

Data Product Operating Framework

One universal operating model for directing, defining, operating and assuring data products, with enterprise and AI-agent-first implementation profiles.



VERSION

0.2.0 · Draft

PUBLISHED

2026-10-06

LICENSE

CC BY 4.0

Draft publication. Version 0.2.0 is available for review and contribution; it is not an adopted framework release.

FOUNDATION

Framework Core

1. Purpose

The Data Product Operating Framework Core is a universal, vendor-neutral and technology-neutral operating model for managing data products from business demand to measurable value.

It provides a common operating structure connecting:

- business objectives
- use cases and demand
- investment decisions
- data products
- machine-readable product contracts
- semantics
- governance requirements
- delivery and consumption
- lifecycle management
- operational evidence
- business outcomes
- value

The Core works for organisations operating with:

- human users
- traditional applications
- analytics systems
- data platforms
- automation
- AI systems
- AI agents

The framework does not require organisations to operate AI agents. It provides one common operating model that supports organisations from traditional enterprise data product management through AI-agent-first operations.

The Core remains machine-readable and automation-friendly by design. The ODPS standards family provides the machine-readable standards foundation underneath the framework. The framework is an operating model, not another ODPS-family specification.

The Core defines required operating concerns rather than one mandatory organisational design. A small organisation may combine several accountabilities in one role; a large organisation may distribute them across product, domain, platform, governance and assurance teams. What matters is that the decision rights, practices, artifacts, evidence and outcomes remain explicit and traceable.

Examples and external references explain how a requirement can be implemented. They do not silently create additional Core requirements. Applicable law, regulation, contracts and organisational policy may impose stronger obligations in a particular environment; those obligations should be represented through the relevant profile, product context and control model.

2. The four questions

DIRECT

Why should we do this, and who decides?

DEFINE

What exactly are we committing to provide?

OPERATE

How do we make it available, use it and change it?

ASSURE

Does it work, comply and create value?

3. Core rule

Intent and directives flow toward execution.

Evidence and value flow toward decision-making.

Forward traceability:

Objective → Use Case → Product Decision → Data Product → Product Contract → Delivery → Consumption →
Evidence → Outcome → Value → Investment Decision

Backward traceability:

Investment Decision → Value → Outcome → Evidence → Consumption → Delivery → Product Contract → Data
Product → Product Decision → Use Case → Objective

The same identifiers and relationships should make this chain traversable in both directions.

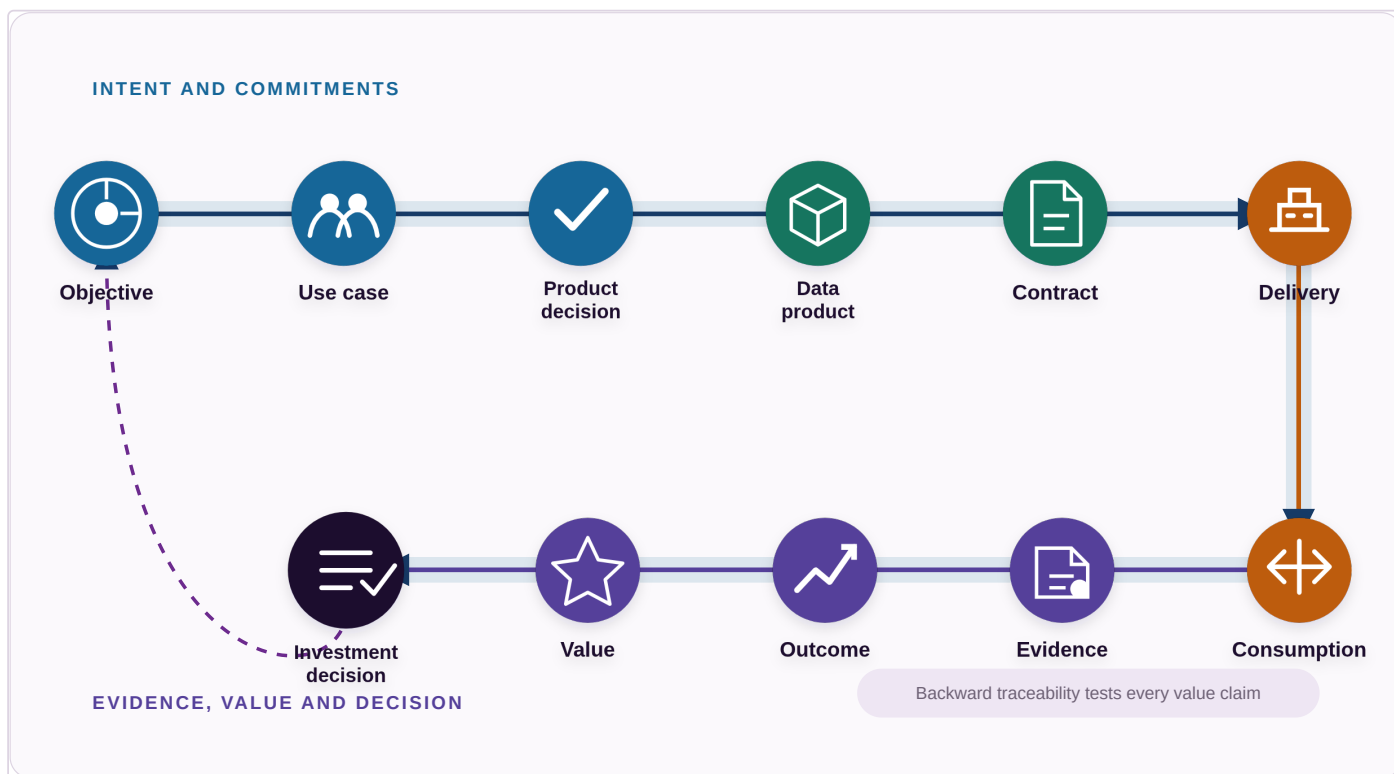


Figure 1. Intent and commitments flow toward delivery; evidence and value flow back toward accountable investment decisions.

4. Core foundation principles

F1. Start from demand

The existence of data does not justify creating a data product.

Investment starts from:

- an objective
- a problem
- a use case
- an obligation
- a consumer need

F2. Separate use cases from data products

A use case explains why data is needed.

A data product defines a reusable capability provided to consumers.

The relationship is many-to-many:

Objective → Use Case → Data Product

One use case can require many products, and one product can support many use cases.

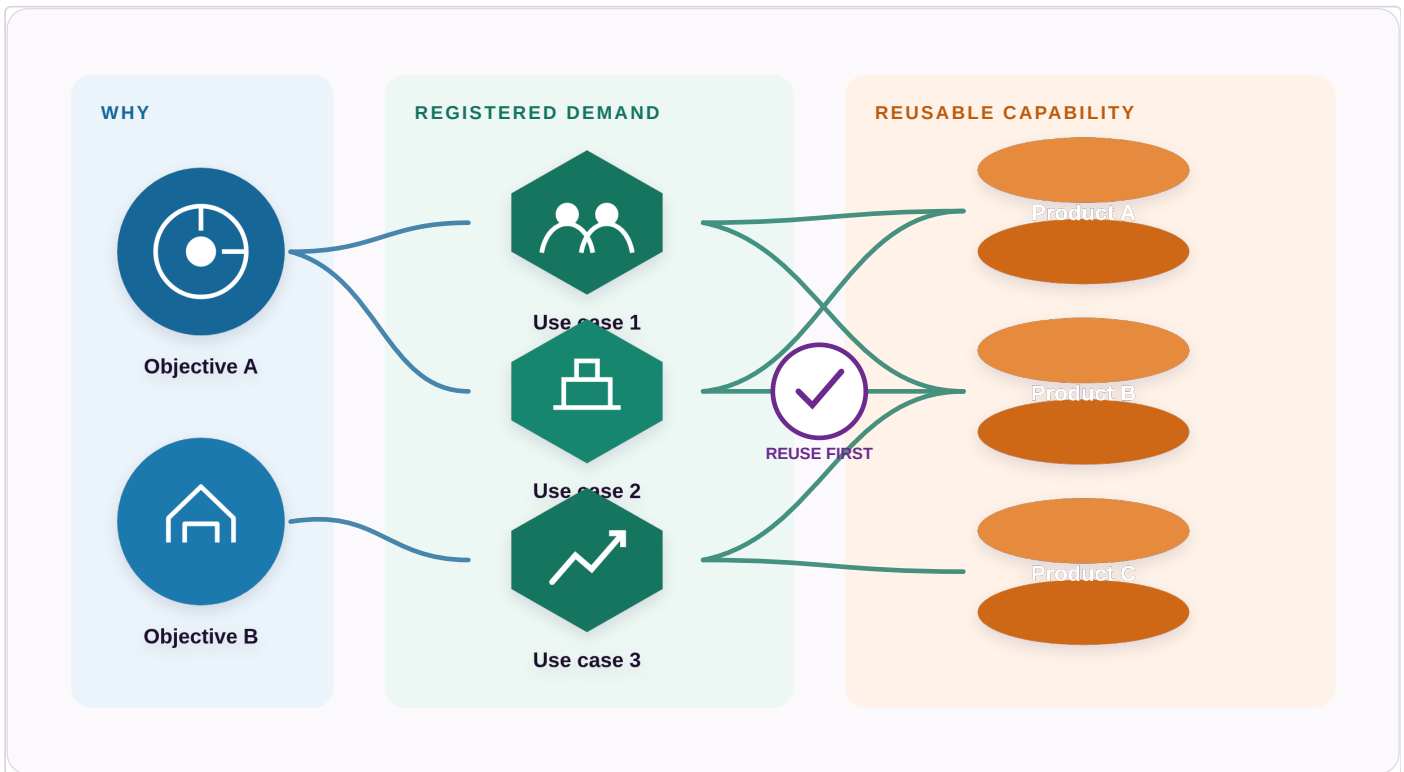


Figure 2. Use cases preserve the reason for demand while reusable data products may serve several needs across the portfolio.

F3. Data products are contracts

A managed data product must have an authoritative definition.

The contract describes:

- what the product is
- who provides it
- what it provides
- how it is accessed
- what it means
- what it promises
- what conditions apply
- how it changes

F4. Machine-readable by default, human-readable by presentation

The authoritative representation should favour machine-readable structures.

Human-readable views such as web pages, PDF documents, catalog pages, dashboards and reports should derive from the underlying structured artifacts where practical.

ODPS is machine-readable by design. Human-readable documentation and interfaces should increasingly be generated from the same underlying product artifacts used by software and AI agents.

F5. Preserve business context

Business context should survive the transition from strategy into technical implementation.

A product should remain connected to:

- objectives
- use cases
- consumers
- KPIs
- owners
- dependencies
- policies

F6. Separate declaration from execution

The ODPS standards family declares portable contracts and context.

Operational platforms execute work.

Runtime systems produce evidence.

DECLARE → EXECUTE → EVIDENCE

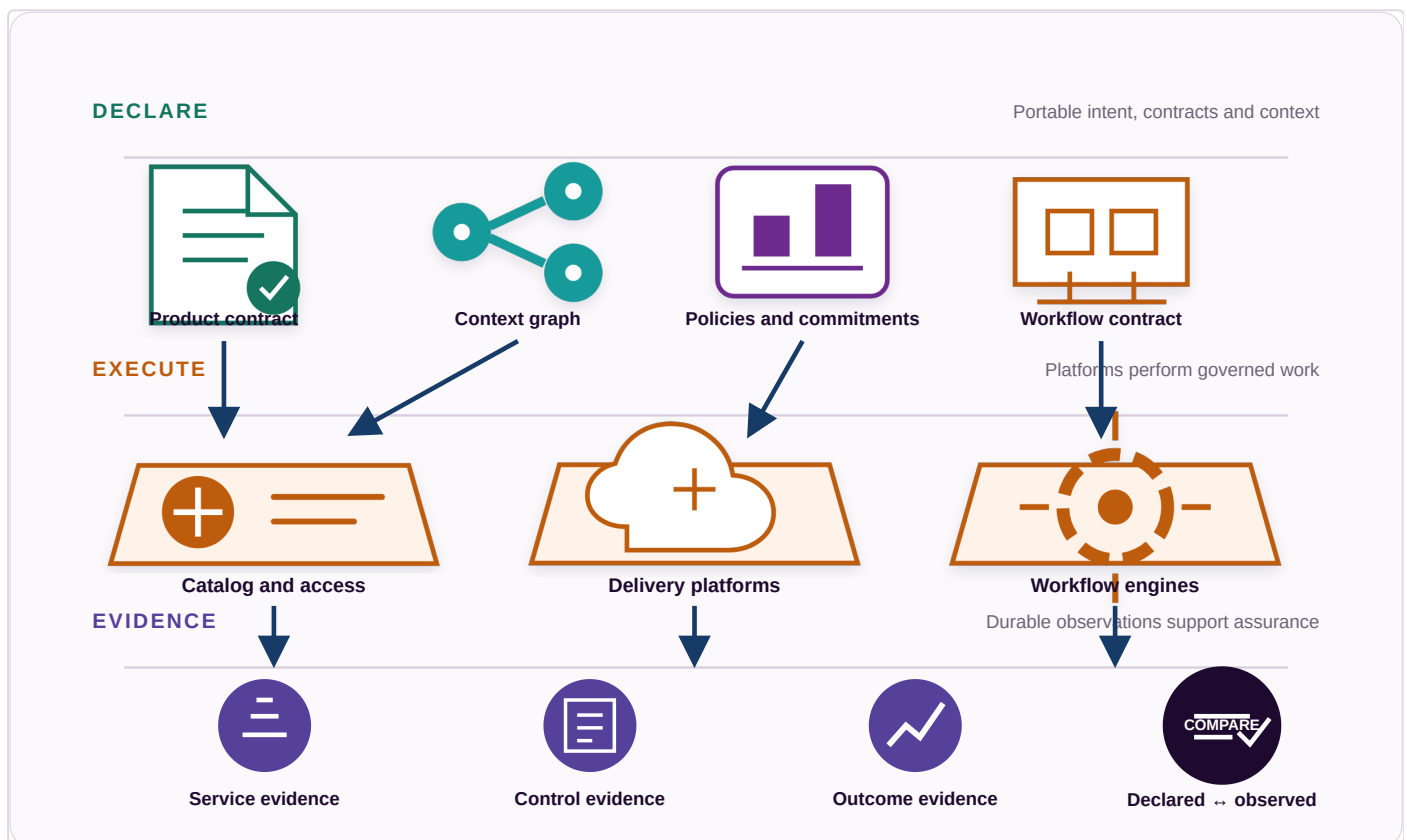


Figure 3. Portable declarations guide operational systems; runtime observations become evidence that can be compared with the declared state.

F7. Treat evidence as a first-class object

Claims must be backed by evidence.

- A quality target is not evidence of quality.
- An SLA is not evidence of service performance.
- A policy reference is not evidence that a control worked.
- A use case is not evidence of adoption.
- An objective is not evidence of value.

F8. Separate declared state from observed state

Declared state describes what should happen.

Observed state describes what runtime evidence shows happened.

The difference between the two is a governance signal.

F9. Measure value beyond usage

Usage is evidence of activity. It is not automatically evidence of value.

The framework preserves the chain:

Discovery → Access → Adoption → Consumption → Use-case outcome → Organisational value

F10. Govern the full lifecycle

Products should be deliberately:

- proposed
- approved
- defined
- developed
- tested
- released
- changed
- deprecated
- retired

F11. Design for interoperability

The framework must work across:

- catalogs
- data platforms

- cloud environments
- governance tools
- workflow engines
- AI systems
- organisational structures

No single implementation platform is required.

F12. Design for both people and machines

The same underlying product context should serve:

- human consumers
- applications
- developer tools
- governance systems
- AI agents

Different interfaces can present the information differently. The underlying meaning should remain consistent.

F13. Automation does not equal maturity

A poor process does not become mature because it is automated.

Maturity continues to be assessed through:

- Accountability
- Practice
- Evidence
- Outcome

F14. One traceability chain

The framework preserves end-to-end traceability:

Objective → Use Case → Product Decision → Data Product → Product Contract → Delivery → Consumption →
Evidence → Outcome → Value → Investment Decision

The chain works in both directions.

F15. One Core, multiple profiles

The Core defines universal capabilities.

Profiles strengthen or specialise those capabilities for specific operating environments. Profiles must not redefine the Core.

5. Framework structure

The universal Core contains:

- 4 functions
- 14 capabilities
- common principles
- common terminology
- a common traceability model
- a common evidence model
- a common assessment model
- a common conformance model
- practices within each capability
- artifacts produced by those practices
- evidence demonstrating execution and outcomes
- measures evaluating performance

Profiles adapt this shared foundation to specific operating environments. They are not separate frameworks.

6. Function 1: DIRECT

Purpose: connect organisational intent to explicit data product investment and accountability.

C1. Strategy and Objectives

Connect data product activity to explicit organisational objectives, mandates and measurable outcomes.

C2. Demand and Use Case Management

Capture, qualify and prioritise the problems, decisions and consumer needs creating demand for data.

C3. Portfolio, Investment and Accountability

Decide which products deserve investment, assign accountability, manage funding and govern portfolio lifecycle decisions.

DIRECT produces:

- objectives
- use cases
- priorities
- investment decisions
- accountability
- product candidates

- success measures

7. Function 2: DEFINE

Purpose: turn an approved product candidate into an explicit and governed data product definition.

C4. Product Definition and Contract

Establish the authoritative machine-readable definition of the data product.

C5. Semantics and Vocabulary

Establish shared meaning for important concepts and terminology, using machine-readable representations where practical.

C6. Product Commitments and Usage Conditions

Define measurable quality, service, access, legal, privacy, security and commercial conditions where applicable.

C7. Relationships, Dependencies and Context

Connect products to objectives, use cases, KPIs, policies, systems, products and other relevant context.

DEFINE produces:

- product contracts
- product identities
- semantics
- quality commitments
- service commitments
- access conditions
- governance context
- relationships
- dependencies

8. Function 3: OPERATE

Purpose: make defined products discoverable, accessible, consumable and manageable throughout their operational lifecycle.

C8. Catalog, Publication and Discovery

Publish and organise governed products and business context so people, applications and automated systems can find suitable products.

C9. Provisioning, Integration and Consumption

Manage the controlled path from product discovery to usable consumer access and ongoing consumption.

C10. Lifecycle, Version and Change Management

Control how products move through lifecycle states and how versions and changes affect consumers.

C11. Workflow, Automation and Agent Operations

Define repeatable and reviewable operating procedures for people, software systems, automation and AI agents where used.

OPERATE produces:

- catalog entries
- access
- subscriptions
- integrations
- versions
- changes
- releases
- workflow executions
- approvals
- operational records

9. Function 4: ASSURE

Purpose: determine whether products operate as promised, remain within required boundaries and produce sufficient value.

C12. Observability and Service Assurance

Compare actual product quality and service performance with declared commitments.

C13. Governance, Risk, Compliance and Control Assurance

Determine whether applicable policies and controls operate effectively and produce sufficient evidence.

C14. Adoption, Outcomes and Value Realisation

Determine whether intended consumers use products and whether that consumption contributes to expected outcomes and value.

ASSURE produces:

- quality evidence
- service evidence
- control evidence
- risk findings
- adoption evidence
- outcome evidence
- value evidence
- investment recommendations

10. Cross-cutting enablers

E1. Platform and Automation

Catalogs, data platforms, workflow systems, observability systems, policy engines, identity systems, developer platforms and AI infrastructure.

E2. Engineering and Interoperability

APIs, schemas, validation, data contracts, CI/CD, metadata exchange, MCP and integration patterns.

E3. Competency and Culture

Product management, domain expertise, data literacy, governance competence, engineering capability, leadership, ownership behaviour and consumer orientation.

11. Authority model

The Core uses the following separation of authority. These responsibilities are complementary and must not be collapsed into a new framework specification.

ODPS

Authoritative machine-readable representation of an individual data product contract.

It answers:

- What is this product?
- What does the provider offer?
- What commitments and conditions apply?

ODPC

Authoritative portable representation of portfolio and discovery objects.

It answers:

- What products, use cases, objectives, signals and related portfolio objects are being managed?

ODPV

Authoritative shared vocabulary and semantics for the standards family.

It answers:

- What do these concepts mean?

ODPG

Authoritative portable representation of relationships and context.

It answers:

- How are these objects connected?

ODPR

Authoritative portable representation of reusable workflow contracts.

It answers:

- How should this type of work happen?

Operational systems

Responsible for runtime execution and relevant runtime observations.

They answer:

- What happened?

Evidence layer

Provides durable assurance information that proves what happened.

It answers:

- What proves it?

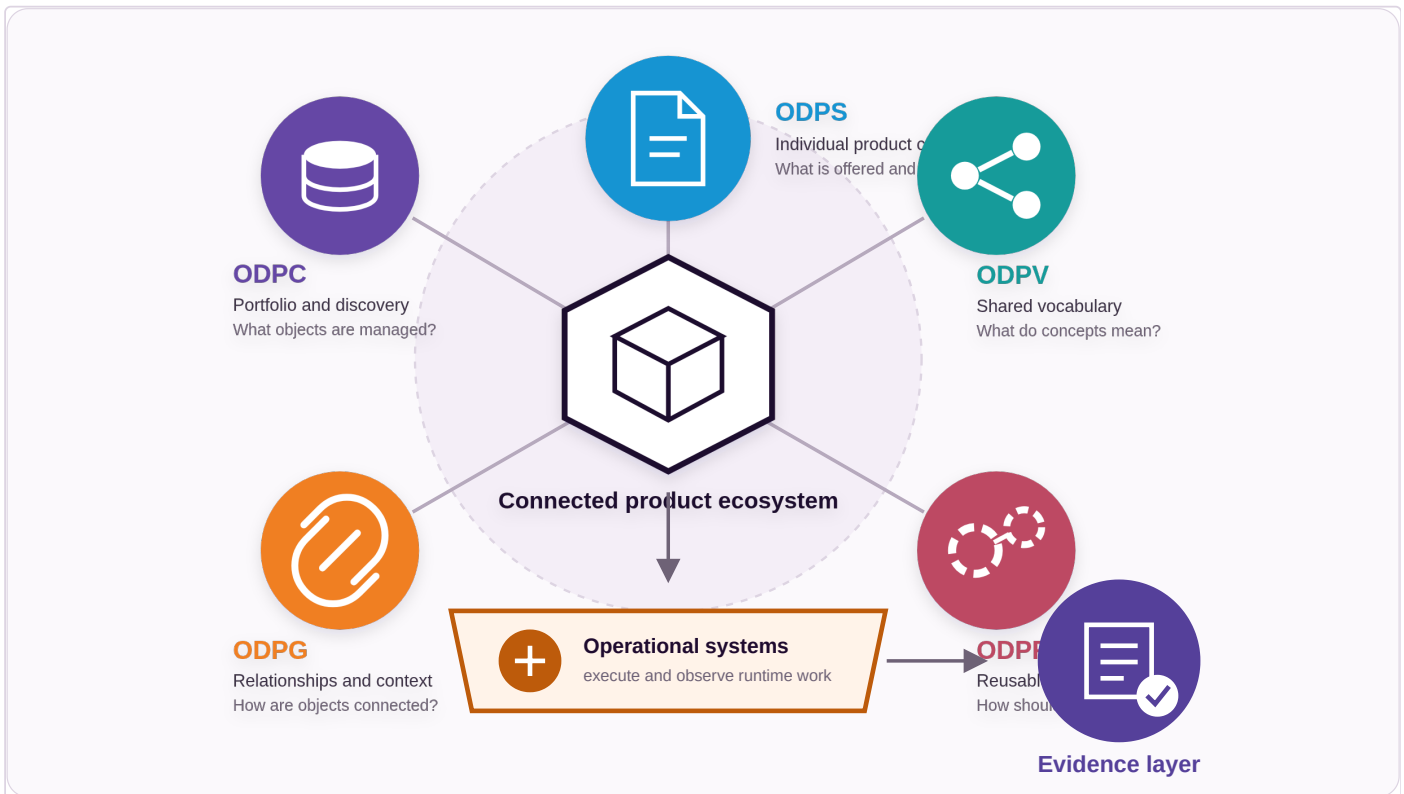


Figure 4. Each ODPS-family standard has a distinct authority; operational systems execute work and the evidence layer proves what occurred.

12. Local context and shared context

Relevant context can appear inside an individual product contract even when a canonical portfolio object exists elsewhere.

The distinction is:

- local context makes the product independently understandable
- shared context creates reusable organisational objects
- relationships preserve alignment between them

For example:

- ODPS can state which objective an individual product supports
- ODPC can represent that Business Objective as a reusable portfolio object
- ODPG connects the product and objective explicitly

This is intentional contextual redundancy, provided identifiers and relationships preserve consistency.

13. Declared state and observed state

Declared state

What the product contract, policy or workflow states should happen.

Observed state

What runtime evidence shows happened.

Example:

Declared quality:

Completeness ≥ 98%

Observed quality:

Completeness = 94.7%

Assurance result:

Commitment not met

The difference between declared and observed state is a first-class governance signal.

14. Evidence model

Evidence is information retained so that a material claim, execution or decision can be examined. It should identify what it concerns, where it came from, when it was produced, which version and scope applied, how it was generated and who or what is accountable for it. The required strength and retention period depend on the consequence of the decision being supported.

Evidence may be a structured event, measurement, validation report, signed approval, decision record, test result or other durable artifact. A screenshot or narrative may be sufficient for a low-risk review but weak for repeatable automated assurance. The framework therefore specifies evidence purpose and traceability while allowing implementation profiles and organisations to define proportionate formats and controls.

Directive evidence

Objectives, priorities, policies, funding decisions, portfolio decisions and approvals.

Definition evidence

Product contracts, vocabulary, commitments, relationships and control definitions.

Execution evidence

Workflow executions, validations, approvals, deployments, provisioning, versions and changes.

Operational evidence

Quality observations, service observations, consumption, incidents, control executions and exceptions.

Outcome evidence

Adoption, use-case performance, KPI changes, cost, benefit, risk reduction, public value and other demonstrated outcomes.

Evidence quality should be evaluated through at least:

- identity — the object, product, control, workflow or decision is unambiguous
- provenance — the producing person, system or activity is known
- time — production time and applicable evaluation period are known
- version — the relevant contract, policy, workflow and implementation versions are known
- method — the collection, calculation or decision method is inspectable

- integrity — material alteration can be detected or governed
- scope — inclusions, exclusions and applicability are stated
- retention — evidence remains available for the decisions and obligations it supports

More evidence is not automatically better evidence. Collection should be proportionate, privacy-aware and connected to a decision or assurance need.

15. Value model

The framework distinguishes six value classes:

1. Financial value
2. Operational value
3. Decision value
4. Risk value
5. Compliance value
6. Public value

The framework distinguishes four attribution levels:

1. Direct attribution
2. Contribution
3. Enablement
4. Mandatory value

Value claims should state the baseline or counterfactual, measurement period, cost boundary, assumptions and attribution method. Financial conversion is useful only when credible; operational, decision, risk, compliance and public value may require different units and decision criteria.

The value model is designed for investment decisions rather than promotional reporting. Uncertainty, negative outcomes and costs should remain visible. A product may be necessary despite weak direct financial attribution, or widely used while contributing little to the outcome that justified it.

16. Four assurance views

Service Health

Does the product perform as promised?

Control Health

Does it operate within required governance boundaries?

Adoption Health

Are intended consumers using it?

Value Health

Are intended outcomes being achieved?

These views should not be collapsed into one opaque score.

17. Capability assessment

Each capability is evaluated independently.

- Level 0: Absent
- Level 1: Emerging
- Level 2: Defined
- Level 3: Operational
- Level 4: Measured
- Level 5: Adaptive

Assessment considers four dimensions:

- Accountability
- Practice
- Evidence
- Outcome

18. Framework profiles

The Core remains context-neutral.

The profile architecture is:

Data Product Operating Framework Core → Enterprise Data Product Profile → AI-Agent-First Data Product Profile

The **Enterprise Data Product Profile** provides the baseline implementation profile for organisations managing data products through conventional enterprise systems and human-led operating models.

The **AI-Agent-First Data Product Profile** extends the Core with stronger requirements for environments where AI agents directly discover, interpret, access, combine or act on data products.

Profiles may:

- extend Core requirements
- identify stronger practices
- identify required evidence
- identify mandatory machine-readable artifacts
- identify stronger control requirements
- define context-specific assessment expectations

Profiles must not:

- duplicate the entire framework
- rename Core capabilities

- create separate maturity models
- fork Core terminology
- redefine ODPS-family standards

Each profile uses the common C1-C14 capability model and the common assessment dimensions of Accountability, Practice, Evidence and Outcome.

19. Framework conformance

The framework distinguishes three forms of conformance:

Artifact conformance

A machine-readable artifact complies with the applicable specification.

Practice conformance

An organisation follows the required operating practices.

Capability conformance

Evidence demonstrates that the capability operates and achieves the required outcomes.

Having a valid ODPS file does not mean the organisation has mature data product management.

It means one important artifact conforms to the standard.

20. Core traceability test

For any operational data product, the organisation should be able to answer:

- What is it?
- Who is accountable for it?
- Why does it exist?
- Which use cases depend on it?
- Which objectives do those use cases support?
- Who approved its investment?
- Which version is running?
- What does it promise?
- How is it accessed?
- Who consumes it?
- What depends on it?
- Which policies apply?
- Which controls protect it?
- Which workflows operate it?
- Did those controls execute?
- Is it meeting its quality commitments?
- Is it meeting its service commitments?

- What changed recently?
- What incidents affected it?
- Is adoption increasing or declining?
- What outcome was expected?
- What outcome occurred?
- What did the product cost?
- What value did it contribute?
- When was continued investment last reviewed?

21. Machine-readable framework objective

Machine-readable by default, human-readable by presentation is a major Core principle.

The objective is that a significant portion of the Core traceability test can be answered from machine-readable artifacts and evidence.

The framework therefore aims toward:

- machine-readable objectives and use cases
- machine-readable data product contracts
- machine-readable semantics
- machine-readable relationships
- machine-readable policies where suitable
- machine-readable workflows
- machine-readable runtime evidence
- machine-readable assurance results
- traceable value

Human-readable web pages, PDF documents, catalog pages, dashboards and reports should derive from the same underlying structured artifacts where practical. Generated views support communication; they do not replace the authoritative artifacts or the normative Markdown sources of this framework repository.

22. Evidence-informed design and external references

The Core is an Open Data Product Initiative operating-model design. External sources inform its reasoning and provide implementation patterns, but they do not become framework requirements merely because they are cited. The governed [Framework Evidence Library](#) records authority, applicability, rights and intended use for each source.

Important supporting sources include:

- [w3c-dcat-3](#) — W3C Data Catalog Vocabulary 3 for interoperable catalog, dataset, service, distribution and relationship patterns
- [w3c-skos](#) — W3C SKOS for concept identity, labels, semantic relationships and vocabulary mappings
- [w3c-prov-o](#) — W3C PROV-O for provenance across entities, activities and agents
- [w3c-dqv](#) — W3C Data Quality Vocabulary for quality dimensions, metrics, measurements and annotations
- [w3c-odrl-2.2](#) — W3C ODRL Information Model 2.2 for expressing permissions, prohibitions, duties and constraints

- **w3c-shacl** — W3C SHACL for machine-readable constraints and validation results
- **wilkinson-fair-principles-2016** — FAIR Guiding Principles for persistent identity, rich metadata, provenance and machine actionability
- **nist-csf-2.0**, **nist-privacy-framework-1.0** and **nist-sp-800-53r5** — NIST Cybersecurity Framework 2.0, NIST Privacy Framework 1.0 and NIST SP 800-53 Rev. 5 for risk, control, lifecycle and assurance patterns
- **nist-ai-rmf-1.0**, **nist-ai-600-1-genai-profile** and **oecd-ai-principles-2024** — NIST AI Risk Management Framework 1.0, its Generative AI Profile and the OECD AI Principles for AI-specific governance where AI systems or agents are in scope
- **eu-ai-act-2024-1689** — Regulation (EU) 2024/1689 as a jurisdiction-specific legal source for applicable AI obligations, not as a universal Core requirement
- **iso-iec-42001-2023** and **edm-council-dcam** — ISO/IEC 42001:2023 and EDM Council DCAM as licensed comparison points; their protected content is not reproduced by this framework

Sources may disagree, address different objects or apply only in particular jurisdictions. Maintainers should cite the primary source, disclose applicability and preserve the framework's authority boundary when proposing changes.

23. Positioning statement

The Data Product Operating Framework is an open operating model for managing data products from business demand to measurable value.

It connects strategy, use cases, investment, product contracts, semantics, governance, delivery, lifecycle management and assurance through shared machine-readable context and evidence.

The framework does not require organisations to operate AI agents. It provides one common operating model that supports organisations from traditional enterprise data product management through AI-agent-first operations.

The ODPS standards family provides the interoperable, machine-readable standards foundation underneath the framework.

The framework explains how the parts work together. It is not itself another ODPS-family specification.

FUNCTION 1

DIRECT

Purpose

DIRECT ensures that data products exist for an explicit reason, serve identifiable demand, have accountable ownership, and receive investment based on expected value.

It creates the governance chain between organisational intent and individual data products.

C1. Strategy and Objectives

Purpose

Ensure data product activity is anchored in explicit organisational objectives and measurable outcomes.

Operating intent

Strategy becomes operational only when it changes a decision. An objective should therefore be specific enough to influence which demand is accepted, which product is funded, what outcome is expected and when investment should be reconsidered. Broad statements such as “become data driven” may express direction, but they do not provide a sufficient basis for prioritisation or assurance.

The objective record should preserve both accountability and evaluation context: the owner, intended beneficiaries, baseline, target, time horizon, constraints and assumptions. Measures may change as the organisation learns, but the reason for the change and the approving authority should remain traceable. This prevents later outcome reporting from being detached from the decision that originally justified the work.

Expected outcomes

- Data product investments connect to recognised organisational objectives.
- Expected outcomes are measurable.
- Evidence from data products feeds back into strategic decision-making.
- Objectives have accountable owners.
- Strategic priorities influence portfolio decisions.

Practices

- 1 Establish data product direction.
- 2 Register relevant objectives.
- 3 Assign objective ownership.
- 4 Define outcome measures.
- 5 Establish strategic guardrails.
- 6 Review strategic alignment.

Evidence

- approved objective
- defined KPI or outcome measure
- relevant policy or mandate
- named accountable owner
- strategic priority decision
- traceable downstream relationships

Example measures

- percentage of active products linked to at least one approved use case
- percentage of funded initiatives traceable to an organisational objective
- percentage of objectives with defined outcome measures
- number of products without identifiable strategic or demand justification

Assurance questions

- ? Can the objective owner explain which product investments the objective has changed?
- ? Is the expected outcome distinguishable from an activity, deliverable or technology deployment?
- ? Is there a baseline or an explicit explanation of why one cannot yet be established?
- ? Can decision-makers see when evidence contradicts the original strategic assumption?

C2. Demand and Use Case Management

Purpose

Translate organisational objectives and real consumer needs into explicit demand for data.

Operating intent

Demand management is the point where an organisational ambition becomes a testable statement of need. A use case should identify the consumer, the decision or process to be improved, the present limitation, the expected change and the evidence that would indicate success. It should not prescribe a new product before existing products and non-product alternatives have been evaluated.

Separating use cases from products makes reuse visible. If several use cases require the same stable capability, that is evidence for a reusable product. If one proposed product bundles unrelated demand, the portfolio may need to split it. The use-case record should remain valid even if the eventual product, interface or implementation platform changes.

Core rule

Objective ≠ Use Case ≠ Data Product

One objective can generate many use cases.

One use case can require many data products.

One data product can support many use cases.

Practices

1 Capture demand.

2 Identify the consumer.

3 Define the problem.

4 Define the expected outcome.

5 Define evidence needs.

6 Search existing products.

7 Identify capability gaps.

8 Qualify and prioritise demand.

9 Connect demand to products.

Evidence

- registered use case
- defined consumer
- expected outcome
- evaluation of existing products
- traceable objective relationship
- named use-case owner
- problem statement
- success measure
- prioritisation decision
- traceable product relationship

Example measures

— percentage of product initiatives originating from registered demand

— percentage of use cases linked to measurable outcomes

— percentage of new demand evaluated against existing products first

— product reuse across multiple use cases

— duplicate product requests avoided

— time from accepted demand to product decision

Assurance questions

-  Is the consumer named or represented by an accountable role rather than an abstract audience?
 -  Does the use case describe a decision, process or obligation that can change?
 -  Were existing products evaluated before a new product candidate was approved?
 -  Are rejected, deferred and consolidated requests retained with their decision rationale?
-

C3. Portfolio, Investment and Accountability

Purpose

Convert qualified demand into explicit decisions about data product investment, ownership and lifecycle.

Operating intent

The portfolio is a decision system, not merely an inventory. It should expose competing demands, product overlap, dependency concentration, total operating cost and the evidence used to continue or stop investment. A catalog can list products, but only accountable portfolio governance can decide whether they should exist.

Accountability should distinguish the person who owns the product outcome from people who provide engineering, governance or platform services. Funding should include the continuing cost of operating, assuring and eventually retiring the product—not only initial delivery. Portfolio review should occur on a defined cadence and when material triggers arise, such as loss of active demand, repeated commitment failures, major dependency changes or a superior reusable product becoming available.

Decision records should state the options considered, evidence available, assumptions made and authority used. This allows later reviewers to distinguish a poor decision from a reasonable decision made with incomplete information.

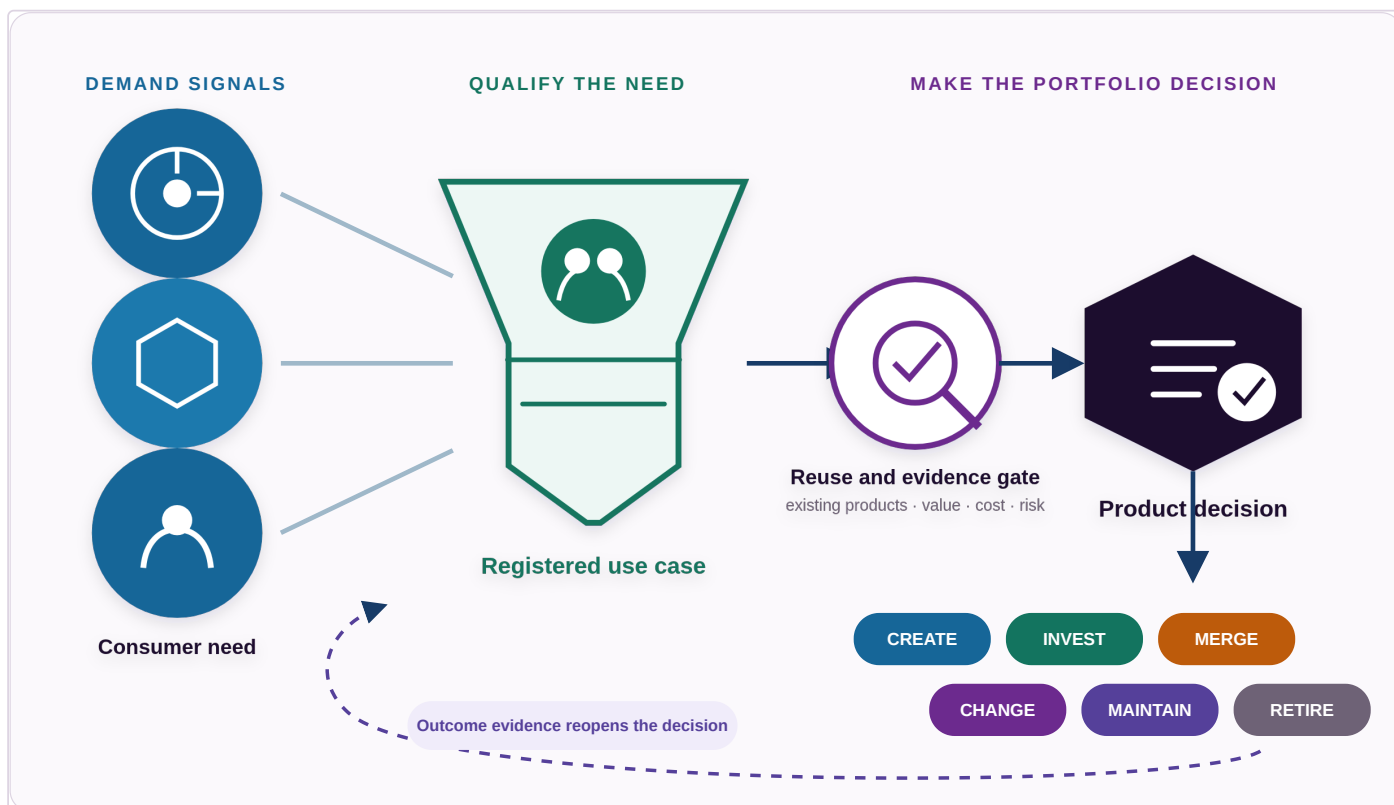


Figure 5. Portfolio governance converts qualified demand into an explicit lifecycle decision and uses later evidence to revisit that decision.

Practices

- 1 Maintain the product portfolio.
- 2 Evaluate product candidates.
- 3 Assign accountable ownership.
- 4 Make investment decisions.
- 5 Establish funding.
- 6 Evaluate reuse and portfolio overlap.
- 7 Manage dependencies.
- 8 Review portfolio performance.
- 9 Make lifecycle decisions.

Lifecycle decisions

create | invest | expand | maintain | change | merge | reduce | retire

Evidence

- approved product decision
- investment rationale
- portfolio priority
- dependency record
- accountable owner
- funding source
- lifecycle state
- review history

- usage evidence
- retirement or continuation decision
- outcome evidence

Example measures

- percentage of active products with accountable owners
- percentage with identified operating funding
- percentage linked to active demand
- portfolio spend by strategic objective
- reuse rate
- overlapping products
- products with no measurable consumption
- products with no active use cases
- products reviewed within policy period

Assurance questions

- ? Can the accountable owner make or escalate lifecycle and commitment decisions?
- ? Does investment cover continuing operation, assurance, change and retirement?
- ? Can the portfolio identify products with overlapping propositions or concentrated dependencies?
- ? Are continuation and retirement decisions supported by outcome evidence rather than usage alone?

DIRECT control loop

Strategy and Objectives → Demand and Use Cases → Portfolio and Investment → DEFINE → OPERATE → ASSURE → Portfolio Review → Strategy Review

DIRECT remains active after approval. Evidence from ASSURE may invalidate an objective assumption, reveal a different consumer need or show that continued investment is no longer justified. The control loop therefore needs explicit review dates, trigger events and decision owners rather than relying on informal escalation.

Evidence basis and external references

The framework requirements above are Open Data Product Initiative operating-model decisions. The following external sources provide supporting context rather than additional conformance requirements:

- **nist-csf-2.0** — NIST Cybersecurity Framework 2.0 uses outcome-oriented guidance and explicitly allows implementation to vary by organisation and use case. That supports separating desired outcomes from prescribed implementation activity.
- **nist-privacy-framework-1.0** — NIST Privacy Framework 1.0 connects enterprise risk management with cross-functional accountability and decision-making. It informs the treatment of ownership, context and risk without making the privacy framework mandatory for all data products.

- **nist-ai-rmf-1.0** — NIST AI Risk Management Framework 1.0 connects governance activity to organisational priorities and risk tolerance. It is relevant when AI is in scope but does not make AI-specific governance a Core dependency.
- **edm-council-dcam** — EDM Council DCAM is retained as a professional-framework comparison point for enterprise capability and assessment coverage. Its licensed content is not reproduced and it does not define framework conformance.

FUNCTION 2

DEFINE

Purpose

DEFINE turns an approved data product candidate into an explicit, governed and machine-readable product definition.

Separation of concerns

- ODPS defines the data product.
- ODPC defines reusable portfolio objects and catalogs.
- ODPV defines shared terminology and semantics.
- ODPG defines relationships between products, use cases, objectives, policies and other objects.
- ODPR defines repeatable work and workflow contracts around those artifacts.



Figure 6. A product contract is the stable centre of a connected definition: identity and version bind semantics, conditions, relationships and interfaces together.

C4. Product Definition and Contract

Purpose

Establish an authoritative machine-readable definition of the data product.

Operating intent

The product contract is the stable point of agreement between provider and consumer. It should be independently identifiable, versioned and sufficiently complete for a consumer to determine what the product offers, who is accountable for it, how it can be accessed and which commitments and conditions apply. A catalog page or platform configuration may present parts of that information, but neither should silently become the only authoritative definition.

The contract should describe the product rather than the internal project that created it. Implementation details belong where they affect consumption, commitments or change impact; temporary delivery tasks do not. When information is maintained in another authoritative artifact, the contract should use a stable reference and define what happens if that reference cannot be resolved.

Human-readable pages should be generated from or demonstrably aligned with the same machine-readable definition. Parallel hand-maintained descriptions create ambiguity precisely when a consumer needs certainty about a version, interface or commitment.

Practices

- | | |
|---------------------------------------|---|
| 1 Assign product identity. | 2 Define the product proposition. |
| 3 Declare provider responsibility. | 4 Define product type. |
| 5 Define data and delivery structure. | 6 Define lifecycle state. |
| 7 Version the product definition. | 8 Declare product-local business context. |
| 9 Validate the product definition. | 10 Approve and publish the definition. |

Primary standard

ODPS.

Evidence

- | | |
|-------------------------------------|----------------------------|
| ■ validated ODPS artifact | ■ version history |
| ■ approval record | ■ schema validation result |
| ■ publication record | ■ change history |
| ■ objective and use-case references | |

Assurance questions

- | | |
|---|--|
| ? | Can a consumer distinguish the product identity from the catalog record, dataset and delivery interface? |
| ? | Is there one authoritative version of the contract and a defined approval authority? |

? Can the contract be validated without depending on a particular catalog vendor?

? Do human and machine presentations resolve to the same meaning and version?

C5. Semantics and Vocabulary

Purpose

Ensure that people and systems, including AI agents where used, interpret product concepts consistently.

Operating intent

Semantic management should focus first on concepts whose ambiguity changes a decision, rule, calculation or integration. Not every label needs central governance. Material terms—such as customer, active account, service region or reportable incident—need stable identity, a clear definition, accountable stewardship and an explicit relationship to local or external terms.

Labels and aliases support discovery, but they are not substitutes for concept identity. A renamed label should not silently create a new concept, and two identical labels should not be assumed to have identical meaning. Where multilingual labels are provided, the underlying identifier and definition should preserve equivalence or disclose the difference.

Semantic consistency must be checked across contracts, catalogs, graphs, policies and workflow inputs. A vocabulary is useful only when implementations actually reference it and changes are governed.

Core rule

Structure tells a machine where information is.

Semantics tells it what the information means.

Practices

- | | |
|--|--------------------------------|
| 1 Identify controlled concepts. | 2 Reuse existing definitions. |
| 3 Assign stable concept identifiers. | 4 Maintain labels and aliases. |
| 5 Define relationships between concepts. | 6 Map external vocabularies. |
| 7 Apply vocabulary across standards. | 8 Detect semantic drift. |
| 9 Expose vocabulary for machines. | |

Primary standard

ODPV.

Evidence

- approved vocabulary
- semantic mappings
- terminology decisions
- vocabulary version
- validation reports
- controlled-term references

Assurance questions

- ? Which concepts would create material risk if interpreted differently by two consumers?
- ? Are labels, aliases, definitions and identifiers managed separately?
- ? Can a reviewer trace where a concept is used across product and governance artifacts?
- ? Are mappings to external vocabularies qualified rather than treated as automatic equivalence?

C6. Product Commitments and Usage Conditions

Purpose

Define what consumers should expect from the product and the conditions under which it is provided and used.

Operating intent

A commitment should identify the subject, metric or rule, threshold, evaluation period, measurement method, scope and consequence of failure. “High quality” and “available when needed” are aspirations, not testable commitments. A product may also disclose non-binding objectives, but consumers must be able to distinguish them from approved obligations.

Usage conditions should be understandable before access is granted. They may include permitted purposes, prohibited uses, attribution, retention, privacy, security, geographic, licensing or commercial conditions. Machine-readable expression improves consistency and automated evaluation, but expression alone does not create legal validity or operational enforcement.

The provider should manage conflicts between conditions, record approved exceptions and identify which rule prevails. A declared rule, an enforcement configuration and evidence that the rule operated are three separate objects.

Practices

- 1 Define quality commitments.
- 2 Define service commitments.
- 3 Define access conditions.
- 4 Define usage conditions.
- 5 Define legal terms.
- 6 Define privacy and security expectations.

- | | |
|-------------------------------------|---|
| <p>7 Define commercial terms.</p> | <p>8 Reuse standard condition patterns.</p> |
| <p>9 Make commitments testable.</p> | <p>10 Manage exceptions.</p> |

Primary standard

ODPS.

Evidence

- | | |
|--|--|
| <ul style="list-style-type: none"> ■ machine-readable ODPS commitments ■ SLA rules ■ license reference ■ pricing plan where relevant ■ exception record | <ul style="list-style-type: none"> ■ quality rules ■ access profile ■ contract reference ■ policy mapping ■ validation evidence |
|--|--|

Assurance questions

- ? Can each material commitment be evaluated using a defined observation and period?
- ? Can consumers distinguish binding conditions, service objectives and descriptive guidance?
- ? Are policy conflicts, exceptions and precedence decisions recorded?
- ? Does the product state what happens when a commitment is missed or a condition changes?

C7. Relationships, Dependencies and Context

Purpose

Connect the individual data product to the wider network of objectives, use cases, KPIs, policies, systems, products, APIs, workflows, consumers and organisational responsibilities.

Operating intent

Relationships turn isolated records into operational context. Each material relationship should identify both endpoints, use a defined relationship type and retain enough provenance to explain who asserted it, from which evidence, when and with what confidence. A free-text note that “product A depends on product B” is difficult to validate, traverse or use for impact analysis.

Not every possible edge belongs in the shared graph. Include relationships that support a decision, control, discovery path, dependency analysis, workflow or value chain. Derived relationships should be distinguishable from confirmed relationships, and conflicting assertions should remain reviewable rather than being silently overwritten.

Graph quality includes more than valid syntax. References must resolve, relationship direction must be meaningful, lifecycle states must be compatible and critical paths must have accountable owners. These checks allow a change in an objective, policy, interface or upstream product to be traced to affected consumers and outcomes.

Core rule

A product definition answers:

What is this product?

A relationship graph answers:

How does this product fit into everything else?

Practices

1 Identify relevant relationship types.	2 Connect products to demand.
3 Connect demand to objectives.	4 Connect products to outcomes.
5 Connect dependencies.	6 Connect governance context.
7 Connect technical context.	8 Connect consumer and automation context where relevant.
9 Record relationship confidence.	10 Review graph integrity.

Primary standard

ODPG.

Evidence

- validated graph artifact
- confirmed edges
- broken-reference report
- value-path analysis
- relationship provenance
- graph review record
- dependency review

Assurance questions

- ? Can critical product dependencies be traversed in both directions?
- ? Does each material relationship have type, provenance, status and review responsibility?

? Are inferred or low-confidence relationships visibly different from confirmed assertions?

? Can the graph answer an impact question, not merely display connected nodes?

DEFINE exit criteria

A product moves into normal operational management when:

- an accountable portfolio decision exists
- the product has an approved identity
- an authoritative product definition exists
- required semantics are resolved
- relevant quality and service commitments exist
- access and usage conditions are defined
- required relationships exist
- the artifact package passes validation
- required approvals are recorded

Exit is a controlled transition, not a claim that the definition is permanently complete. Any unresolved assumption, exception or planned semantic decision should be recorded with an owner and due date. Material changes after the transition return through the relevant DEFINE and lifecycle controls.

Evidence basis and external references

The framework requirements above remain Open Data Product Initiative decisions. These sources support the implementation reasoning:

- **w3c-dcat-3** — W3C Data Catalog Vocabulary 3 provides interoperable patterns for datasets, services, distributions, catalog records, identifiers and relationships. It informs catalog interoperability but is not a substitute for an ODPS product contract.
- **wilkinson-fair-principles-2016** — FAIR Guiding Principles emphasise persistent identifiers, rich metadata, provenance and machine actionability. The framework applies those ideas selectively; a FAIR digital object is not automatically a governed data product.
- **w3c-skos** — W3C SKOS distinguishes concepts, labels, semantic relationships and mappings. It supports the vocabulary guidance without requiring every implementation to use RDF.
- **w3c-odrl-2.2** — W3C ODRL Information Model 2.2 separates permissions, prohibitions, duties and constraints. It informs machine-readable usage conditions but does not prove that a policy was enforced.
- **w3c-prov-o** — W3C PROV-O represents entities, activities, agents and provenance relationships. It informs relationship and evidence provenance without redefining ODPG.
- **w3c-shacl** — W3C SHACL demonstrates how machine-readable constraints can produce validation results. It is an implementation option, not a mandatory validation technology for the Core.

FUNCTION 3

OPERATE

Purpose

OPERATE turns defined data products into available, consumable and managed services.

Operating principle

Declaration and execution are separate responsibilities.

DECLARE → EXECUTE → EVIDENCE

The ODPS standards family declares portable contracts.

Operational platforms execute them.

Runtime systems produce evidence.

C8. Catalog, Publication and Discovery

Purpose

Make governed data products and surrounding business context discoverable by people, applications and automated systems, including AI agents where used.

Operating intent

Publication should help a consumer decide whether a product is suitable, not merely confirm that a record exists. Discovery information therefore needs product identity, proposition, provider, lifecycle state, version, relevant semantics, access path, commitments, usage conditions and relationships to demand or alternatives. Search relevance and business language matter alongside technical metadata.

The catalog is an index and presentation surface, not necessarily the authority for every field. It should preserve resolvable links to authoritative product, vocabulary, graph and policy artifacts and disclose when indexed information was last synchronized. Federation must not hide conflicting identifiers, stale records or different visibility rules.

Machine discovery requires structured metadata and a stable interface. Free-text catalog pages may help people, but automated consumers should not have to infer lifecycle state, access conditions or compatibility from prose alone.

Practices

- | | |
|------------------------------|--------------------------------------|
| 1 Register the product. | 2 Maintain authoritative references. |
| 3 Publish portfolio context. | 4 Apply visibility controls. |

- | | |
|--|--|
| <p>5 Support business-context search.</p> <hr/> <p>7 Federate where required.</p> <hr/> <p>9 Manage catalog lifecycle.</p> | <p>6 Support machine discovery.</p> <hr/> <p>8 Validate catalog integrity.</p> |
|--|--|
-

Primary standard

ODPC.

Evidence

- | | |
|--|--|
| <ul style="list-style-type: none"> ■ valid ODPC catalog ■ publication record ■ search index record ■ broken-reference report | <ul style="list-style-type: none"> ■ resolvable ProductReference ■ catalog validation result ■ visibility configuration |
|--|--|

Assurance questions

- ? Can intended consumers discover a product using their business problem and terminology?

 - ? Can they identify current, deprecated and unsuitable products before requesting access?

 - ? Does each catalog record resolve to authoritative artifacts and disclose synchronization state?

 - ? Can machines retrieve structured discovery metadata without screen scraping?
-

C9. Provisioning, Integration and Consumption

Purpose

Turn a defined product interface into controlled consumer access.

Operating intent

Provisioning begins with an identified consumer and an intended use, then evaluates entitlement against the product's access and usage conditions. Approval is one possible control; it should not be added where an automatic policy decision is sufficient, nor removed where accountable judgement or risk acceptance is required.

Access is not complete when credentials are issued. The consumer needs the interface version, endpoint or delivery location, authentication method, semantic context, operating limits, support path and a successful connectivity or acceptance check. These details should be versioned where a change could affect consumption.

Consumption evidence should associate activity with the product, version and consumer identity at an appropriate level of granularity. Privacy, security and proportionality constraints still apply: the framework does not require invasive tracking of individual behaviour to establish product adoption.

Practices

1 Resolve the access contract.	2 Identify the consumer.
3 Evaluate entitlement.	4 Provision access.
5 Provide implementation context.	6 Validate connectivity.
7 Associate consumption with the product.	8 Manage consumer changes.
9 Revoke access.	10 Support automated consumers where used.

Primary standard

ODPS.

Supporting standard

ODPR for implementation handoff and onboarding workflows.

Evidence

- access request
- provisioning record
- connectivity test
- subscription record
- consumption event
- approval where required
- entitlement record
- product version consumed
- revocation record

Assurance questions

- ? Can every active entitlement be traced to a consumer, product version and applicable condition?
- ? Is the provisioning decision reproducible from policy and approval evidence?
- ? Does onboarding verify usable access rather than merely issuing credentials?



Can access be changed or revoked when the consumer, purpose or product state changes?

C10. Lifecycle, Version and Change Management

Purpose

Control the evolution of a data product from initial development through production, change, deprecation and retirement.

Operating intent

Versioning communicates consumer-relevant meaning. An organisation should define what constitutes a breaking, compatible and editorial change for its product types and apply that classification consistently. A changed implementation does not always require a new public contract version, while a changed meaning, interface, commitment or condition often does.

Impact analysis should use known consumers, dependencies, workflows, controls and value paths rather than relying only on the delivery team's judgement. The change record should identify the prior and proposed state, affected relationships, validation evidence, decision authority, release timing and recovery approach.

Deprecation is a managed period, not a label applied shortly before shutdown. Consumers need a supported alternative or explicit exception path, migration expectations and an end date. Retirement should revoke access, close or transfer obligations, update discovery records and retain enough evidence to explain the decision later.

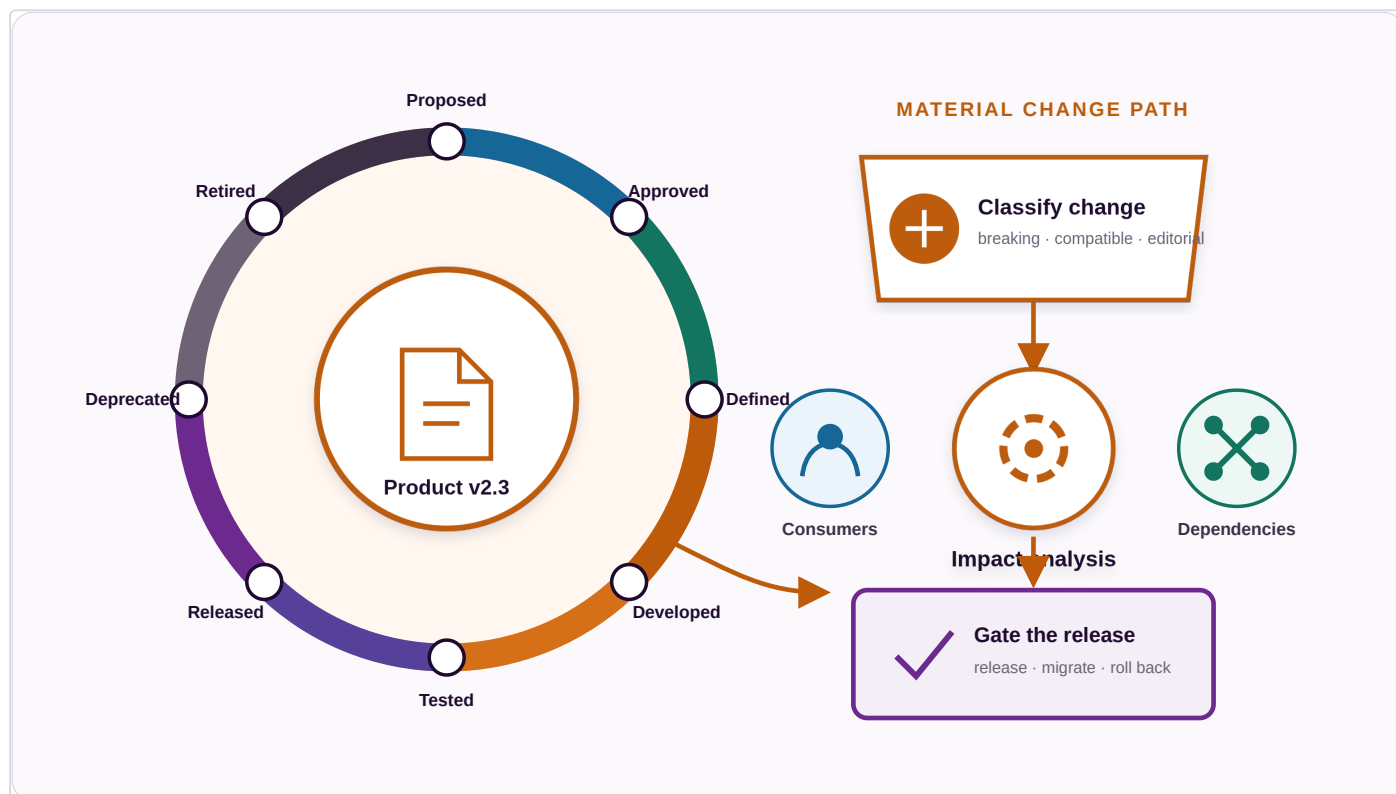


Figure 7. Product change is a governed lifecycle path: material changes are classified, tested against their impact and either released, migrated or rolled back.

Practices

1 Maintain lifecycle state.

2 Version the product.

3 Classify changes.

4 Analyse impact.

5 Validate the change.

6 Apply change gates.

7 Release the change.

8 Communicate change.

9 Deprecate safely.

10 Retire products.

Lifecycle baseline

- announcement
- draft
- development
- testing
- acceptance
- production
- sunset
- retired

Evidence

- | | |
|---|-------------------------|
| ■ previous product version | ■ new product version |
| ■ version difference | ■ change classification |
| ■ impact analysis | ■ validation results |
| ■ gate results | ■ approval |
| ■ publication result | ■ consumer notification |
| ■ rollback or remediation record where required | |

Assurance questions

- ? Are change classes defined from the consumer's perspective?
- ? Can the organisation identify affected consumers and downstream dependencies before release?
- ? Are supported, deprecated and retired versions unambiguous in both contract and runtime state?
- ? Does retirement close access, obligations, catalog records and operational monitoring deliberately?

C11. Workflow, Automation and Agent Operations

Purpose

Make repeatable data product work explicit, portable, bounded and reviewable.

Operating intent

A workflow contract should state the intended outcome, authoritative inputs, ordered activities, decision points, expected outputs, validation gates, accountable roles and evidence to retain. It should remain distinguishable from a specific execution engine so the organisation can review or move the procedure without losing its meaning.

Human-led, application-led and agent-assisted execution are all valid Core implementations. Automation is appropriate when inputs, rules, permissions and failure handling are explicit. Where judgement or risk acceptance is material, the workflow should identify the human decision and the information needed to make it rather than disguising the decision as an automated step.

Execution records should identify the workflow version, input versions, actor or system identities, decisions, exceptions, outputs and final status. A successful technical run does not by itself show that the business outcome was correct; ASSURE evaluates service, control and outcome evidence separately.

Practices

- | | |
|-----------------------------------|--|
| 1 Identify repeatable work. | 2 Declare workflow intent. |
| 3 Declare inputs. | 4 Declare ordered activities. |
| 5 Declare outputs. | 6 Define gates. |
| 7 Define human review. | 8 Bound automated and agent work where used. |
| 9 Separate contract from runtime. | 10 Record execution evidence. |
| 11 Manage workflow versions. | 12 Reuse proven workflows. |

Primary standard

ODPR.

Operational patterns

delivery flows

product handoff flows

machine discovery flows

trigger-based flows

Evidence

ODPR recipe

recipe version

- | | |
|-------------------------|--------------------|
| ■ input manifest | ■ execution record |
| ■ validation result | ■ gate result |
| ■ human review decision | ■ output manifest |
| ■ exception record | ■ run outcome |

Assurance questions

- Is the portable workflow intent separable from the selected orchestration platform?
- Are inputs, permissions, gates, stopping conditions and exception paths explicit enough to review?
- Can each execution be tied to the workflow and artifact versions actually used?
- Does automation preserve accountable decisions rather than merely remove visible human steps?

Declared state versus observed state

OPERATE introduces a formal comparison between:

- what the product contract or workflow says should happen
- what operational systems show happened

This difference becomes evidence for ASSURE.

OPERATE should preserve both sides of the comparison. Updating a declaration to match a failure can erase the signal; retaining observations without the applicable declaration removes the evaluation basis. Version and time therefore matter for contracts, policies, workflows and evidence.

Evidence basis and external references

The framework requirements above remain Open Data Product Initiative decisions. These sources provide supporting patterns:

- **w3c-dcat-3** — W3C Data Catalog Vocabulary 3 provides portable catalog, dataset, distribution and data-service structures as well as catalog-record lifecycle metadata. It informs federation and discovery but does not replace ODPC or ODPS.
- **wilkinson-fair-principles-2016** — FAIR Guiding Principles support persistent identifiers, rich metadata and machine-actionable discovery while preserving the distinction between accessibility and unrestricted access.
- **w3c-prov-o** — W3C PROV-O provides a general model for entities, activities, agents and derivation. It informs workflow and execution provenance without requiring a particular runtime platform.
- **nist-sp-800-53r5** — NIST SP 800-53 Rev. 5 contains established control patterns for access, configuration, change, monitoring and evidence. Organisations should select applicable controls rather than treating the catalog as a universal checklist.
- **nist-ai-rmf-1.0** and **nist-ai-600-1-genai-profile** — NIST AI RMF 1.0 and its Generative AI Profile inform bounded AI operations, inventory, measurement and oversight where AI is used. They do not make AI or agents a Core requirement.

FUNCTION 4

ASSURE

Purpose

ASSURE determines whether data products operate as defined, satisfy applicable governance requirements, are adopted by intended consumers and create enough value to justify continued investment.

Assurance principle

Do not confuse definition with evidence.

A declared quality target is not evidence of quality.

A published SLA is not evidence of service performance.

A policy reference is not evidence that a control worked.

A use case is not evidence of adoption.

A business objective is not evidence of value.

C12. Observability and Service Assurance

Purpose

Determine whether the operating data product satisfies the measurable quality, service and operational commitments made in its definition.

Operating intent

Observability connects a measurement to the exact product, version, commitment and evaluation period to which it applies. A dashboard value without this context may be useful operational telemetry, but it is weak assurance evidence. The measurement method, sampling window, exclusions and status logic should be explicit enough for another reviewer to reproduce or challenge the result.

Service assurance should combine technical observations with consumer impact. A short availability failure may be immaterial for one use case and decisive for another; a quality average may hide a critical segment. Evaluation should therefore preserve the declared threshold while allowing impact, severity and approved exceptions to be assessed separately.

The measurement system also requires assurance. Missing telemetry, changed metric definitions, delayed observations and unmonitored interfaces should be visible rather than interpreted as successful performance.

Practices

1 Identify measurable commitments.

2 Define measurement methods.

- | | |
|---|--|
| <p>3 Collect observations.</p> <hr/> <p>5 Evaluate service performance.</p> <hr/> <p>7 Assess consumer impact.</p> <hr/> <p>9 Retain evidence.</p> <hr/> <p>11 Feed findings into product management.</p> <hr/> | <p>4 Evaluate data quality.</p> <hr/> <p>6 Detect deviations.</p> <hr/> <p>8 Trigger response.</p> <hr/> <p>10 Review measurement quality.</p> <hr/> |
|---|--|

Primary standard

ODPS for declared quality and service objectives.

Evidence

- | | |
|--|---|
| <ul style="list-style-type: none"> ■ measurement result ■ product ID ■ declared objective ■ measurement method ■ incident reference ■ exception where applicable | <ul style="list-style-type: none"> ■ monitoring timestamp ■ product version ■ observed result ■ evaluation status ■ remediation record |
|--|---|

Assurance questions

- ? Is every result tied to the product, contract version, metric and evaluation period in force?
-
- ? Can a reviewer reproduce the status from the recorded observations and method?
-
- ? Are missing, delayed or invalid measurements distinguished from passing results?
-
- ? Does incident and impact evidence reach the owner who can change the product or commitment?
-

C13. Governance, Risk, Compliance and Control Assurance

Purpose

Determine whether applicable governance obligations have been translated into controls and whether those controls operate effectively.

Operating intent

Control assurance begins with applicability. The organisation should be able to explain why an obligation, policy and control applies to a product and which authority accepted that interpretation. Copying a standard control catalog into every product obscures accountability and makes evidence review unmanageable.

A control definition should identify its objective, owner, trigger or frequency, scope, method, expected result, evidence and response to failure. Control execution then produces an observation about a particular product and version. Design approval, execution evidence and effectiveness assessment are different stages and should not be collapsed into one “compliant” status.

Exceptions are governed decisions with scope, rationale, compensating measures, accountable acceptance and expiry. Findings should connect to remediation and retest evidence. Independent assurance may sample evidence or repeat tests, but independence and sampling limits must be disclosed.

Core chain

Requirement → Policy → Control → Product → Execution → Evidence → Finding → Response

Practices

- | | |
|------------------------------------|-----------------------------------|
| 1 Identify applicable obligations. | 2 Map obligations to policies. |
| 3 Map policies to controls. | 4 Map controls to products. |
| 5 Assign control ownership. | 6 Execute controls. |
| 7 Produce control evidence. | 8 Assess risk. |
| 9 Manage exceptions. | 10 Test control effectiveness. |
| 11 Manage findings. | 12 Support independent assurance. |
| 13 Retain traceability. | |

Evidence

- applicable obligation
- control definition
- execution timestamp
- control outcome
- risk decision
- finding
- independent review where required
- policy version
- control owner
- product version
- supporting artifact
- exception approval
- remediation result

Assurance questions

- ❓ Can each material control be traced to an applicable requirement and accountable interpretation?
- ❓ Does the evidence show that the control executed, not merely that it was designed or configured?
- ❓ Are exceptions time-bound, scoped, accepted and reviewed before expiry?
- ❓ Can findings be followed through remediation, retest and closure?

C14. Adoption, Outcomes and Value Realisation

Purpose

Determine whether consumers use the data product and whether that consumption contributes to the outcome that originally justified investment.

Operating intent

Adoption is meaningful use by an intended consumer, not an account count or isolated access event. The organisation should define what meaningful use means for each use case and separate initial trial, recurring consumption, production dependency and reuse by an additional use case.

Outcome measurement compares the present state with a baseline, target or credible counterfactual. It should record the period, population, calculation, assumptions and other changes that may explain the result. Attribution must be proportionate to the decision: a small operational improvement may use contribution evidence, while a major investment claim may need stronger causal analysis.

Value combines the outcome with its significance, cost and attribution. Financial value may be appropriate, but risk reduction, compliance, public value, decision quality and service improvement should not be forced into artificial revenue estimates. A mandatory product can have value even when stopping it is not a realistic option; the portfolio decision may instead concern cost, quality or risk.

Value chain

Product exists → Product is discoverable → Consumer receives access → Consumer uses product → Use case changes → Outcome changes → Value is realised

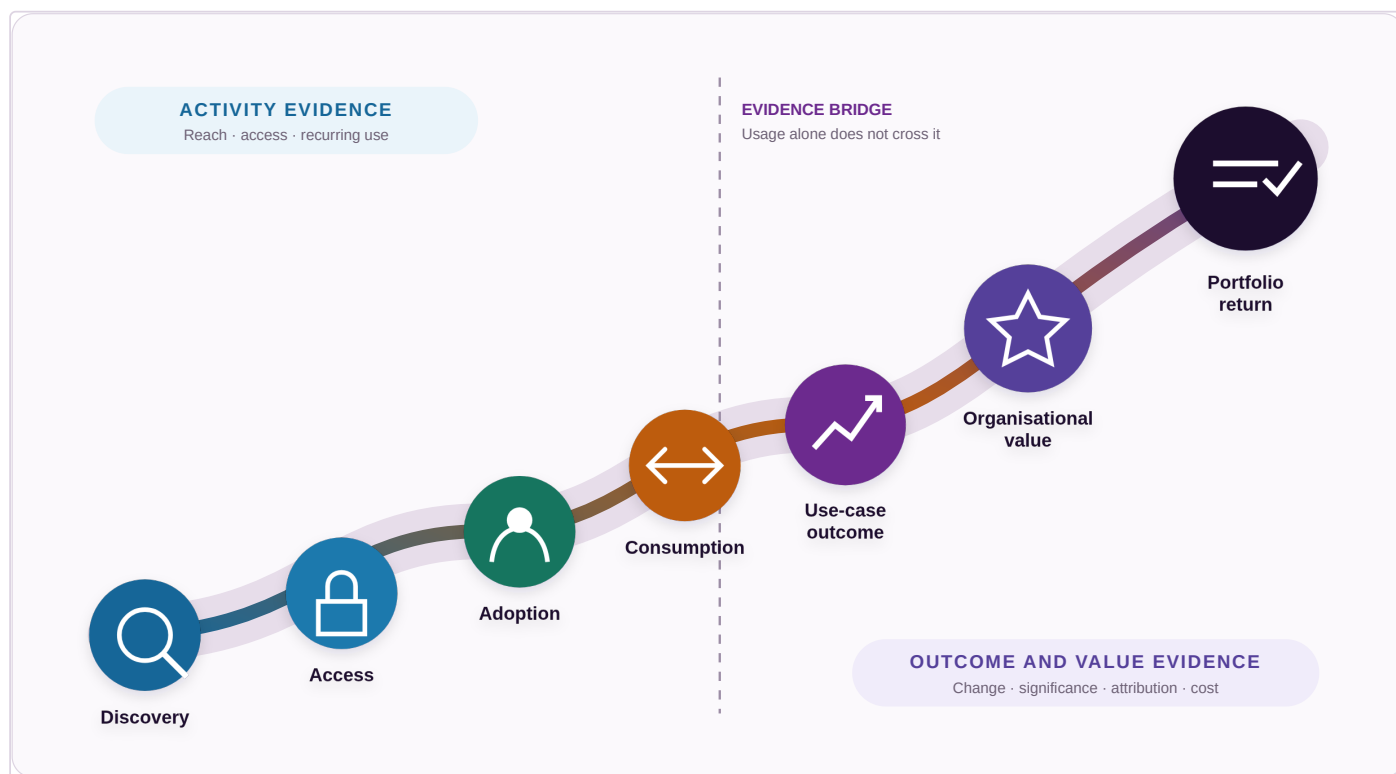


Figure 8. Discovery, access and consumption demonstrate activity; value is established only when evidence connects that activity to use-case and organisational outcomes.

Practices

- 1 Define expected value before investment.
- 2 Establish a baseline.
- 3 Measure discovery and access.
- 4 Measure adoption.
- 5 Measure sustained consumption.
- 6 Measure reuse.
- 7 Connect usage to use cases.
- 8 Measure use-case outcomes.
- 9 Measure business value.
- 10 Measure product cost.
- 11 Assess value contribution.
- 12 Review portfolio investment.
- 13 Retire unsupported products.

Measurement levels

Reach

Can intended consumers find and access the product?

Adoption

Have intended consumers started meaningful use?

Engagement and Reuse

Does meaningful consumption continue?

Use-Case Outcome

Did the supported process or decision improve?

Organisational Value

What is that improvement worth?

Portfolio Return

Was investing in this product preferable to alternative uses of resources?

Value classes

financial value

operational value

decision value

risk value

compliance value

public value

Attribution classes

direct attribution

contribution

enablement

mandatory value

Evidence

- active consumer record
- use-case relationship
- current outcome measure
- cost record
- assumptions
- portfolio review decision
- consumption events
- baseline measure
- KPI result
- benefit calculation
- attribution method

Assurance questions



Is meaningful adoption defined for the relevant consumer and use case?



Can usage be connected to a change in a decision, process, obligation or service outcome?



Are baseline, target, period, assumptions, cost and attribution method visible?



Would the investment decision change if the reported value were lower or more uncertain?

Four assurance views

Service Health

Does the product perform as promised?

Control Health

Does it operate within required governance boundaries?

Adoption Health

Are intended consumers using it?

Value Health

Are intended outcomes being achieved?

These should not be collapsed into one opaque score.

A combined executive view may summarize the four perspectives, but it should preserve drill-down to the underlying measures, evidence, exceptions and decision owners. A healthy service can still be unused, a compliant product can still lack value, and a valuable product can still require urgent control remediation.

Evidence basis and external references

The framework requirements above remain Open Data Product Initiative decisions. These sources support the assurance model:

- [w3c-dqv](#) — W3C Data Quality Vocabulary separates quality dimensions, metrics, measurements, annotations and policy context. It informs machine-readable quality evidence without making RDF mandatory.
- [w3c-prov-o](#) — W3C PROV-O supports attribution of evidence to entities, activities and agents. Provenance improves reviewability but does not by itself prove that a claim is true.
- [w3c-shacl](#) — W3C SHACL distinguishes constraints from validation reports and individual validation results. This supports the framework's declared-versus-observed separation.
- [nist-csf-2.0](#), [nist-privacy-framework-1.0](#) and [nist-sp-800-53r5](#) — NIST Cybersecurity Framework 2.0, NIST Privacy Framework 1.0 and NIST SP 800-53 Rev. 5 provide risk, control and assessment patterns. They inform applicable control design and evidence; they do not establish universal legal obligations for every product.
- [oecd-ai-principles-2024](#) — OECD AI Principles emphasise beneficial outcomes, accountability, robustness and lifecycle risk management for AI. They are relevant when AI participates in the product environment, while C14 continues to measure organisational outcomes rather than AI activity counts.

EVALUATION

Assessment Standard

1. Purpose

The assessment standard defines how organisational capabilities are evaluated.

Assessment is capability-based.

A valid machine-readable artifact does not by itself prove organisational maturity.

2. Assessment dimensions

Each capability is evaluated across four dimensions.

Accountability

Are ownership and decision rights explicit?

Practice

Is there a defined and repeatable way of performing the capability?

Evidence

Can the organisation demonstrate that the capability operates?

Outcome

Does evidence show that the capability achieves its purpose?

The four dimensions should be scored separately before assigning the overall capability level. Strong documentation cannot compensate for missing accountability, and extensive activity cannot compensate for missing outcome evidence. An assessor should record the limiting dimension and the improvement needed to progress.

3. Capability levels

Level 0: Absent

The capability is not established.

Level 1: Emerging

Activity occurs inconsistently and depends heavily on individuals.

Level 2: Defined

Roles, practices and expected outputs exist.

Level 3: Operational

Practices operate consistently and produce evidence.

Level 4: Measured

Performance, exceptions and outcomes are measured.

Level 5: Adaptive

Evidence systematically drives improvement, decisions and suitable automation.

4. Assessment evidence classes

Evidence can include:

- directive evidence
- definition evidence
- execution evidence
- operational evidence
- outcome evidence

Evidence sufficiency

Evidence should be relevant to the capability claim, attributable to a known source, tied to the applicable period and version, and retained in a form that can be reviewed. The assessor should distinguish:

- declared evidence — definitions, policies, plans, contracts and expected practices
- execution evidence — records that the defined practice was performed
- observed evidence — measurements and events showing what occurred
- decision evidence — findings, approvals, exceptions and investment responses based on the observations

A template, policy or configured control is not execution evidence. One successful execution may demonstrate existence but not consistent operation. A dashboard may show observations while providing insufficient provenance or method to support assurance.

Sampling should reflect the frequency, variability and consequence of the capability. Assessors should record the population, period, sample method, exclusions and known limitations. Evidence supplied only for the assessment should be treated cautiously if it is not produced by normal operations.

5. Assessment method

For each capability:

1. Identify accountable roles.
2. Review documented practices.
3. Inspect required artifacts.
4. Inspect evidence that practices operated.

5. Evaluate whether intended outcomes are measured.
6. Score each assessment dimension.
7. Determine the capability level.
8. Record findings and improvement actions.

The assessment scope should identify the organisation, portfolio, product population, profile, period and exclusions. Interviews can establish intent and explain evidence, but statements should not substitute for operating records where those records should exist.

Assessors should test forward and backward traceability. A forward test starts from an objective or use case and follows the chain to a product, execution, evidence and outcome. A backward test starts from a value or conformance claim and verifies the evidence, consumption, product version and original decision that support it.

Where evidence conflicts, record the conflict rather than selecting the more favourable source. Where a capability differs materially between domains or product classes, report the distribution and rationale instead of relying only on an average.

6. Capability assessment result

Organisations should receive capability-level assessment results rather than one opaque maturity score.

Each result should include:

- capability level and scores for Accountability, Practice, Evidence and Outcome
- assessment scope and applicable profile
- evidence reviewed and sampling method
- strengths and observed outcomes
- gaps, exceptions and conflicting evidence
- improvement actions, owners and target dates
- assessor and assessment date
- confidence and material limitations

Example:

Capability	Level
C1 Strategy and Objectives	4
C2 Demand and Use Case Management	3
C3 Portfolio, Investment and Accountability	2
C4 Product Definition and Contract	5
C5 Semantics and Vocabulary	2
C6 Product Commitments and Usage Conditions	4
C7 Relationships, Dependencies and Context	3
C8 Catalog, Publication and Discovery	4
C9 Provisioning, Integration and Consumption	3
C10 Lifecycle, Version and Change Management	2
C11 Workflow, Automation and Agent Operations	4
C12 Observability and Service Assurance	4
C13 Governance, Risk, Compliance and Control Assurance	3
C14 Adoption, Outcomes and Value Realisation	1

7. Conformance types

Profiles use the same capability levels and assessment dimensions as the Core.

Profile requirements add context-specific expectations for practices, artifacts, evidence and controls within the applicable Core capability. A profile must not rename capabilities, create a separate maturity model or treat automation as evidence of maturity.

Artifact conformance

A machine-readable artifact complies with the applicable specification.

Practice conformance

An organisation follows required operating practices.

Capability conformance

Evidence demonstrates that the capability operates and achieves required outcomes.

Conformance and maturity answer different questions. An artifact can conform while the surrounding capability is weak. A mature capability can manage several artifact formats while one particular artifact fails validation. Reports should state the object and scope of each conclusion rather than using an unqualified “compliant” label.

8. Assessment integrity

Assessors should be sufficiently independent from the activity being assessed for the consequence of the conclusion. Self-assessment is useful for improvement, but it should be labelled as such. Claims used for certification, regulatory response or material investment decisions may require independent evidence review.

Automation may validate artifacts, collect observations and detect missing links. It must not silently decide ambiguous applicability, risk acceptance or outcome attribution. Automated results should preserve rule version, input version, execution time and exceptions so a person can examine how the conclusion was reached.

Calibration is required when multiple assessors or organisational units are involved. A common evidence pack and example scoring decisions should be used to test whether assessors interpret levels consistently.

9. Evidence basis and future work

Library sources [nist-csf-2.0](#), [nist-sp-800-53r5](#) and [w3c-shacl](#) support this section. NIST Cybersecurity Framework 2.0 supports outcome-oriented assessment while allowing implementation to vary by context. NIST SP 800-53 Rev. 5 provides established control-assessment concepts, and W3C SHACL demonstrates the separation between declared constraints and validation results. These sources inform the assessment logic but do not define conformance to this framework.

A later version should define:

- required assessment questions per capability
- minimum evidence per level
- scoring rules
- assessor guidance
- sampling rules
- reassessment rules
- organisational certification requirements

ADOPTION

Implementation Playbook

1. Purpose

The playbook provides practical guidance for implementing the framework.

The Core defines what good data product operations require.

The playbook explains how to establish them.

Implementation should begin with a bounded portfolio or value stream, not an enterprise-wide metadata exercise. Select a scope in which accountable decision-makers, product providers and consumers can establish one complete traceability chain and use its evidence to make a real continuation or change decision.

Before starting, record the implementation sponsor, scope, intended profile, product population, current systems, material obligations, success measures and decision date. This baseline prevents tooling activity from being mistaken for operating-model adoption.

2. Recommended implementation sequence

Step 1. Establish DIRECT

Start with:

- organisational objectives
- product portfolio
- investment decisions
- use cases and demand
- accountable ownership

Do not begin by creating hundreds of ODPS files for products that have no clear business demand.

Choose one objective with an accountable owner and one or more active use cases. Confirm how a use-case outcome will be observed and who can approve, defer or reject a product candidate. Establish a lightweight portfolio decision record before designing the product contract.

Step 2. Establish DEFINE

For approved product candidates:

- create authoritative ODPS product contracts
- define commitments and conditions
- align vocabulary through ODPV
- connect products to portfolio context through ODPG

Begin with the smallest contract that lets an intended consumer understand and evaluate the product. Resolve material semantic ambiguity and usage conditions first. Record unresolved decisions explicitly instead of hiding them in prose or implementation defaults.

Step 3. Establish OPERATE

Connect definitions to:

- catalogs
- delivery platforms
- change processes
- access provisioning
- lifecycle management
- reusable workflows

Use the contract identifiers and versions in catalog, provisioning, release and workflow records. The objective is not immediate platform replacement; it is traceability across existing systems. Where integration is manual, define the accountable handoff and evidence before automating it.

Step 4. Establish ASSURE

Collect evidence for:

- service performance
- control execution
- use-case outcomes
- quality
- adoption
- value

Select a small set of observations tied directly to declared commitments and the original use-case outcome. Confirm that missing data and failed measurements are visible. Produce an assurance view that can support a product or portfolio decision, rather than a dashboard that only reports activity.

Step 5. Close the loop

Use ASSURE evidence to change:

- product commitments
- portfolio investment
- strategic priorities
- operational practices
- product lifecycle decisions

Hold the first review while the implementation scope is still small. Use evidence to make at least one explicit decision: continue, change, expand, merge, constrain or retire. If the review cannot change a decision, revisit the accountability and measures before scaling.

3. Minimum viable implementation

An initial implementation should establish at least:

- one governed business objective
- one registered use case
- one approved product decision
- one valid ODPS product contract
- one catalog reference
- one relationship graph connecting objective, use case and product
- one operational workflow
- one service or quality observation
- one adoption measure
- one outcome measure
- one portfolio review decision

This creates one complete traceability chain before scaling.

The minimum viable implementation is complete only when the chain has been exercised. Creating the artifacts without provisioning, observation and review demonstrates definition work, not an operating framework.

4. Adoption waves

Wave 1. Prove the chain

Implement one objective-to-value chain for a bounded product set. Accept manual integration where responsibilities and evidence are explicit.

Wave 2. Establish repeatability

Standardise identifiers, templates, decision records, validation and evidence packaging. Train owners and consumers, then repeat the process across several products or domains.

Wave 3. Connect systems

Integrate portfolio, contract, catalog, provisioning, lifecycle and observability systems. Remove duplicate entry where an authoritative artifact can drive another view.

Wave 4. Automate assurance

Automate stable validations, comparisons and evidence collection. Preserve human authority for material investment, risk and exception decisions.

Wave 5. Adapt by evidence

Use recurring capability assessment and product evidence to improve practices, strengthen selected profile requirements and change portfolio investment.

5. Source-of-truth guidance

Use machine-readable standards as portable sources of truth where appropriate.

Do not let a UI become the only source of truth.

Generated HTML, PDF and catalog views should derive from underlying artifacts.

For each material field, decide which artifact or system has authority and how other systems receive updates. Avoid “bidirectional synchronization” as a default answer; without field-level authority and conflict rules, it creates multiple competing truths.

Migration can use temporary reconciliation reports. Record the old source, new source, transformation rule, unresolved differences and cutover decision. Preserve identifiers so historical evidence remains connected after systems change.

6. Automation guidance

Automate where the rule is clear.

Examples:

- schema validation
- contract validation
- graph integrity checks
- catalog publication
- SLA evaluation
- quality checks
- dependency impact checks
- evidence packaging

Keep human approval where judgement, risk acceptance or investment authority requires it.

Automate only after the manual or semi-automated practice has a clear input, rule, output, exception path and owner.

Automation should expose failures and uncertainty rather than converting them to default success. Record rule versions and input versions so results can be reproduced.

7. Select an implementation profile

Use the [Enterprise Data Product Profile](#) as the baseline for conventional enterprise systems and human-led operating models.

Use the [AI-Agent-First Data Product Profile](#) when AI agents directly discover, interpret, access, combine or act on data products.

The AI-Agent-First Profile extends the Core with stronger machine-readable context, workflow, control, provenance and assurance requirements. It does not create a separate framework.

An organisation can establish the Enterprise profile first and strengthen selected capabilities as it progresses toward agent-ready and agent-first operations.

8. Publication guidance

Markdown should remain the human-maintained source.

Generated HTML and PDF should:

- preserve headings and anchors
- include document metadata
- include framework version
- include publication date
- link back to source files
- avoid changing semantics during rendering

Publication should also expose the framework or artifact version and stable anchors so references remain meaningful.

Accessibility, responsive navigation and print quality are presentation requirements; they do not change the authority of the underlying source.

9. Evidence-informed implementation

The [Framework Evidence Library](#) contains supporting sources and their applicability boundaries. Use it to research an implementation decision, not to copy external requirements into the Core. Relevant starting points include [w3c-dcat-3](#), W3C DCAT 3, for catalog interoperability; [wilkinson-fair-principles-2016](#), the FAIR Guiding Principles, for machine-actionable metadata; [w3c-prov-o](#), W3C PROV-O, for provenance; and the cataloged NIST risk frameworks for proportionate governance and assurance.

For a material framework change, retain the research question, source identifiers, claims, limitations and architecture impact. LLM-assisted synthesis may propose text, but maintainers must verify source support and deliberately edit the normative Markdown.

10. Future playbook modules

Planned modules:

- starting from business demand
- converting use cases into product candidates
- defining ODPS contracts
- building demand-to-data graphs
- catalog federation
- agent-ready discovery
- product lifecycle workflows
- machine-readable governance
- service assurance
- value measurement
- portfolio review

TERMINOLOGY

Glossary

Function

A major area of the operating framework answering one fundamental management question.

Capability

An organisational ability required to operate data products effectively.

Practice

A repeatable activity performed to establish or operate a capability.

Activity

A specific action performed as part of a practice.

Object

An identifiable thing managed by the framework.

Examples include Business Objective, Use Case and Data Product.

Artifact

A durable representation created or used by a practice.

Examples include an ODPS file, ODPC catalog or approval record.

Evidence

Information demonstrating that an activity occurred, a requirement was satisfied or an outcome was observed.

Evidence should be distinguishable from the declaration, target or policy against which it is evaluated.

Evidence Provenance

Information explaining how evidence was produced, including the relevant source, activity or method, time, scope, version and responsible person or system.

Provenance improves traceability and reviewability. It does not by itself prove that the evidence is accurate or that the supported claim is valid.

Measure

A quantitative or qualitative method for evaluating performance or outcomes.

Control

A measure intended to modify risk or enforce a requirement.

Finding

A recorded result identifying a deviation, weakness, failure or other matter requiring attention.

Profile

An implementation adaptation of the Core that strengthens practices, evidence, machine-readable artifacts, controls or assessment expectations for a particular operating environment without redefining Core capabilities.

Core

The universal functions, capabilities, principles, terminology, traceability, evidence, assessment and conformance models shared by every framework implementation profile.

Crosswalk

A mapping between framework concepts and another framework, standard or regulatory system.

Business Objective

An outcome the organisation seeks to achieve.

Use Case

A problem, task, decision or process requiring data.

Consumer

A person, organisation, application, system or AI agent consuming a product.

The consumer identity should be specific enough to apply access conditions, understand context and evaluate adoption while respecting privacy and proportionality.

Accountable Owner

The person or formally delegated role answerable for a decision, product outcome or capability. Tasks may be delegated, but accountability and escalation authority remain explicit.

Product Decision

An explicit portfolio decision concerning creation, investment, change, consolidation or retirement.

Data Product

A reusable data capability offered to consumers.

Product Contract

The authoritative machine-readable definition of a data product.

The contract is distinct from a catalog record, delivery interface and runtime configuration, even when those surfaces present or execute parts of it.

Authoritative Artifact

The approved representation that has decision authority for a defined object, field or scope. Copies and generated views should resolve to or be reconciled with it.

Vocabulary Concept

A shared semantic concept with a stable meaning.

Relationship

A connection between framework objects.

Workflow or Recipe

A repeatable operational contract.

Execution

One occurrence of a workflow or operational process.

An execution should identify the applicable workflow, inputs, actor or system, outputs, decisions, status and evidence where material.

Outcome

An observed change associated with a use case or objective.

Value

The significance of an outcome to the organisation or its stakeholders.

Value is evaluated in relation to cost, alternatives, assumptions and attribution. Usage alone is not evidence of value.

Adoption

Meaningful use of a data product by an intended consumer for an approved or recognised use case. Discovery, access, trial, recurring consumption and production dependency are different stages and should not be treated as equivalent.

AI Agent

An AI-enabled software actor that can interpret context, select or invoke tools, execute steps or take actions with some degree of autonomy. Within this framework, an agent remains subject to product contracts, policy boundaries, identity, authorization, workflow and evidence requirements.

Agent-First Operations

An operating environment in which AI agents actively participate in discovery, product operations, workflows, assurance or decision support within explicit governance boundaries. It is an implementation stage and profile context, not a separate framework.

Machine-Readable

Represented in a structured form that software can parse and process according to explicit rules. A digital document containing only prose is not automatically machine-readable for framework purposes.

Conformance

A scoped conclusion that an artifact, practice or capability satisfies stated requirements. The object, version, profile, assessment method and evidence supporting the conclusion should be explicit.

Declared State

What the product contract, policy or workflow states should happen.

Observed State

What runtime evidence shows happened.

CONTEXT

Framework Profiles

Profiles adapt the [Data Product Operating Framework Core](#) to a specific operating environment.

The profile architecture is:

[Data Product Operating Framework Core](#) → [Enterprise Data Product Profile](#) → [AI-Agent-First Data Product Profile](#)

This is one framework. Profiles strengthen or specialise Core requirements; they do not create separate frameworks.

Available profiles

- [Enterprise Data Product Profile](#) — baseline implementation for conventional enterprise systems and human-led operating models
- [AI-Agent-First Data Product Profile](#) — stronger requirements for environments where AI agents directly discover, interpret, access, combine or act on data products

Profile model

A profile may:

- define its scope and intended environment
- extend Core requirements
- identify stronger practices
- identify required evidence
- identify mandatory machine-readable artifacts
- identify stronger control requirements
- define context-specific assessment expectations

A profile must not:

- duplicate the entire framework
- rename Core capabilities
- create a separate maturity model
- fork Core terminology
- redefine ODPS-family standards

All profiles use the common [Assessment Standard](#) and the same C1-C14 capability model.

Adoption progression

[Enterprise Data Products](#) → [Machine-Readable Data Products](#) → [Agent-Ready Data Products](#) → [Agent-First Operations](#)

Enterprise Data Products

Governed products exist with ownership, contracts, catalogs and lifecycle management.

Machine-Readable Data Products

Core product definitions, semantics, relationships and policies are increasingly structured and machine-readable.

Agent-Ready Data Products

Products contain enough explicit context, semantics, access information and governance information for safe machine interpretation.

Agent-First Operations

AI agents actively participate in discovery, product operations, workflows, assurance and decision support within explicit governance boundaries.

The stages describe an adoption path, not separate framework versions or maturity models. An organisation may adopt stronger practices selectively while retaining the same Core.

IMPLEMENTATION PROFILE

Enterprise Data Product Profile

1. Scope

The Enterprise Data Product Profile provides the baseline implementation profile for organisations managing enterprise data products through conventional enterprise systems and human-led operating models.

It applies the [Data Product Operating Framework Core](#) without redefining its functions, capabilities, terminology, assessment model or conformance model.

Typical environments include:

- data catalogs
- data platforms
- APIs
- analytics systems
- governance systems
- human approval processes
- product managers
- domain owners
- engineers
- business consumers

The profile provides a realistic adoption path for organisations that do not yet operate AI agents extensively. Automation and AI may be used, but neither is required for profile adoption.

2. Intended users

This profile is intended for:

- data product managers
- portfolio and investment decision-makers
- platform and catalog teams
- risk, compliance and assurance teams
- domain and data owners
- data governance teams
- data and software engineers
- business consumers

3. Assumptions

The profile assumes that:

- accountability remains primarily human-led
- catalogs and platforms may be federated or heterogeneous
- approvals may be manual or workflow-supported
- product information may be distributed across several enterprise systems

- machine-readable artifacts can be adopted incrementally
- the ODPS standards family supplies portable contracts and context where applicable
- existing governance and delivery systems remain operational authorities

The profile does not require replacement of existing platforms. It requires clear authority, portable definitions, traceability and evidence across them.

Distributed implementation is acceptable when authority is explicit. For example, the product contract may be maintained in source control, discovery metadata in a catalog, entitlements in an identity platform and observations in monitoring systems. The organisation should define which system is authoritative for each object, how identifiers connect them and how conflicts or stale copies are detected.

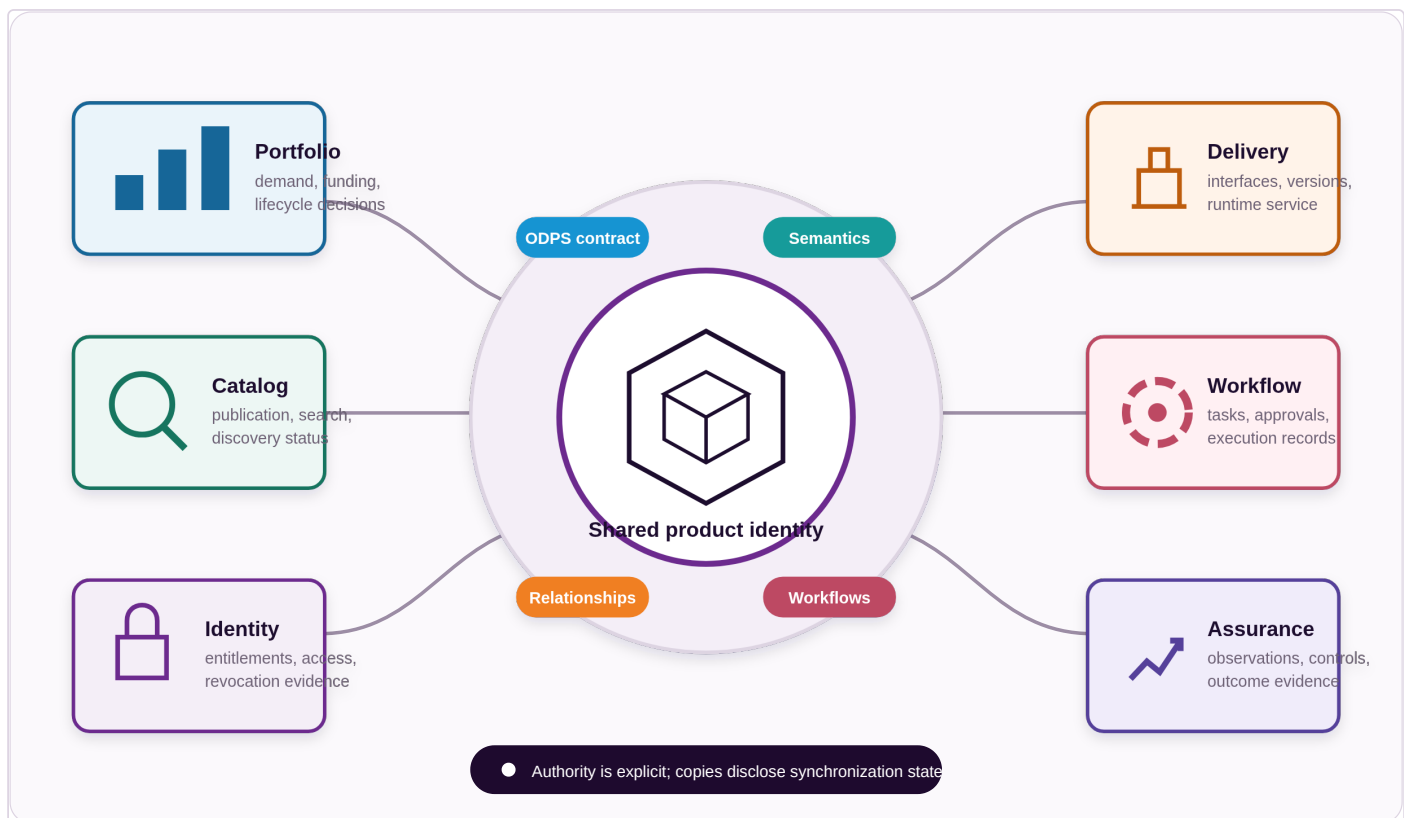


Figure 9. Enterprise adoption does not require one platform: existing systems can remain operational authorities when stable identifiers, ownership and evidence flows connect them.

4. Core capability requirements

All fourteen Core capabilities apply.

C1. Strategy and Objectives

Objectives should be explicit, measurable where practical and owned by accountable roles.

At minimum, retain the objective owner, intended outcome, measure or evaluation method, time horizon and approval status. A generic transformation theme is not sufficient justification for every product placed beneath it.

C2. Demand and Use Case Management

Use cases should be registered, owned and linked to relevant objectives and data products.

The record should identify the consumer, problem or decision, expected change, evidence of success and evaluation of existing products. Registration may begin in an established demand-management tool if stable references connect it to the portfolio and product contract.

C3. Portfolio, Investment and Accountability

Products should have accountable owners, explicit investment decisions and governed lifecycle decisions.

Portfolio review should consider continuing operating cost, active demand, overlap, dependencies, commitment performance and outcome evidence. Human committees may make the decisions, but their rationale and authority should be retained in a durable record.

C4. Product Definition and Contract

Active products should have authoritative ODPS definitions with stable identities, responsible providers, versions, interfaces and lifecycle states.

Existing catalog and platform metadata may supply inputs, but the approved ODPS artifact should be identifiable as the portable contract. Human-readable product pages should be generated from or reconciled with that artifact.

C5. Semantics and Vocabulary

Shared vocabulary is recommended for important cross-domain concepts. Material terms should have consistent definitions across product contracts, catalogs and governance records.

Organisations can start with the concepts that most often cause reconciliation, reporting or policy errors. Stable identifiers and defined ownership are more important than building a comprehensive enterprise ontology before products can operate.

C6. Product Commitments and Usage Conditions

Relevant quality, service-level, access, usage, policy, privacy, security, licensing and commercial conditions should be explicit.

Conditions should be expressed at the level needed for a consumer and approver to act. Commitments intended for assurance need a metric, scope, evaluation period and method; policy references need an applicability rationale and exception path.

C7. Relationships, Dependencies and Context

Critical product relationships and dependencies should be represented, including relationships to objectives, use cases, owners, upstream products, downstream consumers and applicable policies.

A spreadsheet or catalog relationship may be an acceptable starting point if identifiers and relationship types are controlled. Critical dependencies should support change-impact review and should not remain solely in architecture diagrams or individual knowledge.

C8. Catalog, Publication and Discovery

Products should be discoverable through an approved catalog or equivalent governed discovery service.

Discovery should expose proposition, owner, lifecycle state, current version, access path and important conditions in language consumers understand. Federated catalogs should retain authoritative references and synchronization status.

C9. Provisioning, Integration and Consumption

Access and provisioning should follow defined identity, entitlement, approval and revocation controls.

The implementation should record the consumer, intended use, product version, decision and resulting entitlement. Provisioning is complete only when usable connectivity and relevant conditions have been communicated or tested.

C10. Lifecycle, Version and Change Management

Versions, lifecycle states, material changes, deprecation and retirement should be managed and communicated to affected consumers.

Change classification should reflect consumer impact. Known consumers and dependencies should be used for impact analysis, migration and notification rather than relying only on general catalog announcements.

C11. Workflow, Automation and Agent Operations

Critical repeatable workflows should be documented, reviewable and increasingly automated where rules are clear. Human approval should remain where judgement, investment authority or risk acceptance is required.

The baseline can use existing ticketing, workflow or delivery tools when the workflow intent, inputs, decisions, outputs and evidence are explicit. ODPR adoption should improve portability and reuse rather than duplicate an already governed process without purpose.

C12. Observability and Service Assurance

Important service and quality commitments should be measured, compared with declared commitments and retained as evidence.

Measurements should identify the product and contract version, evaluation period, method and status. Missing observations should not be treated as successful performance.

C13. Governance, Risk, Compliance and Control Assurance

Applicable policies and controls should be linked to products, operated by accountable roles and supported by evidence.

The profile does not require every enterprise control to be copied into every product. Applicability, ownership, execution evidence, exceptions, findings and remediation should be traceable through the systems already used for governance and assurance.

C14. Adoption, Outcomes and Value Realisation

Products should be evaluated for discovery, access, adoption, supported use-case outcomes and organisational value rather than usage alone.

Start with a baseline and one outcome measure for each priority use case. Attribution may begin as a documented contribution assessment, provided assumptions, costs and uncertainty are visible to the portfolio decision-maker.

5. Mandatory evidence

At minimum, an in-scope operational product should have evidence of:

- an approved objective or registered demand
- an accountable product owner
- an authoritative, versioned ODPS product contract
- defined access and usage conditions
- an access, provisioning or entitlement decision
- measured quality or service performance where commitments apply
- adoption or meaningful consumption
- one or more registered use cases
- an explicit product decision and lifecycle state
- publication in an approved catalog or equivalent
- recorded critical relationships and dependencies
- a governed release or change decision
- control execution where policies or obligations apply
- an outcome or value review

Evidence may remain in existing enterprise systems when it is identifiable, retained and traceable to the product.

6. Recommended automation

Organisations should progressively automate:

- schema and contract validation
- broken-reference and dependency checks
- entitlement verification
- quality measurement
- control evidence collection
- evidence packaging for assessment
- catalog publication
- access request routing
- release and change notifications
- service-level evaluation
- lifecycle review reminders

Automation should not replace accountable decisions merely to increase throughput.

7. ODPS-family expectations

The profile applies the common authority model:

- **ODPS** represents the individual data product contract.
- **ODPC** represents portfolio and discovery objects.
- **ODPV** represents shared vocabulary and semantics.
- **ODPG** represents relationships and context.
- **ODPR** represents reusable workflow contracts.
- **Operational systems** execute work and provide runtime observations.
- **The evidence layer** preserves what proves what happened.

ODPS definitions are expected for active governed products. The other ODPS-family standards should be adopted where shared portfolio objects, vocabulary, relationship graphs or portable workflows create practical value.

The framework and this profile do not redefine ODPS-family standards.

8. Minimum implementation baseline

An organisation meets the minimum implementation baseline when it can demonstrate one complete governed traceability chain containing:

1. an owned objective
2. a registered use case
3. an approved product decision
4. an accountable data product owner
5. a valid, versioned ODPS product contract
6. an approved catalog entry or equivalent discovery record
7. a controlled access path
8. a managed lifecycle state
9. one measured quality or service commitment
10. applicable control evidence
11. an adoption measure

12. an outcome or value review

The chain should be established for a small number of products before it is scaled across the portfolio.

9. Assessment expectations

Assessment uses the common [Assessment Standard](#), the same C1-C14 capability model and the same four dimensions:

	Accountability
	Practice
	Evidence
	Outcome

Profile requirements add context-specific expectations to those capabilities. They do not create a separate maturity model or allow artifact conformance to substitute for operating evidence.

Assessment should verify that human-led decisions are explicit, machine-readable artifacts are increasing, and operational evidence can be traced back to the relevant product, use case and objective.

10. Evidence basis and applicability

This profile translates the Core into a conventional enterprise environment; it does not require conformance to the external sources below. Library sources [w3c-dcat-3](#) and [wilkinson-fair-principles-2016](#)—W3C DCAT 3 and the FAIR Guiding Principles—support interoperable, machine-actionable discovery. Sources [nist-csf-2.0](#), [nist-privacy-framework-1.0](#) and [nist-sp-800-53r5](#)—NIST Cybersecurity Framework 2.0, NIST Privacy Framework 1.0 and NIST SP 800-53 Rev. 5—provide patterns for proportionate risk, access, change, control and assurance practices.

Organisations should apply legal, regulatory, contractual and sector-specific requirements through C6 and C13. A requirement that applies in one jurisdiction or product class should not be presented as a universal Enterprise Profile requirement without an explicit profile revision.

IMPLEMENTATION PROFILE

AI-Agent-First Data Product Profile

1. Scope

The AI-Agent-First Data Product Profile defines stronger requirements for data product environments where AI agents directly:

- discover data products
- interpret product context
- select products
- access products
- combine products
- invoke tools
- execute workflows
- reason over relationships
- evaluate evidence
- take actions based on data products

The AI-Agent-First Profile extends the [Data Product Operating Framework Core](#). It does not create a separate framework.

It strengthens selected Core capabilities. It does not rename them, fork Core terminology, create a separate maturity model or redefine ODPS-family standards.

2. Intended users

This profile is intended for organisations operating or preparing to operate:

- AI agents that consume or act on data products
- agent-assisted product operations
- tool-using workflows
- agent-oriented discovery services
- automated evidence evaluation
- governed decision-support or action-taking systems

It is also relevant to product, platform, governance, security, risk, compliance and assurance teams responsible for those environments.

3. Assumptions

The profile assumes that:

- the Core is already adopted as the common operating model
- agents have identifiable operating contexts and bounded authority
- agent actions can be observed and retained as evidence
- enterprise ownership and lifecycle controls remain in force
- authoritative artifacts can be resolved programmatically
- human accountability remains explicit

- higher autonomy requires stronger context, controls and assurance

Agent-first does not mean agent-only. Human review, escalation and decision rights remain part of the operating model where material judgement or authority is required.

Proportional strengthening

The strength of implementation should reflect at least:

- autonomy — whether the agent recommends, prepares, decides or acts
- materiality — the consequence for people, services, finances, rights or obligations
- reversibility — whether an action can be safely undone
- reach — the number and variety of products, consumers and systems affected
- uncertainty — the reliability of context, semantics, models and observations
- exposure — the sensitivity of data and power of available tools

Higher consequence, autonomy or irreversibility requires stronger identity, authorization, validation, monitoring, approval, suspension and evidence. Low-risk assistance may use lighter controls, but it should still remain traceable to an approved use case and accountable owner.

4. Strengthened Core capability requirements

All fourteen Core capabilities apply. The profile adds the following context-specific expectations.

C1. Strategy and Objectives

Agent participation should be linked to an explicit objective, approved use case and expected outcome. Autonomy should not be introduced without a justified operating need.

C2. Demand and Use Case Management

Agent-supported use cases should identify the consumer, intended decision or action, acceptable outcome, evidence needs and material failure conditions.

C3. Portfolio, Investment and Accountability

Agent-enabled products and workflows should have accountable owners, explicit investment decisions and lifecycle decisions covering continuation, suspension and retirement.

C4. Product Definition and Contract

Require:

- an authoritative machine-readable ODPS contract
- stable identifiers
- explicit versions
- machine-resolvable interfaces
- clear consumer context
- agent-usable product descriptions

The contract should allow an agent to determine what the product is, what it provides, how it can be accessed and which version is applicable without relying only on free-text pages.

C5. Semantics and Vocabulary

Require:

- machine-readable semantic definitions
- controlled vocabulary for material concepts
- stable identifiers
- explicit aliases
- multilingual labels where relevant
- mappings to external vocabularies where useful
- semantic consistency across artifacts

Ambiguous or conflicting meanings should be treated as operating risks, not only documentation defects.

C6. Product Commitments and Usage Conditions

Require machine-readable:

- quality expectations
- service-level expectations
- access conditions
- usage restrictions
- security conditions
- privacy conditions
- licensing where relevant
- commercial terms where relevant

Executable or automatically testable rules are preferred where practical. Machine-readable conditions do not remove the need for accountable policy ownership or exception decisions.

C7. Relationships, Dependencies and Context

Require graph-traversable relationships for relevant:

- objectives
- use cases
- products
- dependencies
- KPIs
- policies
- controls
- APIs
- agents
- workflows

Relationship provenance and confidence should be retained where relevant. Agents should be able to distinguish authoritative, inferred and unverified relationships.

C8. Catalog, Publication and Discovery

Require agent-oriented discovery through structured metadata and machine interfaces.

AI agents should be able to identify suitable products, resolve authoritative references and evaluate basic fitness for use without depending only on free-text catalog pages.

Discovery results should respect visibility, policy and entitlement boundaries.

C9. Provisioning, Integration and Consumption

Require machine-readable access and integration instructions.

Where required, agents should have identifiable identities and explicit authorization. Agent access follows the same product contract, policy boundaries, entitlement controls and revocation processes as other consumers.

Credentials should not be inferred from the identity of the human or system that initiated the workflow unless explicitly authorised.

C10. Lifecycle, Version and Change Management

Require agents to identify:

- the current version
- supported versions
- deprecated versions

- breaking changes
- migration expectations

Agent consumers should not silently continue using invalid assumptions after product changes. Cached context and workflow dependencies should be invalidated or reviewed when material contracts change.

C11. Workflow, Automation and Agent Operations

This is the strongest capability in this profile.

Agent-assisted workflows should define where appropriate:

- workflow intent
- authoritative input artifacts
- grounding boundaries
- available tools
- permissions
- allowed actions
- iteration limits
- stopping conditions
- expected outputs
- validation gates
- human approval gates
- escalation conditions
- workflow version
- provenance requirements

ODPR should be the preferred portable workflow representation. Runtime-specific orchestration may execute the workflow but should not silently redefine its declared boundaries.

C12. Observability and Service Assurance

Require automated comparison where practical between declared state and observed state.

Examples include:

- declared quality versus measured quality
- declared service level versus runtime performance
- declared version versus deployed version
- declared access rules versus observed entitlements

The comparison result, measurement method, product version and relevant agent or workflow identity should be retained as evidence.

C13. Governance, Risk, Compliance and Control Assurance

Extend control assurance to include:

- agent permissions
- agent identity
- tool access
- action boundaries
- human escalation
- evidence provenance
- workflow provenance
- model provenance where relevant
- policy enforcement
- kill or suspension mechanisms where material

Controls should address both prohibited actions and failure to stop, escalate or request approval when required.

C14. Adoption, Outcomes and Value Realisation

Do not measure agent success through:

- tool calls
- prompts
- token volume
- workflow execution count alone

Measure:

- successful use cases
- consumer outcomes
- decision improvements
- operational outcomes
- financial value where appropriate
- risk reduction
- service improvements

Agent activity is execution evidence. It is not automatically evidence of value.

5. Mandatory evidence

In addition to applicable Core evidence, an in-scope agent-first implementation should retain:

- the authoritative product and workflow versions used
- the agent identity and relevant authorization context

- resolved product, vocabulary and relationship references
- workflow inputs and expected outputs
- execution and decision provenance
- policy and control outcomes
- observed product and service results
- tool and action permissions
- validation and approval-gate results
- material model provenance where relevant
- exceptions and escalations
- the supported use-case outcome

Evidence retention should be proportionate to the materiality, autonomy and reversibility of the agent's actions.

6. Recommended automation

Automate where practical:

- ODPS, ODPC, ODPV, ODPG and ODPR validation
- semantic consistency checks
- breaking-change detection
- declared-versus-observed comparison
- control evidence packaging
- identifier and reference resolution
- policy and entitlement evaluation
- workflow boundary validation
- provenance capture
- suspension of workflows that exceed defined boundaries

Automation should fail safely when authoritative context, permission or required evidence cannot be resolved.

7. ODPS-family expectations

The profile expects stronger use of the common authority model:

- **ODPS** is the authoritative individual data product contract.
- **ODPC** supplies machine-oriented portfolio and discovery objects.
- **ODPV** supplies shared vocabulary and semantics.
- **ODPG** supplies graph-traversable relationships and context.
- **ODPR** supplies reusable workflow contracts.
- **Operational systems** execute work and provide runtime observations.
- **The evidence layer** preserves what proves what happened.

These artifacts should use stable identifiers and resolvable references. The profile strengthens how the standards are used; it does not add a new ODPS-family specification or redefine an existing one.

8. Minimum implementation baseline

An organisation should demonstrate at least one bounded agent-supported use case containing:

1. an approved objective and use case
2. an accountable human owner
3. a machine-readable ODPS contract

4. structured discovery metadata
5. controlled vocabulary for material concepts
6. graph-traversable product and governance context
7. machine-readable access instructions
8. an identifiable agent and explicit authorization
9. a versioned workflow contract with stopping and approval conditions
10. recorded execution and provenance evidence
11. declared-versus-observed assurance
12. a measured use-case outcome
13. a tested suspension or escalation path where material

The baseline should be proven on bounded, reviewable workflows before autonomy is expanded.

9. Assessment expectations

Assessment uses the common [Assessment Standard](#), C1-C14 capability model and dimensions of Accountability, Practice, Evidence and Outcome.

The profile does not award maturity merely because an organisation uses agents or automates workflows. Assessors should verify that:

-
-  accountability and decision rights remain explicit

 -  required machine-readable artifacts are authoritative and resolvable

 -  agent permissions and action boundaries operate in practice

 -  evidence and provenance are retained

 -  declared state is compared with observed state

 -  outcomes and value are measured beyond execution volume

Stronger profile requirements increase the evidence expected within a capability. They do not create a separate capability score or maturity model.

10. Evidence basis and applicability

Library sources [nist-ai-rmf-1.0](#), [nist-ai-600-1-genai-profile](#) and [oecd-ai-principles-2024](#) support this profile. The NIST AI Risk Management Framework 1.0 supports lifecycle governance, contextual risk analysis, measurement and accountable management. Its [Generative AI Profile](#) adds generative-AI risk and action patterns relevant to some agent implementations. The [OECD AI Principles](#) provide intergovernmental context for beneficial outcomes, transparency, robustness and accountability.

Sources [w3c-prov-o](#), [w3c-odrl-2.2](#) and [w3c-shacl](#) provide implementation patterns. [W3C PROV-O](#) informs execution and evidence provenance; [W3C ODRL 2.2](#) informs machine-readable permissions and duties; and [W3C SHACL](#) provides an example of machine-testable constraints and validation results. These are implementation resources, not replacements for

ODPS-family authority.

Library source [eu-ai-act-2024-1689](#), Regulation (EU) 2024/1689, may impose legal requirements for particular AI systems, actors, uses and dates in its jurisdiction. This profile does not make those jurisdiction-specific obligations universal or provide a legal compliance determination. Organisations should map applicable duties to C6 and C13 with qualified legal and risk ownership.

ALIGNMENT

Crosswalks

Crosswalks map the Data Product Operating Framework to relevant external standards and frameworks.

Candidate crosswalks include:

- DCAM
- DAMA-DMBOK
- COBIT
- NIST AI RMF
- ISO/IEC 42001
- ISO 37301
- ISO 8000 family
- ISO/IEC 27001
- TOGAF

A crosswalk should identify:

- source framework element
- mapped DPOF function
- mapped DPOF capability
- relationship type
- notes
- evidence implications