

THE COMMERCE KNOWLEDGE ARCHITECTURE

CKA

LogicShop as the Reference Implementation

**The problem is not content.
The problem is architecture.**

JOE PIZARRO

A concise, complete guide to governing product knowledge for people, systems, and AI.

QUICK ORIENTATION

The architecture in one sentence

The Commerce Knowledge Architecture creates one governed representation of a product, separates that knowledge from presentation, and lets every channel, team, system, and AI experience consume the portion it needs.

One product

A product exists once as governed knowledge rather than being recreated for each destination.

Many expressions

Websites, marketplaces, service tools, print, retail, wholesale, and AI present different views of the same source.

Stable architecture

Platforms and providers can change without forcing the organization to rebuild product meaning.

CKA is not a website, PDP, PIM, CMS, DAM, marketplace tool, compliance platform, or AI assistant. Those are systems and experiences that participate in the architecture. CKA defines how product knowledge is structured, governed, connected, and delivered among them.

How to read this edition: Part I explains the problem. Part II defines the architecture. Part III describes governance and system boundaries. Part IV provides an implementation path. Part V explains why the model matters now.

PART I Why commerce needs a knowledge architecture

01 / THE PROBLEM

Organizations repeatedly recreate the same product

Most commerce organizations do not have one durable definition of a product. Marketing maintains descriptions. Merchandising changes titles and attributes. Compliance revises claims. Marketplace teams optimize listings. Retail, sales, wholesale, and service teams create their own materials. Technical documents and media live elsewhere. Each team is solving a legitimate need, but the organization gradually accumulates competing versions of the same product.

The visible symptom is “too much content work.” The architectural problem is that product information is treated as an output of each channel rather than a shared business asset. Every new destination therefore creates another maintenance obligation.

Duplicate effort

The same fact is written, reviewed, translated, and corrected repeatedly.

Inconsistency

Customers, employees, platforms, and AI systems receive different answers.

Operational risk

Outdated claims, specifications, and policies survive in disconnected systems.

Slow change

Launches and corrections depend on manual updates across many destinations.

Poor machine understanding

Search and AI reconstruct meaning from prose that was never designed for retrieval.

Fragile experiences

Changing a platform or provider forces expensive redesign and reauthoring.

Content is written repeatedly. Knowledge is created once, governed carefully, and reused wherever it provides value.

02 / THE OBJECTIVE

Move product meaning into a governed knowledge layer

CKA introduces a stable layer between enterprise source systems and downstream experiences. That layer is not a replacement for ERP, PIM, DAM, commerce platforms, or specialized providers. It coordinates them around a canonical model and clear ownership boundaries.

The result is a product that can outlast any page template, marketplace, interface, technology provider, or generation of AI. The same architecture supports consumer sites, marketplaces, wholesale, retail associates, customer service, printed materials, search, assistants, and future channels that do not yet exist.

EXPERIENCE	HOW THE SAME KNOWLEDGE IS EXPRESSED
Consumer website / PKE	A complete educational commerce experience with purchasing tools.
Marketplace listing	Channel-compliant attributes, titles, claims, media, and structured metadata.
Wholesale portal	Commercial specifications, packaging, minimums, documents, and assets.
Retail associate application	Concise education, comparisons, recommendations, and talking points.
Customer service	Approved troubleshooting, policies, ingredient explanations, and FAQs.
Print and trade show	Generated sell sheets, technical literature, and QR-led resource access.
AI assistant or agent	Grounded retrieval for explanation, comparison, recommendation, and action.

Design goals

Authoritative: one governed source for product meaning.

Structured first: knowledge exists as attributes, entities, relationships, and evidence before it becomes prose.

Modular: each capability has one purpose and a stable interface.

Portable: the architecture belongs to the organization rather than a platform.

Governed: ownership, approvals, history, and downstream impact are visible.

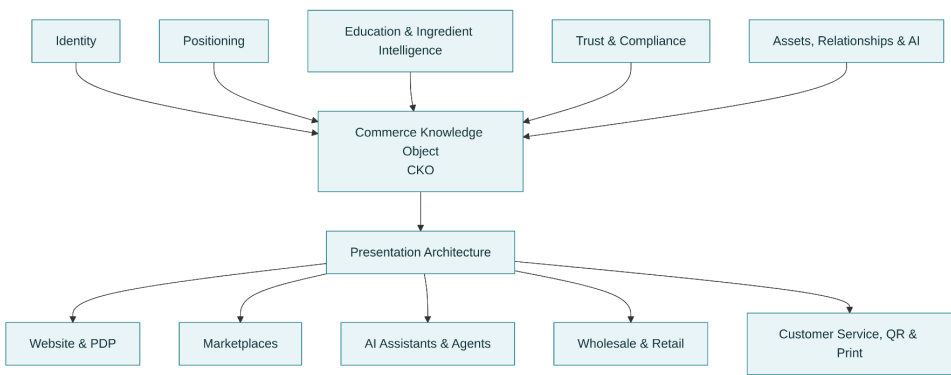
Human and machine readable: the same source can serve people, search, software, and AI.

PART II The architecture

03 / ARCHITECTURE AT A GLANCE

Knowledge is the stable center; experiences are replaceable views

The architecture has four cooperating elements. The **Commerce Knowledge Object** represents the product. **Governance** controls how that object changes. **Presentation Architecture** determines how audiences receive knowledge. **Integrations** connect source, execution, and intelligence systems without duplicating their responsibilities.



One governed Commerce Knowledge Object connects product knowledge domains to presentation architecture and many downstream experiences.

1. Model

What knowledge exists?

2. Governance

Who may change it, and how?

3. Presentation

What does this audience need now?

4. Integration

Which system owns each capability?

04 / OPERATING PRINCIPLES

Ten principles reduce to five durable rules

The principles are deliberately technology-independent. They are decision tests: every new platform, provider, module, channel, and AI capability should strengthen these rules rather than create another isolated product version.

OPERATING RULE	WHAT IT MEANS	ORIGINAL PRINCIPLES
Govern one source	Maintain one authoritative product record. Changes happen once and propagate everywhere.	1. Single source of truth 8. Governed information
Separate knowledge from display	Store facts, meaning, and relationships independently from the page, channel, or interface that presents them.	2. Separate knowledge from presentation 3. Structured by default
Build modularly	Give each capability one purpose and a stable interface so providers can change without redesigning the system.	4. Modular by design 10. Stable interfaces
Design for portability	The architecture belongs to the organization, not a commerce platform or vendor.	5. Portable by design 9. Integration over replacement
Serve people and machines	Reveal detail progressively while keeping the underlying knowledge structured for search, AI, agents, and automation.	6. Progressive disclosure 7. Human and machine readable

Progressive disclosure is a presentation rule, not a reason to omit knowledge. The complete record remains available; each experience reveals detail according to audience, context, and intent.

05 / COMMERCE KNOWLEDGE OBJECT

The product becomes a governed object, not a collection of pages

The Commerce Knowledge Object (CKO) is the canonical model at the center of CKA. It combines product identity, meaning, evidence, commerce, assets, AI metadata, integrations, and relationships while allowing each domain to have its own owner and rate of change.

A downstream experience does not copy the whole object. It retrieves the domains and fields required for its purpose. This makes the CKO broader than a catalog record: it represents the ecosystem of knowledge surrounding a product.

KNOWLEDGE DOMAIN	QUESTION ANSWERED	TYPICAL CONTENTS
Identity	<i>What is this product?</i>	Canonical name, identifiers, brand, category, form, model, variant.
Positioning	<i>Why does it exist, and for whom?</i>	Audience, need, differentiators, benefits, value proposition.
Product attributes	<i>What can be measured or compared?</i>	Dimensions, materials, features, specifications, taxonomy, filters.
Ingredient intelligence	<i>What is it made from, and what does each component mean?</i>	Ingredients, functions, sources, relationships, educational references.
Trust and compliance	<i>Why should it be trusted?</i>	Claims, certifications, testing, regulatory evidence, warnings, provenance.
Usage	<i>How should it be used?</i>	Instructions, routines, compatibility, troubleshooting, care.
Commerce	<i>How can it be bought?</i>	Price, availability, packs, subscriptions, promotions, channel constraints.
Assets	<i>What supports the product?</i>	Images, video, documents, packaging, certificates, downloads.
AI	<i>What helps machines interpret it correctly?</i>	Synonyms, embeddings, retrieval cues, approved answers, guardrails.
Integration	<i>Which external capability is connected?</i>	Reviews, loyalty, subscriptions, laboratories, external services.
Relationships	<i>How does it connect to other products and concepts?</i>	Alternatives, complements, bundles, families, ingredients, goals, use cases.

Products should not be recreated every time the world creates a new channel. They should exist as knowledge that can be governed once and expressed wherever it is needed.

06 / PRESENTATION ARCHITECTURE

Different layouts. Same knowledge.

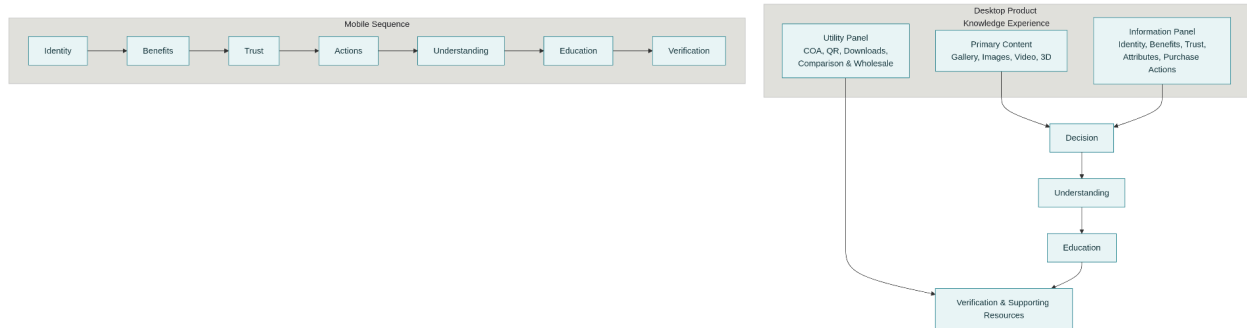
Presentation Architecture defines how knowledge is selected, sequenced, and rendered. The interface never becomes the source of truth. A redesign, new marketplace, mobile application, kiosk, voice interface, or AI agent can change presentation without restructuring the product.

The Product Knowledge Experience

The Product Knowledge Experience (PKE) is the primary reference implementation. It replaces the assumption that a product page is a long promotional document with a modular experience organized around decision-making.

Decision	Understanding	Education	Verification
-----------------	----------------------	------------------	---------------------

Identity, core benefits, trust signals, price, availability, and primary actions.	Attributes, comparisons, positioning, use cases, and practical fit.	Ingredients, mechanisms, guides, demonstrations, and deeper learning.	Testing, certificates, documentation, policies, and supporting evidence.
---	---	---	--



Desktop organizes primary content, decision knowledge, and utilities in parallel; mobile preserves the hierarchy as a progressive sequence.

Module contract

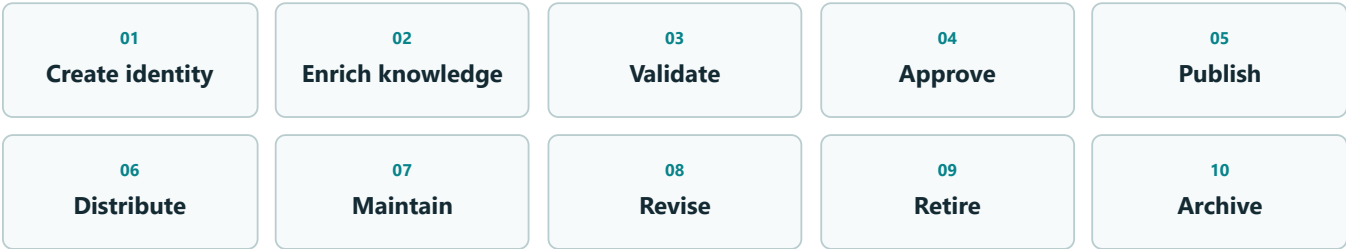
Every module defines a purpose, inputs, outputs, visibility rules, dependencies, analytics, and a stable location in the experience. Empty modules disappear without collapsing the architecture. Reserved slots allow subscriptions, reviews, comparison, certificates, AI, loyalty, and future capabilities to change providers without redesigning the surrounding experience.

The experience should feel organized rather than crowded: educational rather than promotional, trustworthy rather than persuasive, and flexible rather than platform-dependent.

PART III Governance and system boundaries

Knowledge becomes trustworthy through a governed lifecycle

A CKO is not “finished” when a product launches. It moves through a lifecycle in which different owners enrich, validate, approve, distribute, maintain, revise, retire, and preserve knowledge.



Ownership

Each domain has an accountable owner. Marketing may own positioning; product teams own identity and attributes; compliance owns approved claims and evidence; commerce owns price and availability; asset teams own media; governance coordinates changes that affect multiple domains.

Controls

Version history, approval workflows, validation, change rationale, audit trails, effective dates, and rollback make knowledge recoverable. Syndication rules determine which approved fields each destination may receive.

Governance is a competitive advantage

Governance is not administrative friction when it is designed well. It reduces repeated review, accelerates safe change, improves confidence, and allows teams to reuse knowledge without renegotiating its meaning every time.

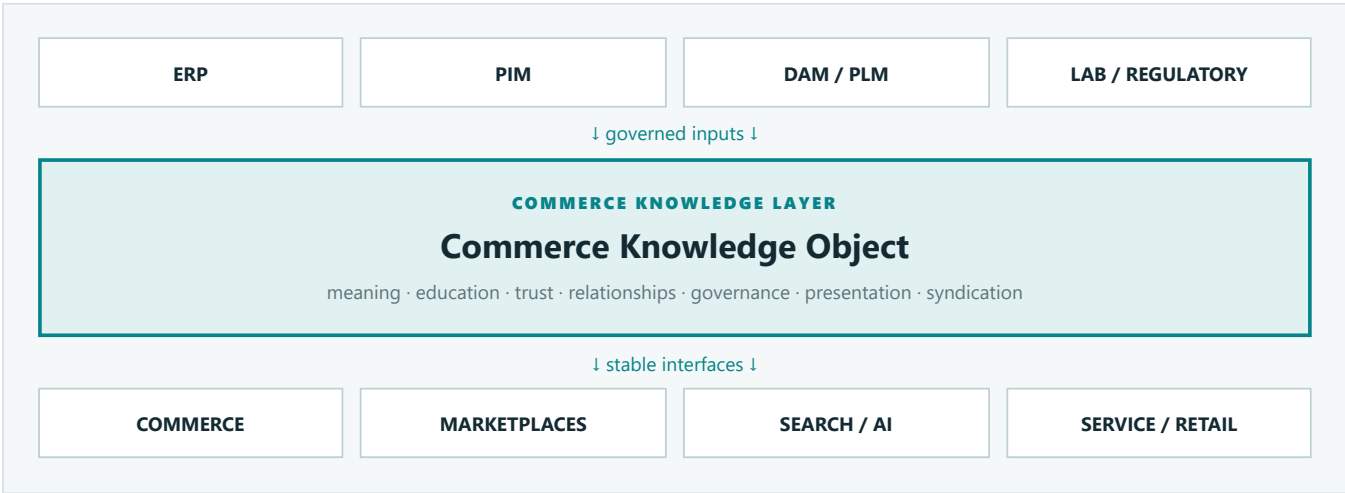
08 / REFERENCE ARCHITECTURE

Clarify ownership instead of replacing every platform

CKA works when every connected system has a clear responsibility. ERP should remain authoritative for operational and financial facts. PIM should remain authoritative for core attributes and taxonomy. DAM should remain authoritative for assets. Commerce systems should continue to execute transactions. The Commerce Knowledge Layer supplies the governed meaning that connects them.

SYSTEM CLASS	PRIMARY RESPONSIBILITY	EXAMPLES
Systems of record	Create and maintain authoritative operational data.	ERP, PIM, DAM, PLM, laboratory, regulatory systems.
Commerce Knowledge Layer	Organize meaning, education, trust, relationships, governance, presentation definitions, and syndication rules.	Commerce Knowledge Objects and their governed domains.
Execution systems	Sell products and manage transactions or customer sessions.	Shopify, Adobe Commerce, marketplaces, POS, wholesale portals.

SYSTEM CLASS	PRIMARY RESPONSIBILITY	EXAMPLES
Intelligence systems	Analyze, retrieve, recommend, automate, and reason.	Search, analytics, personalization, LLMs, agents, decision engines.



Integration patterns

Organizations may use APIs, events, scheduled feeds, batch processes, or hybrid patterns. The important choice is not transport; it is authority. Every domain needs one source of truth, explicit transformation rules, and a visible path from change to downstream effect.

PART IV Implementation

09 / ROADMAP

Adopt the architecture incrementally

CKA is an operating model, not a big-bang replacement program. Start with a representative product set, create value in one experience, and expand product by product, team by team, and channel by channel.

PHASE	OBJECTIVE	PRACTICAL OUTPUT
1. Discovery and assessment	Understand the current state.	System inventory, ownership map, duplication and quality assessment.
2. Commerce Knowledge Model	Define the canonical product model.	CKO schema, domains, vocabulary, identifiers, governance roles.
3. Pilot products	Test the model on a representative set.	A small, varied product cohort that exercises every major domain.
4. Knowledge enrichment	Add meaning beyond commerce basics.	Education, trust, ingredient intelligence, relationships, and reusable modules.

PHASE	OBJECTIVE	PRACTICAL OUTPUT
5. Experience modernization	Render knowledge into customer experiences.	A Product Knowledge Experience and reusable presentation components.
6. Channel syndication	Distribute governed knowledge everywhere.	Mappings, channel rules, APIs, events, feeds, and publishing workflows.
7. AI enablement	Make AI a native consumer of governed knowledge.	Grounded retrieval, approved answers, semantic relationships, agent interfaces.
8. Continuous optimization	Improve quality and outcomes over time.	Analytics, governance reviews, model refinements, new modules and channels.

Implementation guidance

- **Improve before replacing.** Preserve useful investments and clarify their roles.
- **Prioritize reusable knowledge.** Enrich fields and relationships that serve several experiences.
- **Deliver value early.** Use a pilot to improve a visible customer or operational outcome.
- **Govern continuously.** Ownership and approval are part of the model from the first product.
- **Design for expansion.** The pilot should exercise every major domain even when the initial catalog is small.

10 / MEASURING SUCCESS

Measure better knowledge and better decisions, not integration volume

MEASURE	SIGNALS OF PROGRESS
Knowledge quality	Completeness, accuracy, validation pass rate, fewer conflicting records.
Operational efficiency	Less duplicate maintenance, faster changes, faster launches, fewer manual channel edits.
Customer understanding	Engagement with education, stronger discovery, fewer support questions, fewer misunderstanding-driven returns.
Channel performance	Consistent syndication, fewer feed errors or suppressions, improved marketplace coverage.
AI reliability	Higher retrieval accuracy, consistent answers, fewer hallucinations, broader approved use cases.
Business outcomes	Improved conversion, subscription adoption, confidence, retention, and product fit.

Common implementation mistakes

Treating CKA as a website project

The website is one consumer. The architecture must serve the enterprise.

Migrating everything at once

A limited pilot reduces risk and reveals where the model needs refinement.

Duplicating existing systems

CKA coordinates specialized systems; it does not absorb every operational capability.

Ignoring governance

Ungoverned enrichment creates a more sophisticated version of the same inconsistency.

Optimizing for today's platform

Platform-specific modeling undermines portability and long-term value.

11 / WHERE IT APPLIES

The model is useful wherever product understanding is fragmented

CONTEXT	HIGH-VALUE APPLICATION
Brands and retailers	One governed foundation for web, marketplace, retail, support, wholesale, and print.
Regulated or technical categories	Claims, evidence, specifications, ingredients, testing, and traceability.
Marketplaces and commerce platforms	Cleaner structured data, richer product understanding, and portable experience modules.
Brand aggregators	Shared governance and knowledge across portfolios without erasing brand identity.
AI and search providers	Retrieval-ready product knowledge, better grounding, recommendations, and agentic shopping.
Consultancies and integrators	A repeatable architectural framework for modernization programs.
Researchers and standards groups	A practical model connecting knowledge graphs, HCI, commerce, governance, and AI.

The first strong implementations are likely to emerge with design partners: organizations willing to contribute requirements, domain expertise, implementation feedback, and real operating constraints. The architecture should be built with industry rather than merely presented to it.

PART V Perspective and direction

12 / EVOLUTION

The architecture emerged from a sequence of practical questions

CKA did not begin as a theory or a product-page redesign. Each preceding initiative explored one part of the problem: portable experiences, governed records, structured commerce, reusable education, modular capabilities, and a practical reference implementation. Connecting those lessons revealed the missing architectural layer.



01 **Branded Pipeline** **PORTABLE EXPERIENCE**

A brand should remain recognizable wherever a customer encounters it.
project / prototype



02 **Master Catalog** **GOVERNED RECORD**

The product needs one authoritative definition, not departmental copies.
project / prototype



03 **FeedZip** **STRUCTURED COMMERCE**

Product knowledge must be structured and compliant before machines can trust it.
feedzip.ai



04 **SuppFacts** **EDUCATION LAYER**

Education is reusable product infrastructure, not disposable campaign content.
suppfacts.com



05 **Borrego OS** **MODULAR CAPABILITY**

Capabilities should be replaceable without changing the surrounding architecture.
project / prototype



06 **LogicShop** **REFERENCE IMPLEMENTATION**

The ideas converge in a practical product-knowledge system.
project / prototype



CKA

07 **Commerce Knowledge Architecture** **FORMAL ARCHITECTURE**

A durable model for governing product knowledge and expressing it across any interface.

AI makes trustworthy product knowledge more important, not less

AI will become another consumer of product knowledge. Customer assistants must explain products accurately. Sales and support systems need approved claims, specifications, comparisons, and evidence. Recommendation engines need relationships among products, attributes, ingredients, goals, and use cases. Agentic systems need to evaluate options, verify constraints, and act.

These capabilities fail when knowledge is scattered across pages and departmental files. The organizations best positioned for AI will not necessarily have the most tools. They will have the most trustworthy, structured, governed product knowledge.

Commerce has spent decades optimizing transactions. The next decade will be spent organizing knowledge.

The durable idea

Products are becoming knowledge objects. Commerce is the first obvious application because the cost of fragmentation is already visible, but the pattern extends beyond any platform, category, or channel. A product should remain understandable and trustworthy wherever people or machines encounter it.

CLOSING

A shared foundation for product understanding

CKA does not replace the teams and systems that make commerce work. It gives them a shared foundation. It defines what the product means, who governs each part of that meaning, and how every experience can consume it without recreating it.

The framework should be tested, challenged, and refined through implementation. Its underlying belief is simple:

A product should exist once as governed knowledge and be expressed appropriately wherever it is needed.

That is the idea this document is meant to advance.