PART II — FROM PHYSICAL TO DIGITAL · CHAPTER 05

Digital Materials: APIs, Data and Constraints

05

A product takes shape through the materials we can obtain, transform and trust. In a marketplace, APIs and data carry those same conditions into every customer promise.

Oct 5, 2026 23 min read Living edition 0.3 Status: in progress

A vacuum flask is a modest object, and that is why it teaches well. Its cobalt surface hides a vessel, a vacuum, a seal and a finish, each chosen for what it can do, where it can be sourced and how it behaves once someone carries it through a rainy commute. Nobody buying it sees those decisions; everybody feels them when one is wrong.

A marketplace hides decisions of the same kind. Behind a product page sit inventory feeds, payment services, address lookups and data copied from one system to the next, and each has properties that shift with the provider, the market and the day. Those properties set the limit of what the page can honestly promise.

This chapter treats APIs and data as materials. We start with the flask (its purpose, supply, process and market) and then follow the same questions into a hypothetical drinkware marketplace: payments, delivery, data provenance, regional rules and

security. It closes with the Digital Materials Map, a four-stage method for turning a customer promise into evidence a team can test.

591 mL	568 mL	23 mL
20 US FLUID OUNCES, CONVERTED	20 IMPERIAL FLUID OUNCES, CONVERTED	THE GAP BETWEEN THE TWO, IN ONE FLASK

Before the First Drawing



PLATE 5.1 A cobalt double-wall flask on a bare table: the whole product at a glance, before anyone asks what it is made of.

A vacuum flask rests on the table. Its cobalt surface looks simple, and the lid closes with a reassuring turn. Someone sees a useful container; someone else sees a gift. The designer sees steel, a seal, an insulating space and the processes that allow them to work together.

Move that flask into another market and the questions change. Can the same supplier deliver there? Does the finish survive the intended use? Will buyers understand its capacity? An object that looks complete still depends on where it will be made, sold, carried, cleaned and repaired.

A digital product enters the same conversation. The marketplace on a screen is assembled from services and information whose properties vary by provider and region. Before deciding what customers should see, we need to understand what these materials allow the business to promise.

The Purpose Inside the Object



Begin with a commuter who wants coffee to remain warm during a journey and arrive without leaking into a bag. The requirement sounds ordinary, but it gives the object a particular shape. The flask must fit in the available space, remain comfortable to carry, open without excessive effort and be easy enough to clean that yesterday's drink does not become today's problem. Capacity matters, but more capacity is not automatically more value.

Figure 5.2 — One need, three jobs: the commute, the job site and the desk.

An outdoor worker may need a larger container, a surface that remains manageable with gloves and a lid that tolerates dirt. A desk user may prefer an open tumbler because sipping convenience matters more than transport security. If these uses are collapsed into a single brief, the team can manufacture an object that performs adequately in a laboratory and awkwardly everywhere else. The purpose determines which material properties deserve attention and which compromises are acceptable.

Consider an illustrative double-wall stainless-steel flask. Its inner vessel contacts the drink, the outer shell takes handling and impact, the space between them contributes insulation, and the lid and gasket control closure. The alloy grade alone does not establish whether the finished product is suitable for its intended food-contact use. Surface condition, joining, seals, coatings where relevant, manufacturing controls and verification belong to the same decision.

That distinction will matter when we reach digital architecture. Saying that a platform uses a secure cloud or a compliant payment provider resembles saying that a flask uses good steel. It identifies one input. It does not yet explain the behavior of the assembled product, the conditions under which it works, or who will take responsibility when it fails.

A Material Has an Address



Figure 5.3 — The vessel can be made locally while the gasket travels.

A designer can specify a material that exists in a supplier's catalog and still discover that it is unavailable in the form, quantity or schedule the project needs. Sheet thickness, forming capability, tooling access, batch size and replacement components can all affect whether the design can be produced. Availability is therefore more specific than the existence of steel somewhere in the world.

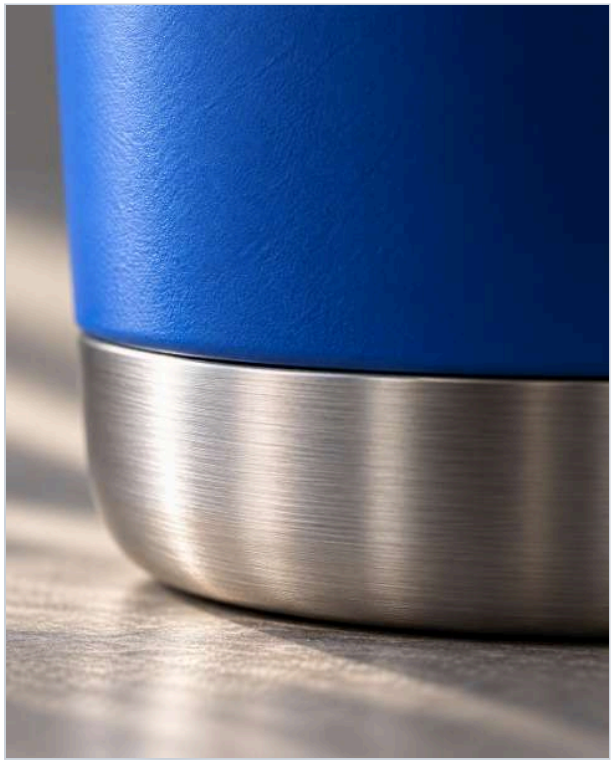
Imagine two manufacturing options. One supplier can produce the vessel and finish locally but imports the gasket. Another offers a complete assembly from an established production line, although replenishment requires a longer international journey. Neither option is inherently superior. The decision depends on volume, delivery variability, inspection, inventory exposure, repair strategy and the consequences of interruption. These are hypothetical options, not claims about the capabilities of a particular country.

The color creates another dependency. A cobalt finish may require a particular coating formulation, preparation process and application capability. Its appearance on a small sample does not guarantee consistency across a curved vessel or across production batches. A supplier may offer the desired color only at a minimum order that exceeds expected demand. Procurement has

now changed the design discussion: perhaps the launch needs fewer variants, a different process or a more patient replenishment plan.

Geography enters through that network of capabilities. Regional availability includes what can be sourced nearby, what must cross a border and what can be replaced after sale. A material choice that appears inexpensive at the factory can become costly when the business carries several months of inventory or cannot supply a replacement lid. The drawing contains an object; the supply arrangement determines whether the object can become a dependable product.

The Process Changes the Material



In one possible manufacturing route, metal is formed into the inner and outer vessels, trimmed, joined and prepared for assembly. The insulating space is evacuated and sealed; the lid and gasket are fitted; the exterior receives its specified treatment. The exact sequence and methods depend on the design and production system. The useful point is that the final properties emerge through processing, rather than arriving intact with the purchased inputs.

A forming operation can create defects that affect later joining. A seal can appear acceptable and

Figure 5.4 — Each stage inherits the condition the previous one left.

still fail under repeated use. A finish may adhere well initially but behave differently under abrasion, cleaning or temperature changes. Each stage inherits the condition produced by the previous stage, and an attractive surface can hide a problem the user discovers only after purchase.

The designer consequently needs evidence tied to the intended use. A thermal-retention test should specify initial temperature, ambient conditions, fill volume, duration and opening behavior. A leak test needs an orientation and handling scenario. A color or finish assessment needs a reference and a repeatable inspection method. Without those conditions, two suppliers can report successful tests while describing different performances.

The parallel with data processing is already visible. Information can be extracted correctly, transformed incorrectly and displayed beautifully. An API can return a technically valid response whose meaning changes during normalization. The final interface inherits these transformations just as the flask inherits its manufacturing history. Examining the finished surface is necessary, but it is too late to be the only form of quality control.

Color Meets the Market



Suppose the cobalt flask reaches buyers. Its color



Figure 5.5 — One name for the color, four different batches.

may suggest technical precision to one audience, personal style to another and an inconvenient mismatch to someone choosing accessories for a workplace. Weight may feel reassuring during an in-store demonstration and burdensome during a commute. These interpretations are hypotheses to investigate with the intended audience; they cannot be assigned reliably to an entire country.

The finish also participates in use. A textured surface may help handling but complicate cleaning. A glossy surface may show fingerprints or wear in ways the buyer dislikes. If photographs exaggerate the color or hide the actual texture, the gap between perception and experience becomes a source of returns. Aesthetic choices have become operational inputs.

We can examine acceptance by observing representative users carry, fill, open and wash the object, then compare those findings with purchases and returns. A stated preference for a color is different from willingness to buy it at the intended price. Sales can also mislead if one variant receives better shelf placement, more advertising or more reliable stock. Market measurement needs to explain the conditions that produced the result.

For the digital product, the equivalent question is whether the offered interaction feels credible and usable locally. A familiar payment method, an address form that accepts the customer’s actual address and a delivery estimate that proves reliable can matter more than visual consistency across countries. Acceptance

emerges from the meeting between a product's capabilities and people's circumstances.

Capacity Needs a Language



Figure 5.6 — Capacity measured to the brim is not capacity you can use.

A capacity label appears to be a small localization task. Yet "20 ounces" is incomplete when we do not know whether it means mass or fluid volume, and which fluid-ounce convention applies. NIST's conversion tables distinguish US and Imperial fluid ounces: approximately 29.57 mL and 28.41 mL respectively. Twenty US fluid ounces therefore correspond to about 591 mL; twenty Imperial fluid ounces correspond to about 568 mL.[1]

These differences do not authorize relabeling every 20 US fl oz flask as 600 mL. The manufacturer needs a verified capacity, an agreed measurement procedure and suitable market labeling. Useful fill capacity can also differ from volume measured to the brim. A buyer should not have to discover that distinction by overflowing the container when inserting the lid.

Capacity returns us to purpose. Does the container hold the drink sold by the nearby café? Can the user carry it comfortably? Is its opening suitable for filling and cleaning? Regional

conventions help people interpret the object, but physical volume still needs to satisfy the intended job. A conversion can be mathematically correct and commercially misleading if the original quantity was never defined.

Now imagine the flask entering a marketplace. One seller submits “20 oz,” another “600 ml,” and a third places the capacity only in an image. Search, filters and comparisons need a canonical representation, while buyers need a locally understandable display. If the platform silently treats every number as equivalent, the catalog introduces defects before the customer reaches checkout. Localization has exposed a data-model problem.

TABLE 5.1 — ONE BOTTLE, THREE WAYS TO STATE ITS CAPACITY

CAPACITY EXPRESSION	APPROXIMATE VOLUME	WHAT THE TEAM MUST ESTABLISH
20 US fl oz	591 mL	The US fluid-volume convention is intended
20 Imperial fl oz	568 mL	The Imperial fluid-volume convention is intended
600 mL	600 mL	The declared capacity and measurement procedure

From Vessel to Marketplace





PLATE 5.7 Many sellers, one shelf: the catalog is assembled from sources that never agreed on a format.

We will follow a hypothetical marketplace selling drinkware from several merchants, initially in Mexico and later in the United States and Portugal. It operates the shopping experience and coordinates sellers, payments and fulfillment partners. The scenario is illustrative: the service choices, thresholds and measurements below are proposed design examples rather than results from an actual implementation.

The marketplace needs product descriptions, seller identities, inventory, prices, payment services, address interpretation, shipping information and order records. Each input has a source and a transformation history. The capacity attribute might originate with a manufacturer, pass through a seller spreadsheet and become a searchable field. Stock might arrive from a warehouse API whose update schedule differs from the seller’s point-of-sale system.

Calling these inputs digital materials is useful because it encourages us to inspect properties before committing to a design. Data has freshness, completeness, provenance and permitted uses. APIs have supported operations, latency, quotas, error behavior and contractual boundaries. Their combination affects the experience in the same way that metal, seals and production processes affect the flask.

There is a limit to the analogy. A supplier can change an API after launch, withdraw a service or alter access conditions. Data can be copied across systems, and those copies can create additional obligations. Digital materials remain active dependencies throughout operation. The product team needs to design for their continuing behavior, not merely approve them at procurement.

TABLE 5.2 — FROM THE FLASK TO THE MARKETPLACE

PHYSICAL DESIGN QUESTION	MARKETPLACE EQUIVALENT
Is the required material available in this region?	Is the service contractually and operationally available for this market and business model?
What happens during forming and assembly?	What changes during ingestion, normalization and orchestration?
How does the finish behave in use?	How do the interface and its underlying services behave in local customer journeys?
Does the capacity specification mean the same thing?	Do units, currency, timestamps and business states retain their meaning?
Can a damaged component be replaced?	Can data and workflows move to another provider at an acceptable cost?

An API Contains a Promise

A syntactically correct response is the beginning of the contract, rather than its entire content.

An API is often described through its endpoints, authentication method and response schema. Those details are necessary, but a product decision requires an additional layer of meaning. An inventory response saying that five units are available does not tell us whether those units can be reserved, whether another channel can sell them simultaneously or how quickly the information becomes stale.

Suppose the page promises “available now,” while the seller updates stock once each evening. Engineering can integrate that feed perfectly and the product can still disappoint customers. The material supports a periodic estimate; the design presents an immediate assurance. The problem lies in the distance between the information’s properties and the promise attached to it.

A reservation operation changes what becomes possible. The marketplace might hold a unit while the buyer completes payment, subject to an expiry rule and a release process. A failed payment must release the reservation; a delayed confirmation needs reconciliation; repeated requests must not create multiple holds accidentally. A new endpoint has created new responsibilities across the journey.

An OpenAPI description helps document HTTP operations, parameters and response structures in a standard form.[2] The product team should complement it with business semantics: what counts as success, when a result is authoritative, how long it remains usable and what happens if confirmation never arrives. A syntactically correct response is the beginning of the contract, rather than its entire content.

Payment Begins With the Model



Figure 5.8 — Who sells, who receives funds and who refunds is settled before the checkout screen exists.

Before choosing a payment service provider, the marketplace needs to establish who sells to the buyer, who receives funds, who issues refunds and who carries the relevant commercial responsibilities. A single merchant selling its own inventory differs from a platform coordinating payments for multiple sellers. Seller onboarding, settlement, disputes and reconciliation may change with that model, even when the checkout screen looks almost identical.

This also clarifies a common ambiguity. We connect to a processor, gateway, acquirer or payment platform; PCI DSS is a security standard, not a service endpoint. The provider's validated scope, the integration method and the marketplace's own environment all matter. Outsourcing payment processing can reduce the requirements directly applicable to the merchant environment, but the PCI Security Standards Council states that responsibilities remain, including oversight of the third party.[3]

For our illustrative marketplace, a hosted payment experience may reduce direct exposure to card details. It also affects navigation, branding, available payment methods and the handling of interrupted journeys. The team must

understand the chosen integration's security and assessment implications with the provider and relevant payment stakeholders. A commercial claim that a vendor is "PCI compliant" does not settle those questions by itself.

Regional capability needs similar precision. A service may accept a customer's card in a country without supporting local seller accounts, desired settlement currencies or the marketplace's payout model. A method that fits one market may require a different provider or workflow elsewhere. The payment material shapes the business model and the customer journey together.

The Order Outlives Checkout



Consider a basket containing one cobalt flask from Seller A and a replacement lid from Seller B. The customer sees one purchase, but fulfillment may involve two inventories, two shipments and different cancellation opportunities. If the chosen payment arrangement cannot support the required allocations or refunds, the marketplace must either change its operating process or change its promise to the buyer.

Figure 5.9 — One purchase for the customer, two shipments for the operation.

Payment completion also has a time dimension. A synchronous response might be interrupted while the provider continues processing. A notification can arrive later, repeat or conflict with an earlier local state. The marketplace needs stable order identifiers, authenticated notifications, deduplication and reconciliation so that uncertainty does not become a duplicate charge or an unpaid shipment.

The interface should reflect these states. “Payment pending” gives the customer a different instruction from “payment failed.” A refund requested is different from a refund confirmed. If operations cannot explain the state using an order reference and a traceable record, customer support inherits an ambiguity created by architecture. The integration has become part of the service customers experience after the sale.

The flask comparison helps here. Testing whether a lid closes once is insufficient when the intended product must survive repeated opening, washing and transport. Similarly, confirming that a payment succeeds once in a sandbox says little about cancellations, partial refunds, retries or settlement. The material must be examined through the lifecycle the product actually offers.

A Pin Is Not Delivery



Figure 5.10 — The coordinate points at the building; the courier needs the entrance.

Geolocation looks just as simple from the interface. The buyer enters an address, a map shows a pin and the checkout estimates delivery. But address autocomplete, geocoding, routing and carrier serviceability answer different questions. Recognizing a location does not establish that a particular seller's carrier serves it, at the proposed price, within the promised period.

Our marketplace might encounter apartment details omitted by autocomplete, a rural address whose useful instructions include a landmark, or a coordinate located at a property's center rather than its accessible entrance. These possibilities should be tested with local address samples rather than converted into generalizations about national behavior. A provider's strong performance in one city does not establish its quality across a country.

Selection should therefore compare representative addresses, ambiguous cases, correction effort and actual carrier acceptance. The product can preserve the customer's original entry alongside the normalized form and request confirmation when the suggested result is uncertain. Device location can be optional, with manual address entry available; the usefulness of precise coordinates depends on the journey, not on the mere availability of a permission prompt.

Commercial and usage terms also affect design. Google's Geocoding API policies, for example, impose conditions on storing or caching content and on attribution, with specified exceptions such as place IDs.[4] We cannot assume that buying API access grants unrestricted ownership

of the returned information. A fallback provider or a long-lived address database may require a different arrangement from the one initially imagined.

Figure 5.1

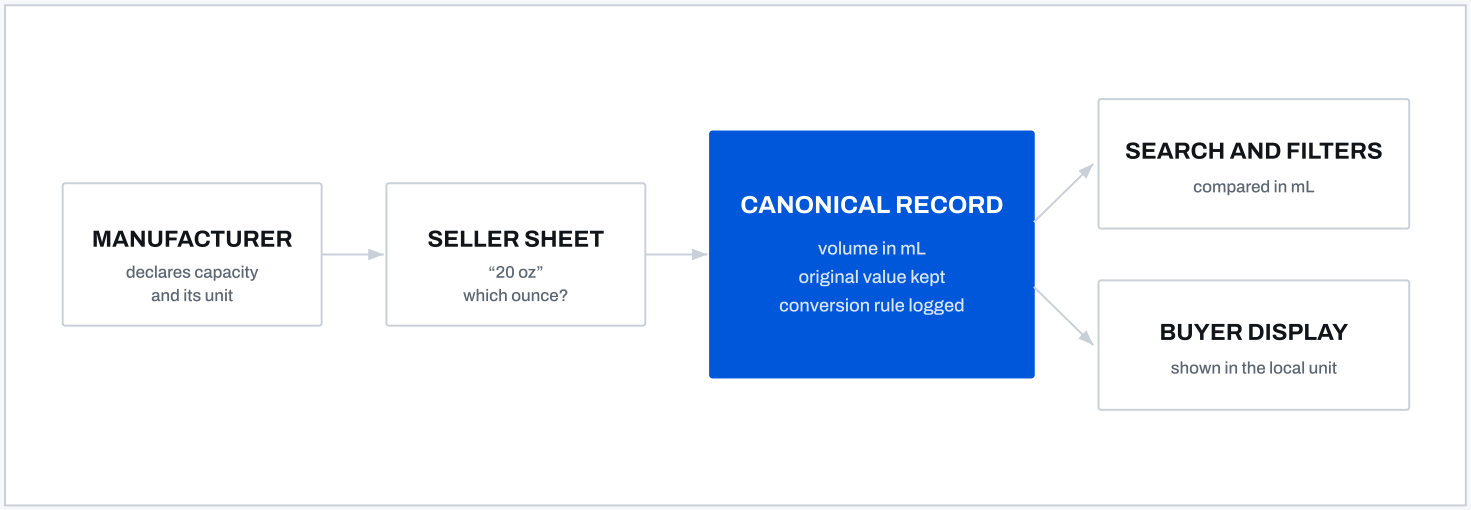


Figure 5.1 — The journey of one capacity value. The canonical record keeps the original value and the conversion rule, so an incorrect conversion can be traced and repaired.

Data Arrives With a History

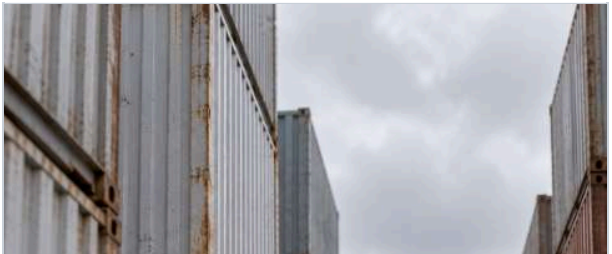
The marketplace's catalog is assembled from sources with different incentives and authority. A manufacturer may supply dimensions and care instructions; a seller supplies price and availability; a logistics partner reports shipment events; the customer supplies an address. An inferred preference derived from browsing is another kind of information again. Treating these records as equally authoritative makes the platform easier to build and harder to trust.

For the cobalt flask, the team should know who declared the capacity, which unit was supplied, whether the value was verified and when the record changed. Normalization should preserve the original value and the transformation rule so that an incorrect conversion can be traced and repaired. A derived field without provenance can spread the same mistake into search, recommendations, exports and advertising feeds.

Freshness also depends on the decision. A color description may remain valid for a long time; availability can change while a customer is browsing. The dataset therefore needs quality expectations connected to use. Completeness, validity, consistency, uniqueness, freshness and accuracy offer useful dimensions, but the target for each should be chosen according to the consequence of error.

Accuracy requires particular care. A value matching its allowed format is valid; it is accurate only if it represents the relevant reality. We can check that capacity is a positive number automatically, but that does not prove the flask holds the declared amount. Likewise, a coordinate can pass a format check and still send a courier to the wrong entrance. The production process for data needs evidence beyond clean formatting.

Geography Reaches the Copies



When the marketplace expands, data location becomes more complex than selecting a database region. An address can travel to a shipping partner, a fraud service, a support tool and an

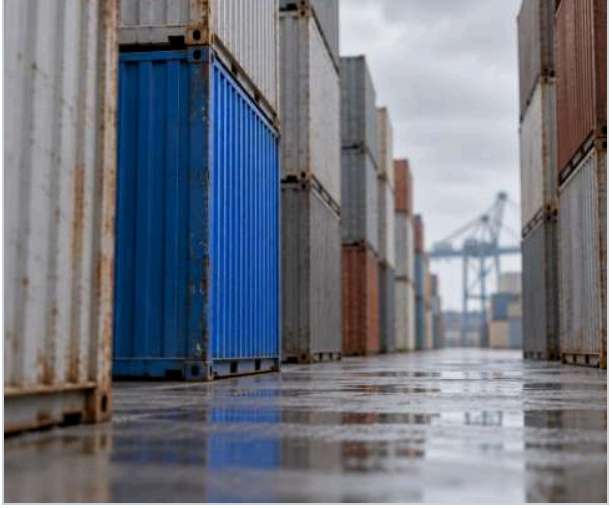


Figure 5.11 — Data, like cargo, leaves a record at every stop.

analytics pipeline. Copies may appear in logs, exports, backups and incident records. Remote access and provider arrangements also need examination; the location of the primary database does not describe the entire processing environment.

For European personal data, the GDPR's transfer framework requires attention when information is transferred outside the European Economic Area. The EDPB describes mechanisms including adequacy decisions and appropriate safeguards such as standard contractual clauses.[5] That is different from claiming that every European dataset must always remain physically inside Europe. The applicable arrangement depends on the actual flow and circumstances.

Mexico requires its own analysis. A new private-sector personal-data law took effect on 21 March 2025, replacing the 2010 text, and supervision moved from the former INAI to the Secretaría Anticorrupción y Buen Gobierno. Its principles include purpose, proportionality, information and accountability; it recognizes access, rectification, cancellation and opposition rights and addresses security and transfers.[6,8] A regional implementation needs those obligations reflected in notices, handling processes and system behavior, rather than inherited unquestioningly from another market.

The United States introduces another reason to avoid a single regional assumption. California's CCPA, where applicable, includes rights concerning access, deletion, correction, sale or sharing and certain uses of sensitive personal information.[7] That example does not describe

every US business or jurisdiction. It shows why a country selector is too coarse to serve as the sole privacy rule engine.

Security Becomes Product Behavior



Figure 5.12 — Different keys for different doors: access follows responsibility.

Security decisions change who can do what inside the marketplace. Seller A should see the orders needed to fulfill its sales without gaining access to Seller B's customers. A support agent may need delivery information without unrestricted access to exported personal records. Administrative actions need appropriate authorization and traceability because the platform can harm customers through misuse as well as through an external attack.

These decisions should follow specific threats and responsibilities. They affect service identities, access rules, secrets, notification verification, audit records and incident response. Encryption contributes protection, but it does not decide whether a user was entitled to retrieve a record or whether a provider should have received it. Product design needs to connect controls to the actual paths through which information and money move.

Deletion also exposes the architecture. Removing a customer profile does not automatically

remove data from every downstream system. Some records may need retention for a justified obligation, while information no longer needed should have a defined disposition. The product must explain and implement those differences, including how restored backups avoid silently reinstating information that should remain deleted.

A regional constraint register makes this manageable. Each requirement should identify the relevant market, data category, purpose, responsible entity, affected flow, implementation decision, reviewer and evidence date. Legal and security specialists establish the applicable obligations; product and engineering translate them into testable behavior. Compliance then influences the design while alternatives remain available, instead of becoming an unexplained obstacle at launch.

Figure 5.2

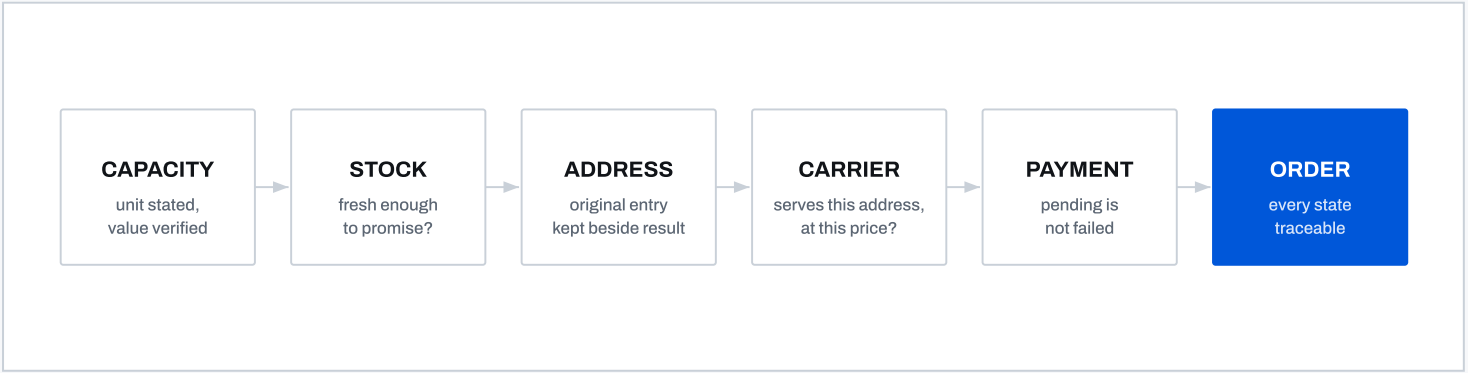


Figure 5.2 — One purchase, six materials. Each box carries a property the promise depends on: a verified unit, a freshness window, the original address entry, carrier coverage, pending versus failed states and a traceable order record.

Map the Digital Materials

Four stages connect a customer promise to the services, information and conditions that make it possible.

01

Define

State what the customer must be able to rely on, in testable terms.

02

Trace

Link each promise to the services and data that produce it.

03

Test

Gather regional evidence: samples, failures and lifecycle cases.

04

Decide

Launch, adapt or withhold, with an owner and a review trigger.

From Promise to Decision

We can bring these questions together through a four-stage **Digital Materials Map**: define, trace, test and decide. This is a working method proposed for this chapter, rather than a formal certification standard. Its purpose is to connect a customer promise to the services, information and conditions that make it possible, so that product, engineering, operations and control functions can examine the same decision.

Define the promise. Begin with a journey narrow enough to examine: a customer buys the cobalt flask for delivery to a specified address. Define what the platform intends to assert about capacity, stock, payment and delivery. “Support international commerce” is too broad to test. “Accept an eligible order, confirm payment without duplicate collection and quote a carrier-supported delivery option” gives the team identifiable states and dependencies.

Set acceptance criteria before comparing providers. For illustration, the team might require capacity units to be explicit, stock to be reserved before an order is finally confirmed, and ambiguous addresses to receive customer confirmation. A proposed latency target should state where it is measured, under what load and with which percentile. These are design targets to validate, not industry-wide benchmarks.

Trace the materials. Create one record for each consequential API and dataset. Include its owner, source or provider, supported markets, operations, schema, business meaning, authentication, sensitivity, processing locations, permitted uses, retention, quotas, cost drivers, failure behavior and exit conditions. Attach evidence and a review date. Unknown properties should remain marked as unknown rather than replaced with optimistic assumptions.

Trace the journey through those records. The capacity originates with the supplier; catalog normalization creates a canonical quantity; inventory supports availability; address interpretation suggests a location; carrier coverage establishes serviceability; payment and order services establish financial and commercial states. Draw the data recipients separately where necessary. A technical dependency map and a personal-data flow map overlap, but neither fully substitutes for the other.

Test the properties. Test candidate services with representative regional samples and lifecycle cases. For data, inspect missing units, conflicting seller descriptions, stale inventory and duplicate identifiers. For APIs, test timeouts, rate limits, authentication failures, duplicate notifications and recovery. Measure end-to-end behavior as well as individual services because a fast endpoint can still contribute to a slow or unreliable journey.

For example, suppose a geocoding pilot uses 200 appropriately handled test addresses in a target delivery area. If 164 are correctly resolved without customer intervention, the observed unassisted resolution rate is 82%. That invented result is a teaching example, not a provider evaluation. Segment the failures by relevant address characteristics and compare with carrier acceptance; the average alone cannot tell us whether the material supports the intended market.

Decide with conditions. Use mandatory gates before a weighted comparison. If a payment provider cannot contractually support the required seller model, a high usability score cannot make it eligible. If a data use is not permitted, low cost cannot compensate. Once candidates pass the necessary gates, compare quality, reliability, regional fit, economics and switching effort using weights agreed for the journey.

An illustrative score can use ratings from one to five and weights totaling one: quality 0.30, reliability 0.25, regional fit 0.20, economics 0.15 and switching effort 0.10, with higher switching scores meaning easier exit. Record the evidence behind each rating and check whether reasonable weight changes reverse the choice. The score supports a decision; it does not eliminate uncertainty or override a failed gate.

200	164	82%
TEST ADDRESSES IN THE ILLUSTRATIVE PILOT	RESOLVED WITHOUT CUSTOMER HELP	UNASSISTED RESOLUTION (AN INVENTED EXAMPLE)

Measure the Assembled Promise

The resulting map should support a short set of measures that operations can maintain. A provider's availability percentage is useful, but the customer depends on successful completion of a journey. Likewise, counting geocoding responses says little about correct delivery. The denominator needs to represent the decision we are evaluating, and exclusions need to be visible.

Suppose the cheapest mapping service requires more customer corrections and produces more failed delivery attempts. Its request price is lower, but its contribution to cost per fulfilled order may be higher. The same logic applies to a payment provider with inexpensive successful charges but costly exceptions, or an inventory feed that is cheap to ingest but expensive to correct after overselling.

Measure these outcomes by market and meaningful segment. A launch limited to one delivery area should not be presented as proof of national coverage. A pilot conducted with carefully curated seller records should not establish a quality claim for unrestricted onboarding. The evidence defines the boundary of the decision, just as a thermal test defines the conditions under which the flask's retention claim was demonstrated.

TABLE 5.3 — FIVE MEASURES FOR THE ASSEMBLED PROMISE

MEASURE	PRACTICAL DEFINITION	DESIGN CONSEQUENCE
Catalog unit completeness	In-scope capacity records with explicit, recognized units / all in-scope capacity records	Hold ambiguous listings for correction
Stock freshness compliance	Required stock records within the agreed freshness limit / records assessed	Restrict immediate availability promises

MEASURE	PRACTICAL DEFINITION	DESIGN CONSEQUENCE
Unassisted address resolution	Addresses correctly resolved without customer correction / representative addresses tested	Introduce confirmation or manual alternatives
Payment reconciliation coverage	Payment attempts linked to an explained authoritative state / attempts in the reconciliation window	Investigate unresolved states before fulfillment
Cost per fulfilled order	Attributable service costs, including failed attempts and retries / fulfilled orders	Compare providers against achieved business output

Let Evidence Change the Design

A promise blocked by an unknown dependency is different from a feature postponed for lack of capacity.

Imagine that our first pilot reveals good catalog quality but weak inventory timeliness from several sellers. The design could allow those sellers to list products while withholding immediate availability claims, or require a stronger inventory integration before enabling that promise. The appropriate response depends on the business model, but either choice acknowledges the material's actual properties.

Now suppose address testing shows that autocomplete performs well in the initial delivery area but does not reliably handle another

intended area. The marketplace can retain manual entry, validate serviceability with carriers and narrow its launch boundary while investigating alternatives. A polished map interaction should not force the business to accept addresses it cannot serve dependably.

Expansion into Portugal might also reveal different payment arrangements, seller onboarding requirements and personal-data flows. The team can retain common definitions for orders, quantities and state transitions while varying providers and approved processing arrangements. Shared architecture should preserve coherent meaning; it need not impose an identical supply network on every region.

Those decisions belong in the roadmap with owners and conditions. A promise blocked by an unknown dependency is different from a feature postponed for lack of capacity. Showing that distinction helps leaders invest in the missing evidence, contractual arrangement or service capability. Product design then evolves through informed choices about materials, rather than through repeated surprise during implementation.

What the Flask Teaches Us



Return to the cobalt flask. Its visible simplicity is the result of many coordinated choices: a purpose, a material specification, a production route, a closure, a finish, a capacity convention and a supply arrangement. When those choices fit, the buyer can use the object without knowing their details. When they do not, the buyer



Figure 5.13 — The same question, in two materials.

experiences the unresolved decision as leakage, inconvenience, wear or disappointment.

The marketplace achieves a similar simplicity only when its underlying services and information support the offered experience. The customer should not need to understand reservation expiry, provider notifications, address normalization or international transfer arrangements. The team must understand them well enough to make the purchase dependable.

This is why APIs and data belong in product discovery. They determine which promises can be made, where they can be made and what maintaining them will require. Regional constraints are part of that design space, alongside commercial capability, security, operating cost and market acceptance. Mapping those conditions early allows the team to shape a product that can survive beyond its demonstration.

If a product's behavior depends on the properties of its materials, and those properties change with source, process and region, then its design must begin with those conditions. The interface is where the customer encounters the decision. It is not where the decision begins.

REFERENCES

Sources checked on 5 October 2026. Examples, pilot figures and evaluation weights are illustrative. The regional discussion identifies design questions; applicable obligations depend on the actual entities,

activities, contracts and data flows.

1. **NIST — Measurement conversions.** Approximate conversions from US customary measures to metric and SI Guide, Appendix B.9. US and Imperial fluid-ounce conversions.
2. **OpenAPI Initiative — OpenAPI Specification.** Standard descriptions of HTTP API interfaces; the material records in this chapter add product and operational context.
3. **PCI Security Standards Council — Does PCI DSS apply to merchants who outsource all payment processing?.** Outsourcing, reduced scope and retained merchant responsibilities. See also PCI DSS.
4. **Google Maps Platform — Policies and attributions for Geocoding API.** Usage, storage and attribution conditions; review the applicable agreement for the intended implementation.
5. **European Data Protection Board — International data transfers.** Transfer mechanisms and safeguards under the GDPR framework.
6. **Cámara de Diputados — Ley Federal de Protección de Datos Personales en Posesión de los Particulares.** Current text displayed with the reform dated 14 November 2025; especially the provisions concerning principles, rights, security and transfers.
7. **California Department of Justice — California Consumer Privacy Act.** Rights and applicability under the CCPA as amended; this is a California example, not a summary of all US privacy requirements.
8. **Creel — Federal Law for the Protection of Personal Data in the Possession of Private Parties.** Client alert on the 20 March 2025 decree: new law in force on 21 March 2025, 2010 law repealed, supervisory functions transferred from the INAI to the Secretaría Anticorrupción y Buen Gobierno.

← CHAPTER 04

The Product Is Never Just the Object

CHAPTER 06 →

Designing for Maintenance, Not Just Launch