FIELD NOTE · OPERATING MODELS · 03

What Is a Digital Product?

The most-used, least-defined phrase in enterprise technology — taken apart into five parts you can fund, own, and retire.

THE ANATOMY, AND WHY THE PRECISION PAYS OFF

- **Service Offer — the menu a consumer chooses from**
- **Contract — what accepting an offer creates**
- **Instance — one per consumer**
- **System — the code that delivers the outcome**
- **Other Digital Products — the real-time supply chain**

Tom Ward
Enterprise Architect · Cloud & Infrastructure

linkedin.com/in/tomward
July 2026

Ask ten leaders what a "digital product" is and you'll get ten answers — Gartner has written that executives agonize over the definition. That isn't a semantic nuisance; it's a management problem. You cannot fund, own, or retire a thing you cannot define, and a vague definition produces vague accountability and a vague budget. The Open Group's lifecycle papers propose a rigorous definition and, more usefully, an anatomy: the same five parts sit inside every digital product, from a consumer mobile app to an internal HR platform.

> **THE DEFINITION**
>
> **A digital product is anything that delivers an agreed outcome for a consumer, requires software to realize it, is actively managed across its whole lifecycle, and comes with a formal offer of value at an explicit price.**

PART 1 Service Offer

"The menu"

The offer describes the contract terms available to a potential consumer — tiers, SLAs, feature sets, consumption rules, and an explicit price — filtered and presented per consumer type. It is derived from the product definition and documented by the product manager, and a single product can give rise to thousands of possible offers. The offer is a full statement of value, not a price sheet.

WHAT IT INCLUDES

- Pricing tiers, SLA templates, feature sets, and consumption rules
- Variants per consumer type — human and machine
- The terms that will become the contract if accepted

WHERE IT BREAKS

- × Treating the offer as a price list rather than a value statement
- × One offer for every consumer, ignoring machine consumers
- × Offers that promise what the system can't actually deliver

PART 2 Contract

"The agreement"

When a consumer accepts an offer, a contract captures the provider-consumer relationship and everything needed to run it: financial recording, warranties and SLAs, chargeback rules, usage monitors, audit rights, and maintenance obligations. Critically, the contract exists *separately* from the product's technical components and may outlive every user — residual retention and compliance obligations remain after the product is gone.

WHAT IT INCLUDES

- Price, SLAs, warranties, and the outcomes both sides commit to
- Chargeback basis, usage rules, and monitoring terms
- Audit rights and long-term ownership/maintenance obligations

WHERE IT BREAKS

- × "It's internal" — so no contract for internal or machine consumers
- × Obligations forgotten once the product retires
- × SLAs with no monitor behind them to prove conformance

PART 3 Instance

"One per consumer"

An instance is a single realization of the product, created when a consumer accepts an offer. It is the consumer's own slice — device, seat, login, entitlements, access controls, and any physical goods specific to that relationship. The instance is not the platform; it's what one consumer actually holds.

WHAT IT INCLUDES

- The consumer's device, seat, login, and entitlement
- Access controls and resources allocated to that consumer
- A traceable link back to the governing contract

WHERE IT BREAKS

- × Conflating the instance with the shared system behind it
- × Instances that can't be traced to a contract for chargeback or audit
- × Entitlements that drift out of sync with what was sold

PART 4 System

"What runs it"

The system is the technology the provider manages to deliver the outcome — code, compute, data, integrations, and third-party components, from a few lines on cloud infrastructure to a large web of interrelated parts. Each resource has its own support lifecycle. Unlike a physical product, the supply chain is real-time: daily updates demand automated toolchains.

WHAT IT INCLUDES

- Code, compute, data, and integrations that deliver the outcome
- Third-party components, each with its own support lifecycle
- The design, source, and configuration needed to sustain it

WHERE IT BREAKS

- × Obsolete subcomponents left unpatched — vulnerabilities, lost support
- × Design and source lost when the project team disbands
- × No automation for the daily/weekly change the product actually needs

PART 5 Other Digital Products

"Products all the way down"

A system routinely holds contracts to consume *other* digital products — internal microservices and external APIs like single sign-on, credit checks, or currency lookup — each a technical *and* financial dependency. These are part of the product's own definition and a factor in its price. The discipline: bind every critical dependency by a formal contract with real governance.

WHAT IT INCLUDES

- Subscriptions to internal and external digital products
- External APIs (SSO, credit, FX) under supplier contracts
- Dependency costs rolled into the product's own TCO

WHERE IT BREAKS

- × Untracked upstream dependencies — a change breaks you silently
- × External deps with no contract or API governance
- × Dependency cost invisible in the product's price

From Definition to Operating Model

The anatomy isn't an academic exercise. A product you can define this cleanly is a product you can actually run — and that is the whole argument of The Open Group's product-centric operating model (D-PROM). Precision at the definition is what unlocks four operating moves that a project-and-silo model can't make.

🎯 One accountable owner

A single Digital Product Line Manager owns the product's value and risk across its whole lifecycle — a role that fuses an IT-management competency with a product-and-marketing one, and carries a general manager's mindset rather than a project manager's. Accountability must come with matching authority; "accountability without authority" is the failure mode.

💧 Continuous funding

You fund the product, not a project. It keeps its budget until leadership deliberately re-prioritizes or retires it — visualized through sliding windows rather than annual approvals. The question shifts from "did this project get approved?" to "which value-delivering products should we invest in?"

∞ Whole-life value

Value is managed from strategy through retirement, not just at build. Two lifecycles run in parallel — the system lifecycle and the contract lifecycle — mapping to the IT4IT value streams (Strategy to Portfolio, Requirement to Deploy, Request to Fulfill, Detect to Correct).

🏢 Run like a micro-enterprise

Each product line operates as a small, semi-autonomous business accountable for its own outcomes — with decentralized decision-making matched by the authority to act. Consumers may be internal or external, human or machine; the definition holds across all four.

Why the definition is the unlock

None of the four is possible while "product" means whatever the last person in the room decided it meant. Name the five parts, and the owner, the budget, and the lifecycle all have something concrete to attach to. **The definition is not documentation — it is the precondition for management.**

The Digital Product Definition Test

Pick your most important digital product and answer honestly. Score one point per "yes." Below seven, "product" is still a word in your organization, not a managed thing — and that is a management gap, not a documentation one.

- ☐ You can name the product's **five parts** — Offer, Contract, Instance, System, and the other products it depends on.
- ☐ A **single accountable owner** holds its value and risk across the whole lifecycle — a named person, not a committee.
- ☐ The **Service Offer is a full value statement** — tiers, SLAs, terms — not a price sheet, and it's defined per consumer type.
- ☐ Every consumer relationship — **including internal and machine-to-machine** — is governed by a contract, explicit or implied.
- ☐ You can **trace any instance back to its contract** for chargeback, audit, and support.
- ☐ The system's dependencies each have a **known owner and support lifecycle**, and none is quietly obsolete.
- ☐ Every **external digital product it depends on is under a formal contract** with defined governance — API governance where external.
- ☐ Its **price is a deliberate decision** — direct or indirect — and dependency costs are rolled into its TCO.
- ☐ **Design, source, and configuration are retained** for long-term support — not lost when a project team disbands.
- ☐ **Residual contract obligations** — retention, compliance — are owned even after the product retires.

Going Deeper: References

The sources behind this note, roughly in reading order. All links verified July 2026.

The Shift to Digital Product: A Full Lifecycle Perspective (W205)
The Open Group white paper this note's definition and five-part anatomy are drawn from. Start here.
publications.opengroup.org/w205

A Digital Product-Centric Reference Operating Model (D-PROM, W223)
The operating model built on the definition — funding, accountability, and org design.
publications.opengroup.org/w223

The IT4IT™ Standard 3.0
The reference architecture underneath the lifecycle — the value streams the product model maps to.
publications.opengroup.org/c221

Project to Product — Mik Kersten
The book that named the movement, and a pivotal influence on the Open Group's thinking here.
projecttoproduct.org

MIT CISR — Designed for Digital (Ross, Beath, Mocker)
The research on digital operating models and product-oriented organizations, cited in W205.
cisr.mit.edu

The Open Group Library
Where these papers and their companion standards (DPBoK, IT4IT) are published.
opengroup.org/library

About the Author

Tom Ward is an Enterprise Architect with 20+ years across Fortune 500 financial services, retail, and government — including Wells Fargo and Bank of America — specializing in EA governance, reference architectures, hybrid cloud, and legacy modernization, with recent hands-on GenAI and mobile delivery work.

More field notes: archreviews.com/field-notes · Connect: linkedin.com/in/tomward