

FOUNDATION AND DECISION GUIDE · F05

Product portfolio and responsibility boundaries

Choose the smallest coherent Dweve product set by responsibility, state, interface, and adoption path.

DECISION

SCOPE

EVIDENCE

OWNERSHIP

PROOF

EXIT



A complete guide to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

F05

CONTENTS

From the first question to a defensible decision

This 36-page guide turns 13 owned topics into an explanation, evidence requirement, proof challenge, and closing decision record.

PRIMARY READERS

Buyers, product owners and architects

READING DEPTH

Professional + technical

DECISION THIS SUPPORTS

Select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

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–...
How to read the portfolio · Fabric: the workspace · Loom: cognition · Nexus: organised agent work · Spindle: knowledge processing and lineage

03 Define the mechanism, controls, and evidence ...
Aura: governed development · Mesh: distributed capacity and learning · Charter: the owned company record · Core: binary AI framework · Kera: language and compiler

04 Prove adoption, operation, continuity, and exit ...–34
Common combinations · Selection worksheet · Availability and commercial boundary

05 Sources and next action 35–34
verification records and closing decision package

06 Subject index 35–36
An alphabetical finder for concepts, controls, products, and decisions.

TARGET LENGTH

9,000–12,000 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

buyers, solution architects, product owners, and portfolio managers trying to determine which parts of Dweve they need.

Select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

The worked scenario is a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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 responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof. 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.

PUBLISHED PRODUCT
PAGES**the current
product set**Current guide lens; verify
source and scope.

SELECTION UNIT

ResponsibilityCurrent guide lens; verify
source and scope.

COMPOSITION

**Typed
contracts**Current guide lens; verify
source and scope.

MINIMUM TEST

**Deletion
boundary**Current guide lens; verify
source and scope.

PRIMARY RISK

Personal and Family include the same core Fabric features. The compute-contribution choice preserves the Personal feature set. The applicable terms state the processing boundary and cancellation conditions.

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 business and workflow owner, enterprise and solution architecture, product owners, integration, operations, procurement, and Dweve product representatives. 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

A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions.

READ THIS FIRST • CANONICAL SPOTLIGHT

the current product set, named owned responsibilities

The current published page set is easier to select when each product is reduced to one accountable responsibility.

Fabric owns the workspace; Loom owns cognition; Nexus owns organised agent work; Spindle owns governed knowledge processing and lineage; Aura owns governed software-development work; Mesh owns distributed capacity and federated workload coordination; Charter owns the portable organisational record; Core owns the binary AI framework and runtime operations; Kera owns typed graph programming, planning and compilation.

A product may call another product or an open foundation. That dependency does not transfer ownership of the caller's work state. Loom reasoning inside a Nexus task remains cognition inside an organised task; Spindle material retrieved into Fabric remains governed knowledge inside a workspace; Core execution lowered by Kera remains runtime computation beneath the programme plan.

Every tier is customer-operated with no outbound dependency. Controlled includes verified escrow and review without modification rights. Autonomous adds full operational source and internal modification during the term. Independent adds development source, technology transfer, and the contracted path to perpetual self-maintenance.

AREA	CURRENT GUIDE RECORD	VERIFY / DECIDE
Fabric / Loom / Nexus	Workspace / cognition / organised agent work	User, reasoning and work-state boundaries
Spindle	Knowledge processing and lineage	Sources, atoms, quality, versions
Aura	Governed development work	Goal, plan, tools, gates, CI, record
Mesh	Distributed estate and workload	Locality, privacy, faults, receipts
Charter	Owned organisational record	Parties, domains, workflows, events
Core / Kera	Runtime framework / language and compiler	Operations, targets, graph plan, lowering

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

GUIDE-SPECIFIC PROOF

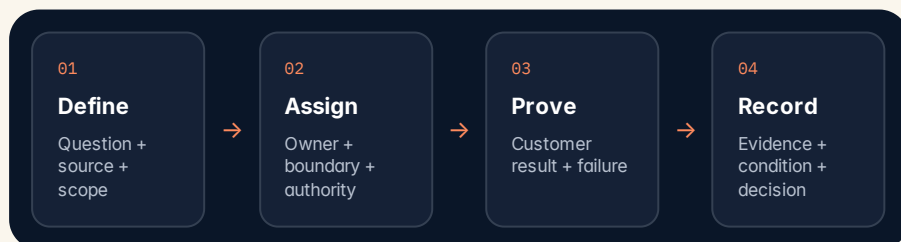
Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss.

01

PRODUCT PORTFOLIO AND RESPONSIBILITY BOUNDARIES

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

How to read the portfolio

responsibilities, layers, surfaces, and adoption paths; record any canonical count conflict instead of hiding it.

This chapter owns a specific part of the reader's decision: responsibilities, layers, surfaces, and adoption paths; record any canonical count conflict instead of hiding it.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "How to read the portfolio", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Responsibilities

Define the exact proposition and current authoritative source.

LENS 2

Layers

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

LENS 3

Surfaces

Specify the evidence and how an independent reviewer checks it.

LENS 4

And adoption paths

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 01 • APPLY

Turn “How to read the portfolio” 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, responsibilities, layers, surfaces, and adoption paths; record any canonical count conflict instead of hiding it., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 how to read the portfolio; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 02 • EXPLAIN

Fabric: the workspace

what users do there and what remains delegated below it.

This chapter owns a specific part of the reader's decision: what users do there and what remains delegated below it.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Fabric: the workspace", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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 users do there

Define the exact proposition and current authoritative source.

LENS 2

What remains delegated below it

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 responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 02 • APPLY

Turn “Fabric: the workspace” 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 users do there and what remains delegated below it., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 fabric: the workspace; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 03 • EXPLAIN

Loom: cognition

expert catalogue, perception, memory, search, reasoning, solver, and expression responsibilities.

This chapter owns a specific part of the reader's decision: expert catalogue, perception, memory, search, reasoning, solver, and expression responsibilities.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Loom: cognition", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Expert catalogue

Define the exact proposition and current authoritative source.

LENS 2

Perception

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

LENS 3

Memory

Specify the evidence and how an independent reviewer checks it.

LENS 4

Search

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 03 • APPLY

Turn “Loom: cognition” 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, expert catalogue, perception, memory, search, reasoning, solver, and expression responsibilities., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 loom: cognition; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 04 • EXPLAIN

Nexus: organised agent work

teams, roles, workflows, approvals, recovery, and why this is not the model layer.

This chapter owns a specific part of the reader's decision: teams, roles, workflows, approvals, recovery, and why this is not the model layer.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Nexus: organised agent 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Teams

Define the exact proposition and current authoritative source.

LENS 2

Roles

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

LENS 3

Workflows

Specify the evidence and how an independent reviewer checks it.

LENS 4

Approvals

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 04 • APPLY

Turn “Nexus: organised agent 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, teams, roles, workflows, approvals, recovery, and why this is not the model layer., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 nexus: organised agent 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 05 • EXPLAIN

Spindle: knowledge processing and lineage

knowledge intake, atomisation, quality, state, and historical record.

This chapter owns a specific part of the reader's decision: knowledge intake, atomisation, quality, state, and historical record.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Spindle: knowledge processing and lineage", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Knowledge intake

Define the exact proposition and current authoritative source.

LENS 2

Atomisation

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

LENS 3

Quality

Specify the evidence and how an independent reviewer checks it.

LENS 4

State

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 05 • APPLY

Turn “Spindle: knowledge processing and lineage” 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, knowledge intake, atomisation, quality, state, and historical record., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 spindle: knowledge processing and lineage; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

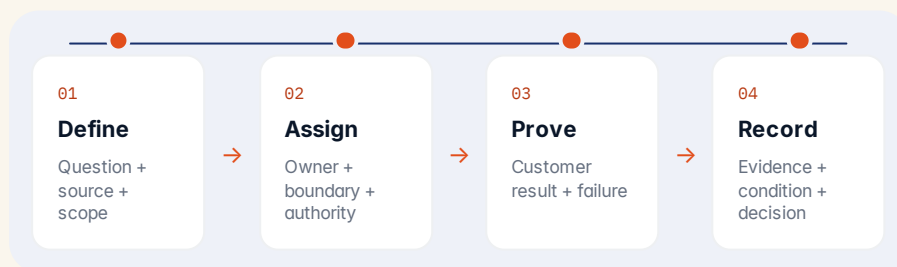
Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

02

PRODUCT PORTFOLIO AND RESPONSIBILITY BOUNDARIES

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

Aura: governed development

goals, plans, tools, models, permissions, quality loops, and engineering records.

This chapter owns a specific part of the reader's decision: goals, plans, tools, models, permissions, quality loops, and engineering records.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Aura: governed development", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Goals

Define the exact proposition and current authoritative source.

LENS 2

Plans

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

LENS 3

Tools

Specify the evidence and how an independent reviewer checks it.

LENS 4

Models

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 06 • APPLY

Turn “Aura: governed development” 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, goals, plans, tools, models, permissions, quality loops, and engineering 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 source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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

Aura ships as a single static binary with no runtime dependencies. Enterprise support, custom development, and training available from Dweve.

DECISION RECORD

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 07 • EXPLAIN

Mesh: distributed capacity and learning

local boundaries, proposals, receipts, faults, settlement, and what is never shared.

This chapter owns a specific part of the reader's decision: local boundaries, proposals, receipts, faults, settlement, and what is never shared.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Mesh: distributed capacity and learning", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Local boundaries

Define the exact proposition and current authoritative source.

LENS 2

Proposals

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

LENS 3

Receipts

Specify the evidence and how an independent reviewer checks it.

LENS 4

Faults

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 07 • APPLY

Turn “Mesh: distributed capacity and learning” 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, local boundaries, proposals, receipts, faults, settlement, and what is never shared., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 mesh: distributed capacity and learning; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 08 • EXPLAIN

Charter: the owned company record

parties, sources, domains, workflows, events, packs, and portability.

This chapter owns a specific part of the reader's decision: parties, sources, domains, workflows, events, packs, and portability.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Charter: the owned company 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Parties

Define the exact proposition and current authoritative source.

LENS 2

Sources

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

LENS 3

Domains

Specify the evidence and how an independent reviewer checks it.

LENS 4

Workflows

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 08 • APPLY

Turn “Charter: the owned company 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, parties, sources, domains, workflows, events, packs, and portability., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 charter: the owned company 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 09 • EXPLAIN

Core: binary AI framework

operations, formats, backends, runtime contracts, and deterministic computation.

This chapter owns a specific part of the reader's decision: operations, formats, backends, runtime contracts, and deterministic computation.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Core: binary AI framework", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Operations

Define the exact proposition and current authoritative source.

LENS 2

Formats

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

LENS 3

Backends

Specify the evidence and how an independent reviewer checks it.

LENS 4

Runtime contracts

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 09 • APPLY

Turn “Core: binary AI framework” 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, operations, formats, backends, runtime contracts, and deterministic computation., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 core: binary ai framework; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 10 • EXPLAIN

Kera: language and compiler

types, effects, graph programmes, planning, lowering, emission, targets, and its relationship to Core.

This chapter owns a specific part of the reader's decision: types, effects, graph programmes, planning, lowering, emission, targets, and its relationship to Core.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Kera: language and compiler", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Types

Define the exact proposition and current authoritative source.

LENS 2

Effects

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

LENS 3

Graph programmes

Specify the evidence and how an independent reviewer checks it.

LENS 4

Planning

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 10 • APPLY

Turn “Kera: language and compiler” 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, types, effects, graph programmes, planning, lowering, emission, targets, and its relationship to Core., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 kera: language and compiler; not every question needs every row, but no row may be implied without evidence.

DECISION RECORD

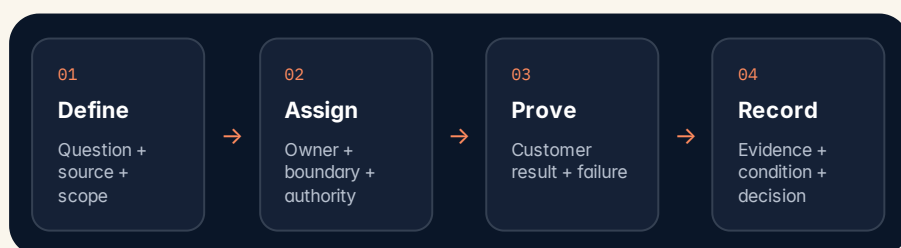
Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

03

PRODUCT PORTFOLIO AND RESPONSIBILITY BOUNDARIES

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

Common combinations

personal workspace, team knowledge, governed agent workflow, autonomous engineering, federated estate, owned company platform, and embedded compute.

This chapter owns a specific part of the reader's decision: personal workspace, team knowledge, governed agent workflow, autonomous engineering, federated estate, owned company platform, and embedded compute.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Common combinations", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Personal workspace

Define the exact proposition and current authoritative source.

LENS 2

Team knowledge

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

LENS 3

Governed agent workflow

Specify the evidence and how an independent reviewer checks it.

LENS 4

Autonomous engineering

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 11 • APPLY

Turn “Common combinations” 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, personal workspace, team knowledge, governed agent workflow, autonomous engineering, federated estate, owned company platform, and embedded compute., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 common combinations; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 12 • EXPLAIN

Selection worksheet

map users, workflow, knowledge, actions, policy, evidence, scale, deployment, and modification rights to the smallest coherent set.

This chapter owns a specific part of the reader's decision: map users, workflow, knowledge, actions, policy, evidence, scale, deployment, and modification rights to the smallest coherent set.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Selection worksheet", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

Map users

Define the exact proposition and current authoritative source.

LENS 2

Workflow

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

LENS 3

Knowledge

Specify the evidence and how an independent reviewer checks it.

LENS 4

Actions

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

OWNED ARTEFACT

This chapter contributes to a responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 12 • APPLY

Turn “Selection worksheet” 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, map users, workflow, knowledge, actions, policy, evidence, scale, deployment, and modification rights to the smallest coherent set., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 selection worksheet; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

CHAPTER 13 • EXPLAIN

Availability and commercial boundary

state only what current pricing and product pages support; move disputed availability into the conflict register.

This chapter owns a specific part of the reader's decision: state only what current pricing and product pages support; move disputed availability into the conflict register.

In this guide, the portfolio is best understood as a set of accountable product jobs: workspace, cognition, organised agent work, governed knowledge, engineering, distributed capacity, organisational record, binary compute, and language/compiler planning. Products can compose without becoming one shared blob. Applied to "Availability and commercial boundary", 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: A product owns a responsibility and its stable state. A surface explains how it is reached, a foundation supplies a bounded mechanism, and a deployment posture explains who operates it. Availability and rights remain product-page, pricing-page and contract questions. 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

State only what current pricing

Define the exact proposition and current authoritative source.

LENS 2

Product pages support

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

LENS 3

Move disputed availability into the conflict register

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 responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.

CHAPTER 13 • APPLY

Turn “Availability and commercial boundary” 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, state only what current pricing and product pages support; move disputed availability into the conflict register., 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: a solution owner maps one governed knowledge-and-action workflow to Fabric, Loom, Nexus and Spindle, then tests whether lower Core/Kera or open-foundation responsibilities are required directly or remain implementation dependencies. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss. 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 availability and commercial boundary; 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 select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.

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, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	A Accountability 5, 7-16, 18-27, 29-34	A Adoption 7-8, 28
A Appeal and challenge 3, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27-28, 30, 32, 34	A Architecture 4, 17	A Assurance 4, 7, 9, 11, 13, 15-16, 18, 20, 22, 24, 26-27, 29, 31, 33
A Aura 5, 18-19	A Authority 4, 6, 8, 10, 12, 14, 16-17, 19, 21, 23, 25, 27-28, 30, 32, 34	A Availability 4-5, 7-16, 18-27, 29-34
B Boundaries 3, 5-34	C Charter 5, 22-23	C Compliance 4
C Configuration 4, 7, 9, 11-13, 15, 18, 20, 22-24, 26, 29, 31, 33-34	C Continuity 28	C Contracts 4, 7, 9, 11, 13, 15-16, 18, 20, 22, 24, 26-27, 29, 31, 33
D Decision records 4, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	D Dependencies 3, 5, 7-16, 18-27, 29-34	D Deployment 4, 7-16, 18-27, 29-34
D Determinism 24-25	D Due diligence 4, 28	E Evidence 3-34
E Exit 8, 10, 12, 14, 16, 19, 21, 23, 25, 27-28, 30, 32, 34	F Fabric 3, 5, 8-10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	F Failure 3, 5-34
H Hardware 4	H Human authority 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	I Inputs 7, 9, 11, 13, 15, 18, 20, 22, 24, 26, 29, 31, 33
I Integration 4	K Kera 3, 5, 8, 10, 12, 14, 16, 19, 21, 23, 25-27, 30, 32, 34	L Licensing 4
L Limits and exclusions 8-10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32-34	L Lineage 5, 15-16	L Loom 3, 5, 8, 10-12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34
M Manifests 4, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	M Mesh 5, 20-21	M Models and solvers 8, 10-14, 16, 19, 21, 23, 25, 27, 30, 32, 34

SUBJECT INDEX CONTINUED

Subject index · N–W

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

N Nexus 3, 5, 8, 10, 12–14, 16, 19, 21, 23, 25, 27, 30, 32, 34	O Operations 8, 10, 12, 14, 16, 19, 21, 23, 25, 27–28, 30, 32, 34	O Outputs 3–4
O Ownership 3–34	P Performance 4	P Policies 3–4, 17, 31–32
P Portability 22–23	P Pricing 4, 7, 9, 11, 13, 15, 18, 20, 22, 24, 26, 29, 31, 33–34	P Privacy 5
P Procurement 4, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	P Proof 3–5, 7–16, 18–27, 29–34	R Recovery 13–14, 28
R Responsibility 3–34	R Rights 4–5, 7, 9, 11, 13, 15, 18, 20, 22, 24, 26, 29, 31–33	R Risk 3–4, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34
R Rules 4, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	S Security 4	S Sources 3–34
S Sovereignty 4–5, 7–16, 18–27, 29–34	S Spindle 3, 5, 8, 10, 12, 14–16, 19, 21, 23, 25, 27, 30, 32, 34	S State 3–16, 18–27, 29–34
T Testing 3–5, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	V Verification 3, 5, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 30, 32, 34	V Versioning 4, 7–9, 11, 13, 15, 18–20, 22, 24, 26, 29–31, 33
W Workflows 3–5, 8, 10, 12, 14, 16, 19, 21, 23, 25, 27, 29–32, 34		

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 responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof. 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: Write the workflow responsibilities, remove every product without a named job, use each selected product handbook to test contracts/state/failure/evidence, verify current availability and commercial inclusion, and demonstrate that one product can be replaced or omitted without hidden state loss.

Have business and workflow owner, enterprise and solution architecture, product owners, integration, operations, procurement, and Dweve product representatives 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 product-portfolio-guide-decision		
QUESTION	select the minimum product set for a workload and understand the boundaries that prevent overlap, double buying, or architectural confusion.	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 responsibility map and minimum product selection for one workload, including callers, outputs, dependencies, non-responsibilities, interfaces, posture and proof.