, cross-backend replay, offline verification, and reviewer usability.

In this guide, evidence is not one thing. Source provenance, a decision trace, system event, action receipt, execution transcript, certificate, signature, Merkle root, verifier result and replay verdict answer different questions and must retain their separate semantics. Applied to "Acceptance tests", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Integrity can show that committed bytes changed; provenance can identify origin; deterministic replay can compare a declared computation; a human receipt can show authority state. None alone proves source truth, policy legitimacy, lawful basis, model fitness, fairness, safety or a correct goal. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

A practical checklist for a customer to test trace completeness

Define the exact proposition and current authoritative source.

LENS 2

Cross-backend replay

Name the responsible person, system, dependency, and version.

LENS 3

Offline verification

Specify the evidence and how an independent reviewer checks it.

LENS 4

And reviewer usability

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.

CHAPTER 15 • APPLY

Turn “Acceptance tests” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, a practical checklist for a customer to test trace completeness, cross-backend replay, offline verification, and reviewer usability., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant contract and assurance, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: an auditor retrieves an old consequential decision, verifies the packet offline, reconstructs the exact source/rule/model/numeric manifest, simulates external effects, compares the declared deterministic path, and records limits and divergence. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

QUESTION	REQUIRED RECORD	OWNER / STOP
What is claimed?	Exact wording, scope, date, source	Guide decision owner
What exists?	Delivered manifest, interface, artefact	Technical/product owner
What is configured?	Customer data, roles, rules, posture	Customer control owner
What happened?	Raw run, target state, packet, metric	Domain/operation s owner
What is assured?	Test/report scope, findings, limits	Independent challenger
What is committed?	Offer/agreement, remedy, exit term	Procurement/legal owner

Evidence ladder for acceptance tests; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.

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, 8, 11, 13, 15, 18, 21, 23, 26, 28, 30, 34, 36, 38, 41-43	A Accountability 8-9, 11, 13, 15-16, 18, 21, 23-24, 26, 28, 30-31, 34, 36, 38-39, 41, 43	A Adoption 32
A AION 25-26	A Appeal and challenge 3, 7-9, 11, 13, 15-16, 18, 21, 23-24, 26, 28, 30-34, 36, 38-39, 41, 43	A Architecture 4, 19
A Assurance 3-4, 7, 9-10, 12, 14, 16-18, 20, 22, 24-25, 27, 29-31, 33, 35, 37, 39-40, 42-43	A Audit 3-4, 9, 16, 24, 31, 39	A Authority 3-43
A Availability 4	B Boundaries 3, 6-7, 9-10, 12, 14, 16-17, 19-20, 22, 24-25, 27, 29, 31-33, 35, 37, 39-40, 42	C Capabilities 5
C Compliance 3-4	C Configuration 4, 7, 10, 12-14, 17, 20, 22, 25-27, 29, 33, 35, 37-38, 40, 42	C Continuity 9, 16, 24, 31-32, 39
C Contracts 4, 7, 9-10, 12, 14, 16-18, 20, 22, 24-25, 27, 29-31, 33, 35, 37, 39-40, 42-43	D Decision records 4-5, 8, 11-13, 15, 18, 21, 23, 26, 28, 30, 34, 36, 38, 41, 43	D Dependencies 3, 5, 7-18, 20-31, 33-43
D Deployment 8-9, 11, 13, 15-16, 18, 20-21, 23-24, 26, 28, 30-31, 34, 36, 38-39, 41, 43	D Determinism 3-4, 6-8, 10-15, 17-23, 25-30, 32-38, 40-43	D Due diligence 4, 32
E Evidence 3-43	E Exit 8-9, 11, 13, 15-16, 18, 21, 23-24, 26, 28, 30-32, 34, 36, 38-39, 41, 43	F Failure 3, 6-7, 9-10, 12, 14, 16-17, 19-20, 22, 24-25, 27, 29, 31-33, 35, 37, 39-40, 42
G Governance 3-4, 9, 16, 24, 31, 39	H Hardware 4	H Human authority 8, 11, 13, 15, 18, 21, 23, 26, 28, 30, 33-34, 36, 38, 41, 43
I Identity 5, 33-34	I Incident response 3, 7-9, 11, 13, 15, 18, 21, 23, 26, 28, 30, 34, 36-39, 41, 43	I Inputs 7, 10, 12, 14, 17, 20, 22, 25, 27, 29, 33, 35, 37, 40, 42
K Keys and signing 5, 7, 10-12, 14, 17, 20, 22, 25, 27, 29, 33, 35, 37, 40, 42	K Knot 25-26	L Ledger 3, 5, 25-26
L Licensing 4	L Limits and exclusions 3, 5, 8-9, 11, 13-16, 18, 21-24, 26, 28, 30-31, 34, 36-39, 41, 43	M Manifests 3-4, 8-9, 11, 13, 15-16, 18, 21, 23-24, 26, 28, 30-31, 34, 36, 38-39, 41, 43

SUBJECT INDEX CONTINUED

Subject index · M–W

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

M Models and solvers 3–5, 7–18, 20–31, 33–43	O Operations 8–9, 11, 13, 15–16, 18, 21, 23–24, 26, 28, 30–32, 34, 36, 38–39, 41, 43	O Outputs 3–4
O Ownership 4, 6–43	P Performance 4	P Policies 3–5, 7, 10, 12, 14, 17, 19–22, 25, 27, 29, 33, 35, 37, 40–42
P Pricing 4	P Privacy 4, 9, 16, 24, 31, 35–36, 39	P Procurement 8, 11, 13, 15, 18, 21, 23, 26, 28, 30, 34, 36, 38, 41, 43
P Proof 3–6, 8–9, 11, 13, 15–16, 18–19, 21, 23–24, 26, 28, 30–32, 34, 36, 38–41, 43	P Provenance 4–5, 7, 10, 12, 14, 17, 20, 22, 25–27, 29, 33, 35, 37, 40, 42	R Recovery 5, 8–9, 11, 13, 15–16, 18, 21, 23–24, 26, 28, 30–32, 34, 36, 38–39, 41, 43
R Replay 3–8, 10–15, 17–23, 25–30, 32–43	R Responsibility 4, 6	R Rights 3–4, 9, 16, 24, 31, 39
R Risk 3–4, 8–9, 11, 13, 15–16, 18, 21, 23–24, 26, 28, 30–31, 34, 36, 38–39, 41, 43	R Rules 3–5, 8–11, 13, 15–16, 18, 21, 23–24, 26, 28, 30–31, 34, 36, 38–39, 41, 43	S Security 4, 9, 16, 24, 31, 39
S Selvedge 25–26	S Sources 3–43	S Sovereignty 4–5, 7–8, 10–15, 17–18, 20–23, 25–30, 33–38, 40–43
S State 3–18, 20–31, 33–43	T Testing 3–5, 8–9, 11, 13, 15–16, 18, 21, 23–24, 26, 28, 30–31, 34, 36, 38–39, 41–43	T Trace 3, 5–7, 9–12, 14, 16–17, 19–20, 22, 24–25, 27, 29, 31–33, 35, 37, 39–40, 42–43
V Verification 3, 5, 8–9, 11, 13, 15–16, 18, 21–24, 26–31, 34, 36, 38–39, 41–43	V Versioning 4–5, 7–10, 12, 14, 16–17, 20–22, 24–25, 27, 29, 31, 33–35, 37, 39–40, 42	W Workflows 3

CLOSING DECISION RECORD

Close with one evidence-backed, reversible decision

The guide is complete when its reader can decide whether to proceed, narrow, remediate, contract a condition, hold, reject, or exit, and another person can understand why.

Assemble a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict. State the question, scope and exclusions, scenario or workflow, current baseline, canonical source snapshot, delivered manifest, customer configuration, corpus and method, results and failures, owner decisions, risks, operational and commercial conditions, evidence links, remedies, stop/exit and review triggers.

Run the final discipline: Create a packet during real-shaped work, alter, delete and reorder material, change a source/model/target, replay on supported environments, verify offline with separate trust material, test redacted views, restore old dependencies, and ask an independent reviewer what each artefact establishes.

Have domain and decision owner, evidence/records owner, engineering, model/method governance, security and keys, human reviewer, target-system owner, privacy/disclosure, audit and independent verifier sign only the boundaries they own. Record pass, conditional pass, gap, fail or unverifiable per proposition. Preserve dissent and raw evidence. A brochure can organise the decision; it cannot make the evidence true or take responsibility from the people who rely on it.

DECISION RECORD	proof-trace-determinism-replay-decision	
QUESTION	determine whether Dweve's evidence model can answer the organisation's actual review, appeal, incident, and independent-verification questions.	BOUNDED
SOURCES	website copy + delivered material	CAPTURED
PROOF	corpus + failures + recovery	RUN
OWNERS	domain + technical + assurance	SIGNED
CONDITIONS	gaps + remedies + contract	DATED
DECISION	go / narrow / hold / stop / exit	RECORDED
The closing record remains useful after the presentation, release, team, supplier, or contract changes.		

GUIDE OUTPUT

Produce a portable decision or work packet with resolvable identities, versions, state, authority, target receipts, integrity, disclosure views, verifier dependencies, and a scoped replay verdict.